Schriften zur Informationswissenschaft 39

Thomas Mandl

Tolerantes Information Retrieval

Neuronale Netze zur Erhöhung der Adaptivität und Flexibilität bei der Informationssuche

UVK

Hochschulverband für Informationswissenschaft (HI) e.V. Konstanz

Vorwort

Die vorliegende Arbeit versucht, einen Beitrag zum Bereich Mensch-Maschine-Interaktion zu leisten und beschäftigt sich mit Information Retrieval. Im Mittelpunkt steht die empirische Untersuchung von lernenden Systemen bei Suchprozessen.

Diese Arbeit entstand in den Jahren 1995 bis 2000 an verschiedenen beruflichen Stationen. In dieser Zeit war ich zunächst bei der Fachgruppe Linguistische Informationswissenschaft der Universität Regensburg und dann von 1995 bis 1998 am Informationszentrum Sozialwissenschaften in Bonn im Forschungsprojekt ELVIRA (Elektronisches Verbandsinformations-, Retrieval- und Analysesystem) tätig. Im Wintersemester 1998 wechselte ich zur Informationswissenschaft an der Universität Hildesheim, wo der Fachbereich IV Sprachen und Technik diese Arbeit im Dezember 2000 als Dissertation annahm.

Wichtig für den Erfolg der Arbeit war das Projekt ELVIRA, an dem ich in Bonn und Regensburg beteiligt war. Möglich wurde dieses Projekt durch die fruchtbare Kooperation mit dem Zentralverband der Elektrotechnik- und Elektronikindustrie, Frankfurt (ZVEI) und durch die Förderung des Bundesministeriums für Wirtschaft (BMWi, Fördernummer IV C2-003060/22).

Ich danke all denen, die zum Gelingen dieser Arbeit beigetragen haben. Zuerst gebührt der Dank Prof. Dr. Christa Womser-Hacker (Informationswissenschaft, Universität Hildesheim), die meine akademische Laufbahn buchstäblich vom ersten Moment an begleitet hat und die Dissertation in allen Phasen befördert hat. Ihr Interesse und ihre wertvolle, konstruktive Kritik waren in der Zeit in Hildesheim besonders intensiv. Undenkbar wäre diese Arbeit auch ohne Prof. Dr. Jürgen Krause (Informatik, Universität Koblenz-Landau, Informationszentrum Sozialwissenschaften, Bonn), der mich gerade in der Regensburger und Bonner, aber auch in der Hildesheimer Zeit als engagierter akademischer Lehrer förderte. Ich danke auch dem dritten Gutachter, Prof. Dr. H.-J. Bentz (Mathematik, Universität Hildesheim), der in der Endphase noch wertvolle Anregungen gab. Weiterhin danke ich Prof. Dr. Christa Hauenschild, die den Vorsitz der Prüfungskommission übernahm. Ihr und allen anderen Beteiligten am Promotionsverfahren gilt mein Dank für die unkomplizierte Durchführung.

Daneben danke ich allen Kolleginnen und Kollegen, die meine Arbeit auf unterschiedlichste Art und Weise unterstützt und vorangebracht haben, sei es durch anregende Diskussionen, praktische Tipps, das Bereitstellen von Daten

oder das Schaffen von günstigen Arbeitsbedingungen. Namentlich nennen möchte ich Dr. Maximilian Eibl, Matthias Müller und Maximilian Stempfhuber.

Zu Dank bin ich auch der Firma *Telcordia* (vorher *Bellcore*) verpflichtet, die die Software für *Latent Semantic Indexing* (LSI) unentgeltlich zur Verfügung gestellt hat.

Für die Aufnahme in die Reihe *Schriften zur Informationswissenschaft* möchte ich Prof. Dr. Rainer Kuhlen (Informationswissenschaft, Universität Konstanz), den Mitgliedern des wissenschaftlichen Beirates der Schriftenreihe und dem Hochschulverband Informationswissenschaft (HI) danken.

Ganz besonders danke ich meiner Familie, insbesondere meiner Frau Patrícia. Sie und mein Sohn Matthias mussten mich oft und lange entbehren und haben mir trotzdem viel Kraft geschenkt.

Hannover, im Januar 2001 Thomas Mandl

Inhaltsverzeichnis

Zusammenfassung:

Information Retrieval befasst sich mit vagen Anfragen und der vagen Modellierung von Benutzerverhalten. Neuronale Netze sind eine Methode zur vagen Informationsverarbeitung und zur Implementierung kognitiver Fähigkeiten. Diese Arbeit gibt einen umfassenden Überblick über den state-of-the-art zu neuronalen Netzen im Information Retrieval und analysiert, gruppiert und bewertet zahlreiche Systeme.

Als Konsequenz von Schwächen bestehender Modelle wird das COSIMIR-Modell entwickelt, das auf dem neuronalen Backpropagation-Algorithmus aufbaut. Es erlernt den im Information Retrieval zentralen Vergleich zwischen Dokument und Anfrage anhand von Beispielen. Die kognitive Modellierung ersetzt so ein formales Modell und führt zu höherer Adaptivität und damit zu verbesserter Toleranz gegenüber Benutzereigenschaften. Das Transformations-Netzwerk ist ein weiteres System, das auf dem Backpropagation-Algorithmus basiert und Retrieval bei heterogenen Daten ermöglicht. In mehreren Experimenten werden das COSIMIR-Modell und das Transformations-Netzwerk mit realen Daten getestet. Das COSIMIR-Modell hat sich dabei für Fakten-Retrieval bewährt. Die Experimente mit dem Transformations-Netzwerk und alternativen Verfahren ergaben je nach Datengrundlage unterschiedliche Ergebnisse. Das optimale Verfahren hängt also vom Anwendungsfall ab. Bei gleicher Qualität ist die Überschneidung der Ergebnisse verschiedener Verfahren relativ gering, so dass Fusionsverfahren erprobt werden sollten.

Abstract:

Information retrieval deals with vague queries and vague models of user behavior. Neural networks are a method to process vague information and implement cognitive skills. This thesis provides an overview of the state-of-the-art of neural networks in information retrieval by analysis, clustering and evaluation of a large number of systems.

The shortcomings of existing models have led to the development of the COSIMIR model, which is based on the neural backpropagation algorithm. It learns from examples to compare queries and documents, a central process in information retrieval. The cognitive approach replaces a formal model and leads to higher adaptivity and tolerance with regard to user interests. The transformation network is another system which is based on the backpropagation algorithm and which makes the retrieval of heterogeneous data possible. In several experiments, the COSIMIR model and the transformation network are tested with real world data. The COSIMIR model achieves good results for factual retrieval. The experiments with the transformation network and alternative methods lead to different results for different data sets. Which method performs best depends to a considerable extent on the data. Where the different methods are of comparable quality, the overlap of the results is relatively low. It is therefore recommended that fusion methods be employed.

1 Einleitung

Diese Arbeit aus der Informationswissenschaft befasst sich mit Modellen, die sowohl die Adaptivität als auch die Flexibilität von Informationssystemen verbessern.

Information Retrieval (IR) Systeme suchen nach Informationen in großen Datenmengen. Der Benutzer[1] formuliert eine Anfrage und erhält eine Menge von Dokumenten als Ergebnis. Im ihrem Kern benutzen IR Systeme mathematische Modelle, die weder natürliche Sprachen noch andere kognitive Fähigkeiten des Menschen adäquat modellieren. Da die Entwicklung solcher Modelle mittelfristig nicht absehbar ist, müssen andere Möglichkeiten gefunden werden, um die Qualität von Information Retrieval Systemen und ihre Toleranz gegenüber menschlichen Fähigkeiten zu verbessern.

Das in dieser Arbeit vorgestellte COSIMIR-Modell verankert lernende Komponenten im Kern von Information Retrieval Systemen und verbindet so die menschliche Fähigkeit, die Relevanz eines Textes oder allgemeiner gesprochen eines Dokuments zu einer Anfrage einzuschätzen, mit bestehenden mathematischen Modellen. Das Ausnutzen von bereits gefällten Entscheidungen erlaubt die Integration kognitiver Fähigkeiten, ohne dass diese vollständig analysiert und modelliert sind. Im Mittelpunkt stehen neuronale Netze, ein häufig benutzter Ansatz zur Modellierung vager, kognitiver Prozesse.

1.1 Fachliche Einordnung

Mit dem Gebiet Information Retrieval befassen sich Wissenschaftler aus verschiedenen Fachgebieten, insbesondere aus der Informationswissenschaft und der Informatik.

Diese Arbeit positioniert sich in der Informationswissenschaft wie sie etwa Kuhlen 1999 definiert. Informationswissenschaft befasst sich mit Informationsprozessen, bei denen Information aus Wissen entsteht. Information ist dabei die relevante Teilmenge aus dem zur Verfügung stehenden Wissen, die ein Benutzer während der Interaktion mit einem Informationssystem erarbeitet. Was letztendlich in einem konkreten Informationsprozess zu Information

[1] Bei männlichen Plural-Formen sind auch weibliche Personen gemeint. Die Techniken, die dies auszudrücken, sind jedoch nicht sehr ökonomisch und lesefreundlich, so dass der Autor nach Abwägung auf sie verzichtet hat.

wird, bestimmen der Benutzer und sein individueller Kontext. Für Informationssysteme ist also „Pragmatik-Design“ (Kuhlen 1999:142) entscheidend.

Dementsprechend stehen die Benutzer im Zentrum des Interesses informationswissenschaftlicher Forschung. Dem trägt auch diese Arbeit Rechnung. Die adäquate Modellierung der kognitiven Prozesse, um ein Informationssystem tolerant an die pragmatische Situation des Benutzers anzupassen, ist das oberste Ziel. Auf dem Weg dorthin thematisiert die Arbeit in erheblichem Umfang auch Fragen der Formalisierung und Algorithmisierung.

Eine strikte Abgrenzung der Arbeit oder auch einzelner Teile zur Informatik ist weder notwendig noch sinnvoll. Die Informatik betont bei der Sicht auf Informationsprozesse die algorithmischen und formalen Aspekte, die auch hier eine große Rolle spielen. Das primäre Interesse jedoch, das sich in Ausgangspunkt und Zielen widerspiegelt, liegt auf einer pragmatischen Sichtweise im Sinne der Informationswissenschaft.

1.2 Information Retrieval und Heterogenität

Ein wesentliches Bedürfnis im Rahmen der Mensch-Maschine-Interaktion ist die Suche nach Information. Das Finden und die damit verbundene Speicherung und inhaltliche Erschließung von Text-Dokumenten gewinnt mit der fortschreitenden Verbreitung des Internet und der steigenden Menge an online bereitstehenden Texten stark an Bedeutung. Der Bereich Information Retrieval befasst sich mit der Erschließung und dem Finden von Dokumenten, die meist auf Text-Dokumente beschränkt sind. Diese Arbeit nimmt eine umfassendere Sichtweise ein und bezieht auch Fakten-Retrieval-Systeme mit ein, die sonst häufig unter dem Begriff Datenbanksysteme behandelt werden. Auch bei Fakten ergeben sich aus Sicht des Benutzers ähnliche Probleme wie bei IR-Systemen für Texte.

Bei der inhaltlichen Erschließung von Texten analysieren die meisten IR-Systeme kaum Syntax und Semantik. Linguistische Komponenten beschränken sich auf die Ebene der Morphologie zur Reduktion von Wörtern auf ihre Stammformen oder der Analyse von Komposita. Die Repräsentation der Textdokumente enthält Informationen über das Vorkommen und die Häufigkeit von Begriffen in den Texten. Die Anfrage wird ebenso behandelt, so dass Dokument und Anfrage homogen repräsentiert und verglichen werden. Fast alle IR-Systeme berechnen zwischen den auf diese Weise gewonnenen Repräsentationen von Dokument und Anfrage eine Ähnlichkeit auf der Basis mathematischer Ähnlichkeitsfunktionen. Die Auswahl der Ähnlichkeitsfunktion erfolgt aufgrund heuristischer Faktoren und basiert nicht auf kausalen

Zusammenhängen zwischen den Eigenschaften der Funktion und den Anforderungen des Anwendungsbereiches.

Ein weitere Schwäche bestehender IR-Modelle ist die Behandlung von Heterogenität der Informationsobjekte. Bei der Informationssuche wollen Benutzer häufig verschiedene Modalitäten abfragen, also z.B. Texte, Grafiken und numerische Daten zu ihrem Problem finden. Die technischen Schwierigkeiten der Heterogenität sind heute weitgehend gelöst, während die semantischen Probleme meist ignoriert werden.

1.3 Neuronale Netze und kognitive Modellierung

Die Modellierung im Information Retrieval geht bisher weitgehend nicht von den kognitiven Eigenschaften des Menschen aus, sondern von mathematischen Modellen. Dies liegt natürlich vor allem daran, dass über die kognitive Informationsverarbeitung noch nicht genügend Kenntnisse vorliegen. Um aber IR-Systeme an den Menschen anzupassen und für den Benutzer optimal zu gestalten, muss sich die Modellbildung stärker an den kognitiven Eigenschaften orientieren.

In dieser Arbeit werden lernende Systeme eingesetzt, um menschliche Entscheidungen zu modellieren. Der empirisch zugängliche Ausdruck der kognitiven Fähigkeiten und nicht ihre interne Struktur wird zum Gegenstand. Ein derartiges System lernt anhand von Beispielen die Lösung des Problems.

Neuronale Netze sind ein Modellierungsverfahren, das von der Funktionsweise im menschlichen Gehirn inspiriert ist. Neuronale Netze zeichnen sich durch Fehlertoleranz, Effizienz und die Fähigkeit zur vagen Verarbeitung von Informationen aus. Sie zählen zu dem Paradigma *Soft Computing*. Diese Familie von Verfahren zur *weichen* Informationsverarbeitung sieht bewusst von der exakten Modellierung eines Problems ab. Exaktheit ist oft wegen hoher Komplexität oder inhärenter Vagheit nicht erreichbar. *Soft Computing* erlaubt es, mit bewusst nicht exakten Modellen zu agieren.

1.4 Neuronale Netze im Information Retrieval

Zahlreiche Information Retrieval Modelle basieren bereits auf neuronalen Netzen, jedoch lösen sie die Schwächen bestehender IR-Systeme nicht umfassend und schöpfen die Möglichkeiten des „Soft Computing“ nicht vollständig aus.

Das in dieser Arbeit vorgestellte COSIMIR-Modell (Cognitive Similarity learning in Information Retrieval) reagiert mit vager Modellierung auf einige

der Probleme bestehender IR-Systeme. COSIMIR trägt die lernende Komponente in Form eines neuronalen Netzes und damit die kognitiv adäquate Modellierung in den Kern eines Information Retrieval Systems. Es toleriert die vage Modellierung der schwer exakt zu fassenden menschlichen Eigenschaften, die bei der Interaktion mit dem System entscheidend sind.

Zudem ist COSIMIR tolerant gegenüber heterogenen Repräsentationen und damit heterogenen Dokumenten. Ein weiteres neuronales Netzwerkmodell, das sich besonders für die Heterogenitätsbehandlung im Information Retrieval eignet, ist das Transformations-Netzwerk, das aufgrund von Beispielen lernt, die Repräsentation eines Objekts von einem Repräsentations-Schema in ein anderes zu übertragen.

1.5 Experimentelle Evaluierung

Neben der Modellierung von Informationssystemen spielt auch die Evaluierung eine entscheidende Rolle. Besonders die Informationswissenschaft betont die empirische Überprüfung ihrer Modelle in benutzerorientierten und damit praxisbetonten Experimenten. Dem trägt auch diese Arbeit Rechnung. Die Evaluierung der vorgeschlagenen Modelle für Information Retrieval Systeme nimmt eine wichtige Stellung ein.

Im Information Retrieval hat in den letzten Jahren die Diskussion um Evaluierung neue Impulse erhalten. Die Text Retrieval Conference (TREC) hat sich als wichtige Messlatte etabliert. TREC ist eine Initiative des National Institute of Standards and Technology (NIST), das eine sehr große Kollektion von Texten, Anfragen und Relevanzbewertungen anbietet, mit der die Entwickler von IR-Sytemen experimentieren. Die Ergebnisse analysiert und vergleicht das NIST nach standardisierten Verfahren. Nicht zuletzt die steigende Zahl von Teilnehmern zeigt die Bedeutung von TREC (cf. Voorhees/Harman 1999).

Nach wie vor sind aber auch Experimente außerhalb des TREC-Kontexts sinnvoll und sogar notwendig. Zum einen bestehen die TREC-Daten hauptsächlich aus Zeitungstexten, und es ist bekannt, dass sich die Ergebnisse nicht ohne weiteres auf andere Bereiche übertragen lassen, sondern dass für unterschiedliche Texttypen weitere empirische Untersuchungen erforderlich sind. Auch die Problematik der Heterogenität untersucht TREC nicht.

Zum anderen zeigt sich bei TREC, dass die Qualität eines einzelnen Systems offensichtlich nicht beliebig gesteigert werden kann. Die besten Systeme bei TREC erreichen sehr ähnliche Qualität, die Ergebnisse selbst sind jedoch sehr verschieden. Jedes System liefert andere relevante Dokumente. Fusionsansät-

ze versuchen vermehrt dies auszunutzen und dieser Trend wird sich vermutlich verstärken. Ein neues Information Retrieval System muss also nicht auf Anhieb die beste Performanz erreichen, vielmehr ist wichtig, dass es wiederum von anderen Systemen nicht gefundene Dokumente nachweist. Mit diesem Ergebnis hat sich TREC in gewissem Maße selbst relativiert. Die in dieser Arbeit vorgestellten Modelle erfordern teilweise spezifische Evaluierungsformen, so dass dies bei der Suche nach geeigneten Daten im Vordergrund steht und nicht auf TREC-Daten zugegriffen wurde.

1.6 Aufbau der Arbeit

Das folgende, zweite Kapitel gibt einen Überblick über den Bereich Information Retrieval und betrachtet besonders die Problematik und Behandlung von Vagheit. Es weist v.a. auf Schwächen bestehender IR-Systeme hin. Kapitel 3 behandelt neuronale Netze, wobei ein Kompromiss zwischen einer leicht lesbaren Einführung und der Diskussion komplexer Details und aktueller Ergebnisse gesucht wurde. Dazu gehört auch die Entscheidung für das formale Niveau bei der Behandlung dieses sehr formalen und mathematischen Gegenstandes. Algorithmische Details finden sich, soweit sie für das Verständnis folgender Kapitel wichtig sind.

Kapitel 4 kombiniert die Bereiche der beiden vorhergehenden Kapitel und diskutiert den Einsatz neuronaler Netze im Information Retrieval. Ein umfassender state-of-the-art Bericht stellt zahlreiche Systeme und Modelle vor, bewertet sie und positioniert sie in ihrem Kontext. Es zeigt sich, dass sie das Potenzial neuronaler Netze nicht vollständig ausschöpfen.

Kapitel 5 greift ein Thema aus Kapitel 2 auf und gibt einen detaillierten Überblick über die Herausforderungen und Behandlung von Heterogenität im Information Retrieval. Bestehende Systeme reagieren noch zu wenig auf die semantischen Probleme der Heterogenität. Zu den Verfahren zur Heterogenitätsbehandlung gehört auch das neuronale Transformations-Netzwerk.

Kapitel 1 baut auf den beiden vorhergehenden Kapiteln auf und entwickelt im Rahmen neuronaler Netze Lösungsmöglichkeiten für die dort beschriebenen Probleme. Es führt das COSIMIR-Modell (COgnitive SIMilarity Learning in Information Retrieval) ein, das eine Reaktion auf den in Kapitel 4 erläuterten State-of-the-art zum Einsatz neuronaler Netze im Information Retrieval darstellt. Neben einem Überblick zu den Vor- und Nachteilen von COSIMIR und seinen Erweiterungsmöglichkeiten diskutiert Kapitel 1 denkbare Architekturen für ähnliche Modelle. Dabei behandelt es neben dem COSIMIR-Modell auch das Transformations-Netzwerk zur Heterogenitätsbehandlung und stellt so den Bezug zu Kapitel 5 her.

Kapitel 6 referiert die Experimente mit dem COSIMIR-Modell und dem Transformations-Netzwerk. Damit wird sowohl Adaptivität (COSIMIR) als auch Flexibilität (Transformations-Netzwerk) mit Backpropagation-Netzwerk modelliert und evaluiert. Ein Fazit schließt die Arbeit.

Zur leichteren Lesbarkeit und zur inhaltlichen Geschlossenheit der Kapitel wurden einige Redundanzen eingebaut. So spielen neuronale Netze (Kapitel 3) bereits im Überblick zu Information Retrieval (Kapitel 2) eine Rolle und für Heterogenitätsbehandlung (Kapitel 5) eignet sich ein adaptiertes COSIMIR-Modell (Kapitel 1). Die inhaltlichen Zusammenhänge zwischen den Gebieten erfordern diese Vorgriffe, bei denen einige Aspekte eines Bereichs bereits vor dem Kapitel dafür erscheinen.

Diese Arbeit schlägt eine Brücke zwischen zwei Wissensgebieten und ist als Lektüre für Fachleute und Praktiker aus den Bereichen Neuronale Netze als auch Information Retrieval ebenso geeignet wie für Studierende, die sich für diese Themen interessieren. Zahlreiche Querverweise ermöglichen auch einen themenorientierten Einstieg, von dem aus bei Bedarf auf andere Bereiche der Arbeit zugegriffen werden kann.

2 Grundlagen des Information Retrieval

Information Retrieval beschäftigt sich mit der Suche nach Information und mit der Repräsentation, Speicherung und Organisation von Wissen. Information Retrieval modelliert Informationsprozesse, in denen Benutzer aus einer großen Menge von Wissen die für ihre Problemstellung relevante Teilmenge suchen. Dabei entsteht Information, die im Gegensatz zum gespeicherten Wissen problembezogen und an den Kontext angepasst ist.

Das Informationsbedürfnis eines Benutzers ist der Ausgangspunkt eines Dialogs mit einem Informationssystem. Der Benutzer formuliert sein Informationsbedürfnis in der Benutzungsoberfläche eines Information Retrieval Systems. Das System vergleicht die Anfrage mit den im System vorhandenen Dokumenten oder deren Repräsentationen. Ein Teil der Dokumente, die aus Sicht des Systems gut zur Eingabe des Benutzers passen, wird dem Benutzer als Ergebnis vorgelegt. Das System sucht dazu die Dokumente, die sehr ähnlich zu der Anfrage sind. Der Benutzer kann nun entscheiden, ob die gefundenen Ergebnis-Dokumente für sein Problem relevant sind und es eventuell lösen oder nicht. Abbildung 2-1 zeigt den Ablauf eines Informationsprozesses.

Information Retrieval befasste sich in der bibliothekarischen und dokumentarischen Tradition lange vorwiegend mit Text-Dokumenten. Da Wissen überwiegend in der Form von Texten in natürlichen Sprachen vorliegt, betont das Information Retrieval nach wie vor die Suche in Textdatenbanken.

Bis in den 60er Jahren dominierte die Annahme, der Retrieval Prozess könne sowohl auf seiten der Repräsentation als auch beim Retrieval exakt abgebildet werden. Seit den 70er Jahren haben sich vage oder sogenannte *best match* Verfahren etabliert. Parallel vollzog sich eine Entwicklung, die ausgehend von der Sicht auf das System immer stärker den Benutzer in den Mittelpunkt rückte. Die Systemsicht auf IR befasst sich ausschließlich mit der Bearbeitung einer formalen Anfrage durch ein Informationssystem und ihrer mathematischen Modellierung. Eine holistische Sichtweise bezieht sowohl Fragen der Repräsentation von Dokumenten in einem IR-System mit ein und berücksichtigt den Benutzer im Kontext seines Informationsproblems und die Interaktion zwischen Benutzer und System. Diese Position wird oft als benutzerorientiert oder *cognitive viewpoint* (cf. Ingwersen 1992, Belkin 1993) bezeichnet.

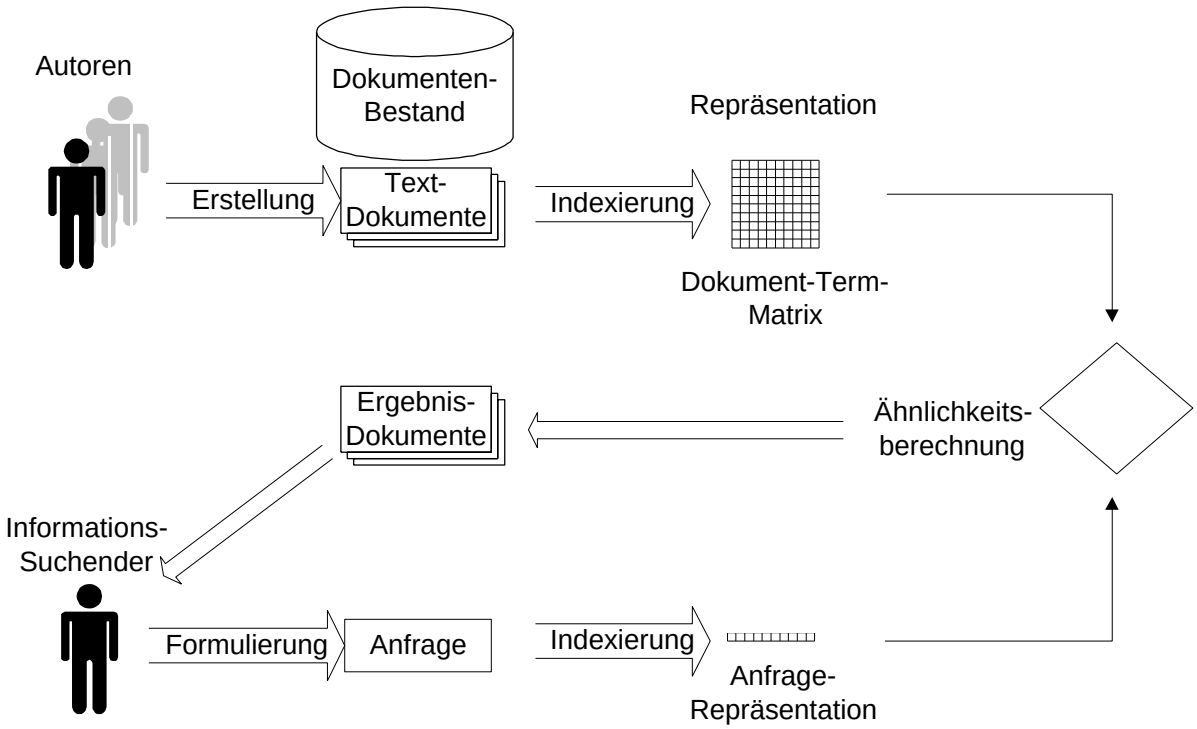


Abbildung 2-1: Der Information Retrieval Prozess

Diese und weitere Aspekte finden sich in der umfassenden Definition aus der Satzung der Fachgruppe Information Retrieval (cf. Fachgruppe IR 1996) in der Gesellschaft für Informatik. Demnach beschäftigt sich Information Retrieval

> „schwerpunktmäßig mit jenen Fragestellungen, die im Zusammenhang mit vagen Anfragen und unsicherem Wissen entstehen. Vage Anfragen sind dadurch gekennzeichnet, dass die Antwort a priori nicht eindeutig definiert ist. Hierzu zählen neben Fragen mit unscharfen Kriterien insbesondere auch solche, die nur im Dialog iterativ durch Reformulierung (in Abhängigkeit von den bisherigen Systemantworten) beantwortet werden können: häufig müssen zudem mehrere Datenbasen zur Beantwortung einer einzelnen Anfrage durchsucht werden. Die Darstellungsform des in einem IR-System gespeichertem Wissens ist im Prinzip nicht beschränkt (z.B. Texte, multimediale Dokumente, Fakten, Regeln, semantische Netze). Die Unsicherheit (oder die Unvollständigkeit) dieses Wissens resultiert meist aus der begrenzten Repräsentation von dessen Semantik (z.B. bei Texten oder multimedialen Dokumenten); darüber hinaus werden auch solche Anwendungen betrachtet, bei denen die

> gespeicherten Daten selbst unsicher oder unvollständig sind (wie z.B. bei vielen technisch naturwissenschaftlichen Datensammlungen). Aus dieser Problematik ergibt sich die Notwendigkeit zur Bewertung der Qualität der Antworten eines Informationssystems, wobei in einem weiteren Sinne die Effektivität des Systems in Bezug auf die Unterstützung des Benutzers bei der Lösung seines Anwendungsproblems beurteilt werden sollte.“ (Fachgruppe IR 1996).

Diese Definition stellt die Vagheit als zentrales Merkmal von Information Retrieval Prozessen in den Mittelpunkt. Die inhärente Vagheit des Suchprozesses und dessen adäquate Modellierung bilden demnach den Kern der Forschung im Information Retrieval. Die Frage, welche Methoden diese Vagheit adäquat modellieren, führt zum Paradigma des Soft-Computing, das Lofti Zadeh folgendermaßen definiert:

> "What is soft computing?
> Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty and partial truth. In effect, the role model for soft computing is the human mind. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost. ... At this juncture, the principal constituents of soft computing (SC) are fuzzy logic (FL), neural network theory (NN) and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, chaos theory and parts of learning theory. What is important to note is that SC is not a melange of FL, NN and PR. Rather, it is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. In this perspective, the principal contributions of FL, NN and PR are complementary rather than competitive." (Zadeh 1994)

Diese Position zeigt eine Abkehr von rein formal bestimmten Entwicklungen hin zu toleranten und an den Menschen angepassten Informationssystemen. Die formulierten Ziele von Soft-Computing entsprechen den Anforderungen des Information Retrieval in hohem Maße, so dass eine Anwendung von Soft-Computing Verfahren nahe liegt.

Die obige Definition von Information Retrieval betont die Rolle der Evaluierung, die beim Anwendungsproblem des Benutzers ansetzt. Bemerkenswert ist insbesondere die Offenheit bei den Daten, die mit IR erschlossen werden. Die traditionelle IR-Forschung beschäftigte sich überwiegend mit Text-Dokumenten, während z.B. das Retrieval von Faktendaten im Rahmen der Datenbankforschung behandelt wurde. Dort geht man meist davon aus, dass es ein exaktes Ergebnis zu einer Anfrage gibt. In der Praxis zeigt sich häufig, dass auch im Fakten-Retrieval vage Informationsbedürfnisse und unsicheres Wissen eine erhebliche Rolle spielen.

Die Bereiche Information Retrieval und Datenbanken nähern sich immer stärker an. Eine Konsequenz besteht z.B. darin, dass die Fachgruppen Information Retrieval und Datenbanken der Gesellschaft für Informatik 1999 erstmals eine Tagung gemeinsam ausgerichtet haben (cf. Düsterhöft 1999). Auch dieses Einführungskapitel zum Information Retrieval räumt dem Retrieval von Fakten großen Raum ein, wobei gerade vage Fragestellungen und Aspekte der Interaktion behandelt werden.

Die Integration multimedialer Dokumente verweist bereits auf das Problem der Heterogenität, da die verschiedenen Medien meist unterschiedlich repräsentiert werden und somit zunächst kaum vergleichbar sind. Der Rest des Kapitels ist nach den Objekt-Typen beim Retrieval unterteilt. Der folgende Abschnitt befasst sich mit dem klassischen Text-Retrieval und bespricht die Repräsentation, Modelle, Ähnlichkeitsberechnung, Evaluierung und zeigt ein Beispiel. Der zweite große Abschnitt 2.2 behandelt Fakten-Retrieval und hebt besonders die Unterschiede zum Text-Retrieval hervor. Als Beispiel für die Repräsentation unsicherer oder vager Fakten wird die Fuzzy Logik vorgestellt, die oft zur Erweiterung von Datenbankabfrage-Sprachen eingesetzt wird. Da Fakten-Retrieval in der Literatur meist nicht sehr ausführlich besprochen wird, räumt Abschnitt 2.2.3 den Beispielen großen Raum ein.

Neben Text und Fakten entstehen immer mehr Information Retrieval Systeme für spezifische Anwendungsfälle. Wechsler 1995 und Schäuble 1997 stellen Audio-Retrieval Systeme für gesprochene (Speech-) Dokumente vor, in denen Phonemsequenzen das Indexierungsvokabular darstellen. Daneben gibt es Ansätze für das Retrieval von Bildern, bei denen Farben, Texturen, Formen und räumliche Anordnung eine Rolle spielen (cf. Gupta/Jain 1997, Del Bimbo 1999). Bei spezialisierten Retrieval-Verfahren für bestimmte Arten von Grafiken oder Bildern können spezifische Verfahren eingesetzt werden und so die Qualität verbessern. Ein Beispiel für Firmenlogos bieten Wu et al. 1996. Einen Überblick über das Retrieval von Videosequenzen liefern Agrain et al. 1996.

2.1 Text-Retrieval

Die klassische Aufgabe von Information Retrieval besteht in der Suche nach Text-Dokumenten. Eine klassische Einführung in IR bieten Salton/McGill 1983, eine aktuelle Einführung mit Schwerpunkt auf Algorithmen findet sich in Baeza-Yates/Ribeiro-Neto 1999. Eine themenorientierte Einführung liegt mit Fuhr et al. 1998 vor.

2.1.1 Indexierung und Gewichtung

Um Texte recherchieren zu können, werden sie zunächst indexiert. Allgemein gesprochen werden Repräsentationen der Retrieval-Objekte erstellt. Textdokumente werden im Information Retrieval durch Mengen von Begriffen oder Termen repräsentiert. Obwohl dadurch die Semantik natürlicher Sprache nicht adäquat abgebildet wird, ist dies nach wie vor die einzige realisierbare Repräsentationsform für große Datenmengen. Ein Beispiel für einen Ansatz mit semantischer Analyse für eine beschränkte Datenbasis bietet Haenelt 1996.

Für die Indexierung von großen Textmengen gibt es zwei Verfahren:

- intellektuelle (oder manuelle) Indexierung
  Ein menschlicher Indexierer evaluiert das Objekt, das in die Datenbasis eingefügt wird. Er vergibt dazu inhaltskennzeichnende Schlagwörter, die meist aus einem kontrolliertem Vokabular stammen.
- automatische (maschinelle) Indexierung
  Ein Algorithmus analysiert das Objekt und extrahiert aus Text-Dokumenten alle vorkommenden Wörter (außer besonders häufig vorkommenden Stoppwörtern), reduziert eventuell morphologische Formen und speichert die Vorkommenshäufigkeit der Wörter.

Die maschinelle Indexierung versucht nicht, den Text zu verstehen und semantisch zu modellieren. Statt dessen werden Begriffe gewählt, die einen Text repräsentieren, ohne deren Zusammenhang zu analysieren.

Die Repräsentation durch Schlagwörter oder Terme benutzen auch Retrieval Systeme für andere Objekttypen. Viele Bild-Retrieval-Systeme arbeiten mit Termen bzw. Schlagwörtern, die den Bildern intellektuell zugewiesen werden. Zur Zeit werden im Text-Retrieval manuelle und automatische Indexierungsmethoden parallel eingesetzt (für eine Diskussion der Vor- und Nachteile cf. Krause/Mutschke 1999). Die Hauptunterschiede der entstehenden Repräsentationen bestehen darin, dass die Menge der Terme bei automatischer Indexierung sehr viel größer ist und ein manuell vergebener Begriff nicht notwendigerweise in dem Dokument vorkommt. Besonders für die manuelle Indexierung ist es üblich, das Indexierungsvokabular zu begrenzen,

und den Indexierer zu zwingen, die Terme aus einer Schlagwortliste, einer Klassifikation oder einem Thesaurus zu wählen. Dadurch wird z.B. vermieden, dass mehrere Indexierer unterschiedliche Synonyme für einen Term verwenden. Diese Form der inhaltlichen Erschließung sorgt für eine gewisse Konsistenz (cf. Buder et al. 1997; cf. auch 5.2.1).

Die automatische Indexierung benutzt nicht alle Wörter, die in einem Text vorkommen, selbst wenn kein kontrolliertes Vokabular vorgegeben ist. Zunächst werden sehr häufige Wörter und Funktionswörter wie Artikel, Präpositionen, die isoliert betrachtet semantisch nicht relevant sind, ausgefiltert. Diese Wörter liegen in einer Stoppwortliste vor, die natürlich von Sprache zu Sprache unterschiedlich ist und sogar anwendungsabhängig sein kann. Daneben greifen automatische Indexierungsverfahren mehr oder weniger auf linguistische Verfahren zurück. Grundformreduktion und Kompositazerlegung sind die wichtigsten Schritte.

Nach diesen Vorarbeiten besteht die automatische Indexierung in dem Zählen der Wörter im Text. Zwischenergebnis ist die Häufigkeit aller Terme in allen Dokumenten. Die Häufigkeit der Terme in der Kollektion ist dabei meist sehr unterschiedlich. Da bei der Suche sehr häufig vorkommende Terme keine guten Ergebnisse erzielen, geht die Häufigkeit eines Terms in der Kollektion als wichtiges Element in die Gewichtung mit ein. Die inverse Dokument-Häufigkeit (*inverse document frequency*, idf) wird meist mit dem Logarithmus berechnet. Der Logarithmus mindert die Effekte extremer Werte ab und sorgt dafür, dass z.B. ein Term, der 2000 mal in einem Dokument vorkommt, nicht ein doppelt so hohes Gewicht erhält, wie ein Term, der 1000 mal vorkommt. Dieser Unterschied ist bei weitem nicht so relevant wie der Unterschied zwischen einer Frequenz von eins und 100. Folgende Formel für die Berechnung der idf zeigt dies beispielhaft:

$$idf_i = \log \frac{Anzahl\ der\ Dokumente\ in\ der\ Kollektion}{Anzahl\ der\ Dokumente\ mit\ Term_i}$$

Baeza-Yates/Ribeiro-Neto 1999:29

Die Häufigkeit des Terms in der Kollektion wird meist als die Anzahl der Dokumente bestimmt, in denen der Term vorkommt. Daneben gibt es auch die Möglichkeit, alle Token zu berücksichtigen und die Frequenz des Terms in der Kollektion zugrunde zu legen. Das Endergebnis der automatischen Indexierung und der Gewichtung ist ein Gewicht für jeden Term in jedem Dokument.

2.1.2 Modelle

Die folgenden Abschnitte führen in die wichtigsten Modelle im Information Retrieval ein. Ein ausführlicher Überblick, der auch zahlreiche weniger gebräuchliche Modelle umfasst, findet sich in Womser-Hacker 1997 und Baeza-Yates/Ribeiro-Neto 1999.

2.1.2.1 Das Boolesche Modell

Das Boolesche Retrieval ist das älteste Information Retrieval Modell. Es hat noch in den 80er und teilweise in den 90er Jahren den Markt für IR Systeme beherrscht. Das Boolesche Modell betrachtet IR als eine Mengenoperation mit dem Ziel, die Menge der relevanten Dokumente in einer Grundmenge zu finden. Es beruht auf der elementaren Mengenlehre und operiert im Wesentlichen mit den Operationen Schnittmenge, Vereinigungsmenge und Komplementärmenge. Das Boolesche Modell geht davon aus, dass das Ergebnis einer Suchanfrage als exakte Menge von Dokumenten bestimmt werden kann.

Die elementaren Mengen sind exakt über die Deskriptoren oder Terme definiert. So bilden alle Dokumente, die mit dem Term A indexiert sind, eine elementare Menge. Dies zeigt bereits, dass Boolesche Systeme nicht mit gewichteten Repräsentationen arbeiten, da in der klassischen Mengenlehre eine Element zu einer Menge gehört oder nicht. Eine Lockerung dieser strikten Logik erlaubt die Fuzzy Set Theory (cf. Abschnitt 2.2.1), auf der auch einige IR-Systeme basieren. Die Anfrage in einem Booleschen Modell besteht aus einem syntaktisch gültigen Ausdruck aus elementaren Mengen und Mengenoperatoren.

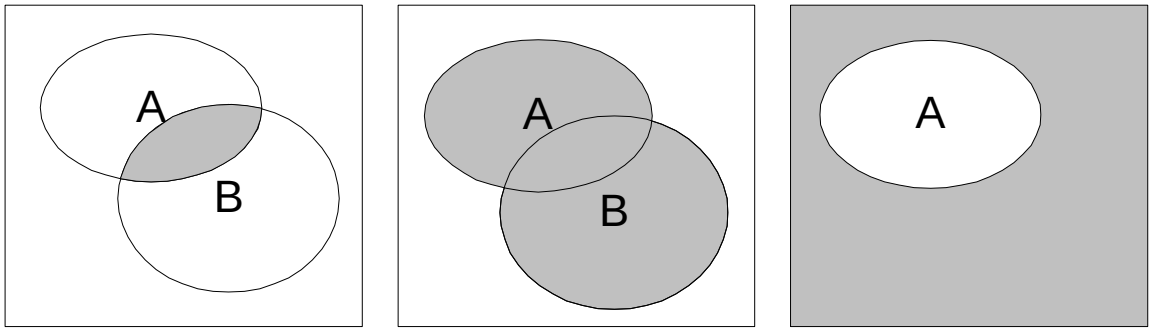


Abbildung 2-2: Die elementaren Operationen im Booleschen Retrieval-Modell in Venn-Diagrammen

Die wichtigsten Schwächen des Booleschen Modells fasst z.B. Cooper 1988 zusammen:

- Strikte Einteilung des Dokumentenbestands
  Der Dokumentenbestand wird in zwei Mengen unterteilt (relevant und nicht relevant), wobei der Benutzer die Größe der einzelnen Mengen nur schwer steuern kann. Zwar ist die Größe der Ergebnismenge nicht das primäre Ziel, aber aus pragmatischen Gründen sollte die Menge weder zu groß noch zu klein sein.
- Die relevanten Dokumenten werden ungeordnet präsentiert
  Die Ergebnismenge wird nicht weiter geordnet, so dass der Benutzer prinzipiell die gesamte Menge betrachten muss.
- Die Mengenlogik ist für Benutzer schwierig anzuwenden
  Trotz der scheinbaren Einfachheit der Operatoren zeigt sich in Benutzertests immer wieder, dass die meisten Benutzer die Operatoren nicht richtig interpretieren. Selbst erfahrenen Rechercheuren unterlaufen in Tests teilweise Fehlinterpretationen. Die Mengenlehre entspricht also offensichtlich nicht dem menschlichen Denken.

Um diese Nachteile auszugleichen, werden bereits seit langem alternative Modelle diskutiert. Fast alle nicht-Booleschen Modelle sind Ranking-Modelle, die eine strikte Teilung der Dokumente in relevant und nicht relevant vermeiden. Dieser Nachteil Boolescher Systeme wirkt sich besonders bei sehr großen Kollektionen negativ aus. Dies ist ein Grund, dass seit der starken Verbreitung des Internet die Ranking-Modelle nach einer Phase der vorwiegend wissenschaftlichen Entwicklung in die kommerzielle Entwicklung vordringen. Praktisch alle Internet-Suchmaschinen (für einen Überblick cf. Mönnich 1999) bieten als Default-Modell ein Ranking-Verfahren an und auch die wichtigen Retrieval-Systeme bieten inzwischen zumindest zusätzlich ein Ranking-Verfahren an.

2.1.2.2 Das Vektorraum-Modell

Ranking-Modelle berechnen für eine Anfrage jedem Dokument eine Relevanz, die die Grundlage des Ranking bildet. Diese Zahl wird auch Retrieval Status Value (RSV) oder System-Relevanz genannt. Nach dieser Relevanz ordnet das System die Dokumente und der Benutzer findet so die relevantesten Dokumente zu Beginn der Ergebnisliste.

Das Vektorraum-Modell beschreiben Salton/McGill 1983 ausführlich. Es bildet zugleich auch ein Meta-Modell für IR, da die meisten anderen Modelle

damit kompatibel sind und sich im Kontext des Vektorraum-Modell formulieren lassen. Selbst das Boolesche Modell lässt sich als Spezialfall des Vektorraum-Modells betrachten.

Das Vektorraum-Modell nimmt eine geometrische Sichtweise auf ein Information Retrieval System ein. Die Dokumente sind demnach Punkte in einem vieldimensionalen Koordinaten-System, dessen Achsen die Terme repräsentieren. Formal kann ein Punkt in einem Koordinaten-System immer auch als ein Vektor vom Nullpunkt zu diesem Punkt interpretiert werden. Die Betrachtungsweise als Vektoren hat Eingang in den Namen des Modells gefunden. Auch die Anfragen interpretiert das Vektorraum-Modell als Vektoren oder Punkte im Term-Raum. Dokumente die zu einer Anfrage passen, sind im Vektorraum-Modell Punkte, die nahe nebeneinander liegen. Die Ähnlichkeit ergibt sich also aus der räumlichen Nähe bzw. Distanz der Punkte. Einige Ähnlichkeitsfunktionen messen die Richtung der Vektoren anhand des Winkels zwischen ihnen.

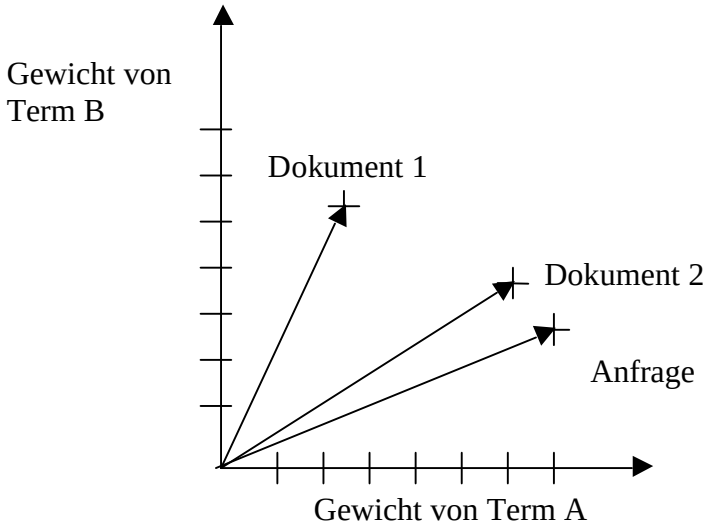


Abbildung 2-3: Zweidimensionaler Vektorraum mit zwei Dokumenten.

Abbildung 2-3 veranschaulicht das Prinzip des Vektorraum-Modells für ein Modell mit zwei Dimemsionen. Die zwei Terme A und B formen ein Koordinatensystem mit zwei Dimensionen. Darin liegen die beiden Dokumente als Punkte, die sich auch als Vektoren vom Ursprung des Koordinatensystems zu den Punkten betrachten lassen. Jedes Dokument und jede Anfrage erhält an den Term-Achsen das Gewicht für diesen Term zugewiesen.

Für die Berechnung der Ähnlichkeit von Anfrage und Dokumenten existieren zahlreiche verschiedene Formeln. Neben den Distanzmaßen wird häufig der

Kosinus als Maß für den Winkel zwischen zwei Vektoren benutzt. Abschnitt 2.1.3 vertieft den Aspekt der Ähnlichkeitsberechnung. In dem Beispiel in Abbildung 2-3 liegt die Anfrage näher bei Dokument 2, das damit ähnlicher zur Anfrage ist als Dokument 1.

Das Vektorraum-Modell ist ein einleuchtendes Modell für den Information Retrieval Prozess; allerdings ist die Anschaulichkeit stark eingeschränkt. Für den Menschen sind nur drei Dimensionen vorstellbar, während selbst ein kleiner Dokumentenbestand wesentlich mehr Terme enthält und damit mehr Dimensionen erfordert.

Innerhalb des Vektorraum-Modells lassen sich Strategien zur Modellierung der Interaktivität des Retrieval-Prozesses integrieren. Eine wichtige und erfolgreiche Strategie, um die Qualität eines Retrievalergebnisses zu erhöhen, ist Relevanz-Feedback (cf. Harman 1992). Dabei beurteilt der Benutzer eine Teilmenge von Dokumenten und weist ihnen einen Relevanz-Wert zu. Das System nutzt diese Einschätzung, indem es die beurteilten Dokumente analysiert und davon ausgehend die Anfrage modifiziert. Die Terme der positiv eingeschätzten Dokumente werden stärker gewichtet bzw. kommen zur Anfrage hinzu. Die Terme der negativ beurteilten Dokumente werden entsprechend schwächer gewichtet. Auch Relevanz-Feedback kann mit der räumlichen Metapher des Vektorraum-Modells interpretiert werden. Die Anfrage wird im Raum in Richtung der relevanten Dokumente verschoben.

2.1.2.3 Das probabilistische Modell

Ein weiteres Ranking-Modell ist das inzwischen weit verbreitete probabilistische Information Retrieval Modell. Der Ähnlichkeitswert zwischen Anfrage und Dokument oder die Retrieval Status Value (RSV) wird dabei als Wahrscheinlichkeit für die Relevanz eines Dokuments interpretiert. Die Verwendung der Wahrscheinlichkeitsrechnung betont die mit dem Retrievalprozess verbundene Unsicherheit.

Die Wahrscheinlichkeit bezieht sich auf ein Paar von Objekten, das in der Regel aus einer Anfrage und einem Dokument besteht. Als Grundlage für die Bestimmung dienen die bei der Indexierung bestimmten Vorkommenshäufigkeiten der Terme, die als Wahrscheinlichkeit für das Vorkommen eines Terms in einem Dokument gewertet werden. Diese bekannten Wahrscheinlichkeiten für das Vorkommen der Terme sind die Bedingung für die Relevanz eines Dokuments. Im probabilistischen Modell spielt dementsprechend die bedingte Wahrscheinlichkeit eine große Rolle. Da die Wahrscheinlichkeit für die Relevanz nicht direkt bestimmbar ist, wird sie indirekt über die dafür vorliegende Evidenz berechnet. Die folgende Formel drückt die bedingte Wahrscheinlich-

keit für die Relevanz eines Dokuments aus, wobei die Bedingung das Vorkommen des Terms T ist:

$$P(R \mid T) = \frac{P(T \mid R)P(R)}{P(T)}$$

cf. Spies 1993:40

Auf der rechten Seite der obigen Formel treten zwei schwer bestimmbare Größen auf. So kann die Wahrscheinlichkeit für die Relevanz eines Dokuments nicht ohne die vollständige Durchsicht eines Dokumentenbestandes bestimmt werden. Entsprechend ist auch der erste Teil des Zählers, die bedingte Wahrscheinlichkeit für das Vorkommen des Terms unter der Bedingung der Relevanz dieser Dokumente, nicht einfach zu bestimmen. Dabei handelt es sich um die Häufigkeit des Vorkommens des Terms in relevanten Dokumenten. Lediglich die Wahrscheinlichkeit für das Vorkommen des Terms im Nenner ergibt sich eindeutig aus seiner Häufigkeit in der Kollektion, die während der Indexierung bestimmt wird.

Analog wird die Wahrscheinlichkeit für die Nicht-Relevanz berechnet, die sich mit der Relevanz zu Eins summiert. Ist die Wahrscheinlichkeit für Relevanz größer als die für Nicht-Relevanz, so gilt das Dokument als relevant.

Da immer das Vorkommen mehrerer Terme Anhaltspunkte für die Relevanz bietet, wird die Evidenz kombiniert. Die Berechnungsgrundlage liefert die Bayes'sche Regel:

$$P(R \mid T_1 \cap T_2) = \frac{P(T_1 \mid R \cap T_2)P(R \mid T_2)}{P(T_1 \mid T_2)}$$

cf. Womser-Hacker 1997:28

Im Nenner erscheint die Wahrscheinlichkeit des Vorkommens von Term T_1 in den Dokumenten, in denen Term T_2 vorkommt. Um diese Formel für alle Terme zu berechnen, muss paarweise die Abhängigkeit der Terme bestimmt werden. Um diese komplexe und aufwendige Berechnung zu vermeiden, nehmen die Modelle meist die paarweise Unabhängigkeit zwischen den Termen an. Die Annahme ist unrealistisch, da semantisch verwandte Begriffe häufiger gemeinsam in Texten vorkommen als semantisch nicht in Beziehung stehende Terme. Dieses Problem diskutiert z.B. Cooper 1991. Die Herleitung des probabilistischen Modells findet sich in van Rijsbergen 1979. Die Grundlagen wie etwa bedingte Wahrscheinlichkeit stellt Spies 1993 ausführlich vor.

Auf der Ebene der Implementierung sind das Vektorraum-Modell und das probabilistische Modell oft nicht klar zu unterscheiden. Beim Relevanz-Feedback allerdings zeigt sich die Stärke des probabilistischen Modells deutlich. Während sonst alle Terme gleich behandelt werden, bieten die Relevanz-Feedback-Entscheidungen des Benutzers bessere Evidenz. Die in den relevanten Dokumenten vorkommenden Terme bedingen Relevanz mit einer höheren Wahrscheinlichkeit als Terme, die in den nicht relevanten Dokumenten vorkommen.

2.1.2.4 Modelle mit komprimierten Repräsentationen

Die bisher vorgestellten Modelle erzeugen aufgrund der im Text-Retrieval vorliegenden Massendaten sehr große und spärlich besetzte Vektorräume oder Matrizen. Große Datenbanken enthalten Hunderttausende oder gar Millionen von Dokumenten. Durch die automatische Indexierung entstehen sehr große Matrizen, da jede vorkommende Wortform zum Term werden kann. Spärlich besetzt bedeutet, dass prozentual nur wenige der Zellen in der Dokument-Term-Matrix einen Wert ungleich Null besitzen. Dies ist unvermeidlich, da Dokumente die meisten Terme nicht enthalten. Verglichen mit ihrer Größe enthält die Matrix relativ wenig Wissen. Der Umgang mit diesen großen Matrizen führt zu technischen Schwierigkeiten und dem Bedarf nach Steigerung der Effizienz. Als Reaktion auf diese Problematik, versuchen einige Verfahren, die Dimensionalität der Repräsentation zu reduzieren.

Dazu zählen z.B. der Kontext-Vektor in MatchPlus (cf. Gallant et al. 1993/4, cf. auch Abschnitt 4.8), Latent Semantic Indexing (cf. Dumais 1994) und der Ansatz von Gauch/Wang 1997, bei dem ein Term durch die Terme in seiner näheren Umgebung repräsentiert wird.

Die starke Reduktion der Merkmalsräume ist auch die Grundlage für viele Visualisierungen im Information Retrieval. Zahlreiche Ansätze reduzieren die Dokument-Term-Matrix auf eine darstellbare Anzahl von Dimensionen, also auf zwei oder drei. In den letzten Jahren erfolgten insbesondere eine Reihe von Implementierungen auf der Basis von Selbstorganisierenden Kohonen-Netze mit großen und realen Datenmengen (cf. Abschnitt 4.4).

2.1.2.4.1 Kontext-Vektor

Der Kontext-Vektor (cf. z.B. Gallant et al. 1993) als Repräsentationsform für Wissen stammt aus der Künstlichen Intelligenz. Das Verfahren bildet alle Terme intellektuell auf einen Vektor von Basiseigenschaften ab.

Waltz/Pollack 1985 setzen den Kontext-Vektor zur Disambiguierung von sprachlichen Äußerungen ein. Der Kontext-Vektor bildet den Kontext eines Satzes ab und ermöglicht so die eindeutige Zuordnung zu einer Bedeutung.

Das Kontext-Vektor-Verfahren entspricht der Indexierung der Terme mit restringiertem Vokabular. Das Indexierungsvokabular sind Basiseigenschaften, die versuchen, alle semantischen Werte einer Sprache auszudrücken. Dieses Vokabular wird intellektuell bestimmt und umfasst gewissermaßen semantische Atome, die den Sinn eines Wortes konstituieren. Bei der Indexierung wird zunächst in der Regel automatisch das gesamte originale Indexierungsvokabular gewonnen. Damit liegen alle Terme vor, die den Inhalt der Dokumente beschreiben wie in einem Standard IR-Modell. Der entscheidende zweite Schritt weist jedem Term intellektuell einen Vektor mit Gewichten für die Basiseigenschaften zu.

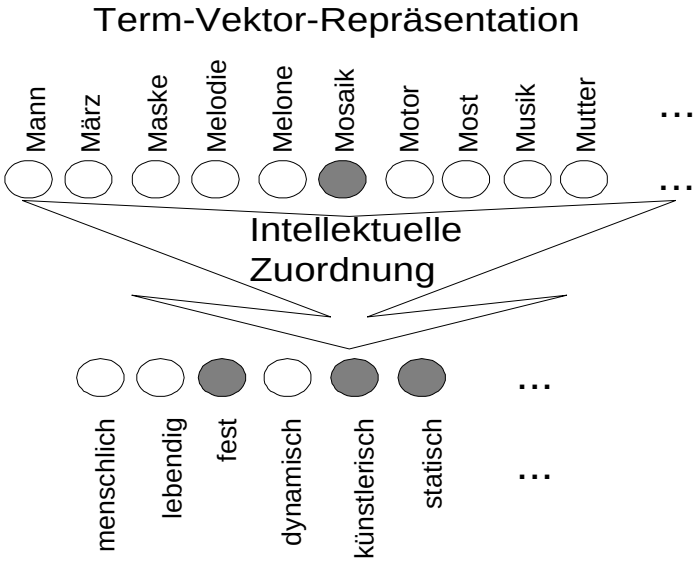


Abbildung 2-4: Der Term *Mosaik* wird auf einen Kontext-Vektor abgebildet

Abbildung 2-4 zeigt dies beispielhaft für den Term *Mosaik*. Er erhält die Basiseigenschaften *fest, künstlerisch* und *statisch*. Die anderen Basiseigenschaften treffen für den Term *Mosaik* nicht zu.

Diese menschliche Zuordnung fällt nur einmal für einen Bestand von Termen an. Auch wenn neue Terme hinzukommen, wird die Zuordnung in der Regel nur für die neuen Terme nachgeholt.

Aus den nun festgelegten Beziehungen zwischen Dokumenten und Termen einerseits und Termen und Kontext-Vektor-Elementen andererseits ergibt sich die Beziehung zwischen Dokumenten und Kontext-Vektor-Elementen. Dazu

werden für ein Dokument die Kontext-Vektoren aller Terme addiert. So entsteht eine Repräsentation aller Dokumente durch einen gewichteten Vektor von Basiseigenschaften. Dieser entstehende Vektor ist wesentlich kürzer als der ursprüngliche, der für alle vorkommenden Terme ein Element besitzt. Die lokale Repräsentationsform, bei der ein Term ein Element eines Vektors darstellt, wird so durch eine stark verteile Repräsentation ersetzt, bei der ein Term über mehrere Eigenschaften verteilt ist.

Die Erstellung des Kontext-Vektors ist symbolisch interpretierbar und nachvollziehbar. Ein großer Nachteil besteht darin, dass der Vektor für jeden Term manuell erstellt wird. Gallant et al. 1993 und 1994 berichten zwar von einer Automatisierung, doch der firmeninterne Algorithmus ist nicht offengelegt. Während der Kontet-Vektor hohen manuellen Aufwand erfordert, arbeiten die folgenden Verfahren völlig automatisch.

2.1.2.4.2 Komprimierung mit neuronalen Netzen

Ein neuronales Backpropagation-Netzwerk komprimiert beliebige mehrdimensionale Merkmalsräume bei geeigneter Wahl der Architektur (cf. Merkl 1995). Backpropagation-Netzwerke sind lernende Verfahren, die Funktionen von einem Merkmalsraum in einen anderen aufgrund von Trainingsbeispielen nähern. Nach dem Lernen bildet die Funktion dann auch bisher unbekannte Daten erfolgreich ab. Neuronale Netze und den Backpropagation-Algorithmus führt Kapitel 3 ein. Dieser Abschnitt geht kurz auf die spezifische Architektur zur Komprimierung ein.

Das Backpropagation-Netzwerk zur Komprimierung lernt, den Input im Output zu reproduzieren. Dazwischen liegt eine versteckte Schicht mit wesentlich weniger Elementen als in den Original-Daten. In dieser versteckten Schicht entsteht beim Training eine reduzierte Repräsentation, aus der der Gesamtvektor annäherungsweise gewonnen wird. Die reduzierte Repräsentationist Grundlage des Retrievals. Merkl 1995 verkürzte mit diesem Ansatz die Trainingszeit für ein neuronales Netz, das eine Klassifikation von Dokumenten erstellt (Kohonen Self-Organizing Map, cf. Abschnitt 3.5.1). Das System von Merkl 1995 bildet Cluster von Software-Klassen und soll dadurch die Wiederbenutzbarkeit von Software-Code verbessern. Die Dokumente werden von 489 Termen repräsentiert, wobei es sich in diesem Fall vorwiegend um Befehle einer Programmiersprache handelt. Die 489 Terme komprimierte der Autor auf 30 bzw. 75 und erreichte damit eine erhebliche Reduktion der Trainingszeit der SOM. Zur Qualität der Clusterbildung merkt der Merkl 1995 nur an, im Wesentlichen hätten sich mit der reduzierten Repräsentation die gleichen Cluster gebildet.

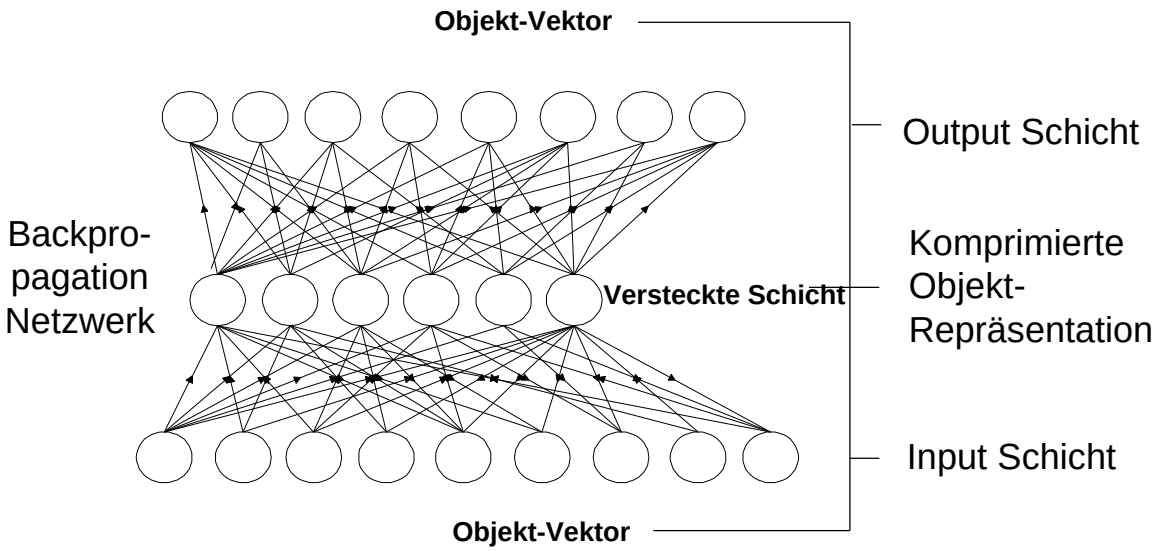


Abbildung 2-5: Architektur eines neuronalen Netzes zur Komprimierung von Dokument-Repräsentationen

Auch das Kohonen-Netzwerk reduziert die Dimensionalität der Eingangsdaten. Der Input wird derart auf eine meist zweidimensionale Karte abgebildet, dass ähnliche Muster nahe zusammenliegen. Kohonen-Netze stellt Abschnitt 3.5.1 vor, Anwendungen im IR finden sich in Abschnitt 4.4.

2.1.2.4.3 Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) komprimiert die Dokument-Term-Matrix und nutzt dazu das mathematische Verfahren Singular Value Decomposition (SVD). Dabei wird der ursprüngliche Termraum auf in der Regel zwichen 100 und 300 Variablen reduziert. LSI hat in empirischen Tests insbesondere im Rahmen der TREC-Konferenz (cf. Abschnitt 2.1.4.2) positive Ergebnisse erzielt (Dumais 1994). LSI wird in Deerwester et al. 1990, Berry 1993, Berry et al. 1995, Letsche 1996, Syu et al. 1996, Gordon/Dumais 1998 und Letsche/Berry 1997 beschrieben, einen Überblick über Singular Value Decomposition bieten Berry 1992 und Berry et al. 1993.

Latent Semantic Indexing arbeitet ähnlich wie eine Faktorenanalyse (z.B. *principal component analysis*), wie sie oft bei der statistischen Datenanalyse benutzt wird (cf. Rodeghier 1997:174ff.). Für die Erstellung einer Faktorenanalyse wird zunächst eine Korrelationsmatrix aller Merkmale erstellt, die analysiert, wie stark die beteiligten Merkmale korrelieren. Im ersten Schritt bestimmt die Faktorenanalyse wieviel jede Variable einer Matrix zur Erklä-

rung der Varianz in einer Korrelationsmatrix beiträgt. Im zweiten Schritt wählt der Anwender die *n* wichtigsten Variablen in der originalen Matrix. Die Faktorenanalyse fasst dann alle Variablen zu *n* Faktorengruppen oder Hintergrundvariablen zusammen, wobei der Einfluss zueinander ähnlicher Faktoren in einer Gruppe gebündelt wird. Nachträglich lässt sich der Anteil jedes originalen Faktors an einer Hintergrundvariablen bestimmen und auch visualisieren (cf. Rodeghier 1997:174ff.).

Latent Semantic Indexing produziert zueinander orthogonale oder rechtwinklige Faktoren. Dadurch bilden diese wieder ein Koordinatensystem. Die entstehenden Eigenschaften oder LSI-Dimensionen lassen sich nicht inhaltlich interpretieren, noch lässt sich ein Original-Term eindeutig einer Dimension zuordnen.

Latent Semantic Indexing nutzt aus, dass es sich bei der Dokument-Term-Matrix um eine spärlich besetzte Matrix handelt, bei der nur ein kleiner Teil der Zellen mit einem Gewicht belegt ist und die somit relativ wenig Information enthält. Die vorhandene Informationmenge fasst auch eine kleinere Matrix. Durch das mathematische Verfahren der Single Value Decomposition wird aus der ursprünglichen eine neue und kleinere Matrix gewonnen. Dabei werden die Singular Values und zwei weitere Matrizen bestimmt, von denen eine einen reduzierten Term-Raum und die andere einen reduzierten Dokumenten-Raum darstellt. Die zweite Matrix stellt eine komprimierte Beschreibung der Terme durch die Dokumente dar und. Aus der reduzierten Form kann die ursprüngliche vollständige Repräsentation wieder gewonnen werden. Die Berechnung erfolgt nach der Formel in der folgenden Abbildung:

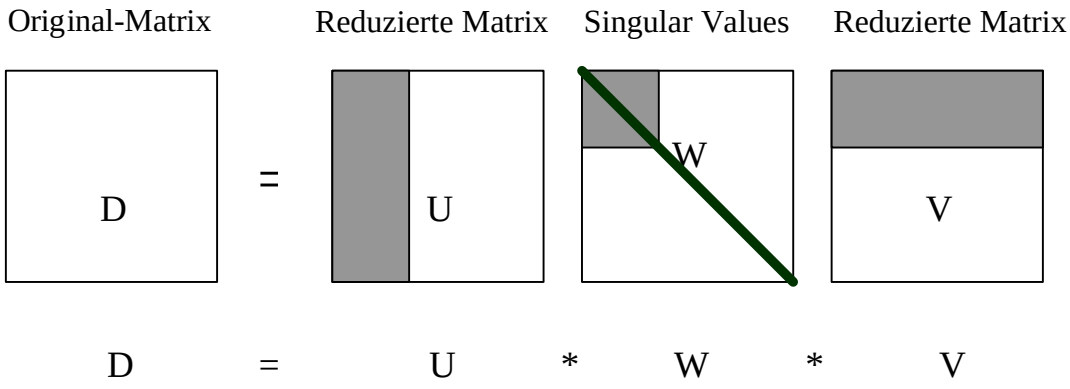


Abbildung 2-6: Schematische Darstellung der Reduktion mit Latent Semantic Indexing (cf. Syu et al. 1996)

Die Matrix W enthält die Singular Values, die alle auf der Diagonale liegen. Ihre stetig fallenden Werte sind ein Maß für die Wichtigkeit dieser Singular Value für die gesamte Matrix. Wichtigkeit bedeutet hier, wieviel diese Di-

mension zur Erzeugung der originalen Matrix beiträgt. Alle Singular Values reproduzieren die ursprüngliche, nicht komprimierte Matrix. Außer der Matrix W entstehen die Matrizen U und V, wobei U einen komprimierten Term-Raum mit allen Dokumenten enthält, während V alle Terme mit weniger (Pseudo-) Dokument-Profilen beschreibt. Die Größe der beiden Matrizen ergibt sich zum einen aus der ursprünglichen Matrix und zum anderen aus der Anzahl der verwendeten Singular Values und damit der LSI-Dimensionen.

Hinter dem Einsatz von Latent Semantic Indexing im Information Retrieval steht die Überlegung, dass unwichtige Dimensionen sehr wenig zur Semantik der Kollektion beitragen und bei der Berechnung der Ähnlichkeiten nur störend wirken. Zahlreiche weniger wichtige Dimensionen werden weggelassen und nur die ca. 100 bis 300 (cf. Berry 1992:4, Berry et al. 1995) wichtigsten repräsentieren die Dokumente. Da die LSI-Dimensionen letztlich komplexe Kombinationen realer Terme darstellen, wird durch das Weglassen der weniger dominanten Dimensionen der Einfluss der Terme auf das Ergebnis gewichtet. Dabei ist anzunehmen, dass der Einfluss vieler Terme sehr gering wird. Es entsteht eine gleichmäßiger verteilte Repräsentation wie bei allen Reduktionsverfahren. Im eigentlichen Retrievalprozess ersetzt die reduzierte Matrix über Dokumente und LSI-Dimensionen die ursprüngliche Dokument-Term-Matrix. Die Anfragen werden nachträglich mit den gleichen Singular Values bearbeitet und so gewissermaßen im reduzierten Raum platziert. Beim Vergleich zwischen Dokument und Anfrage werden die Ähnlichkeiten zwischen reduzierten Vektoren berechnet.

Diese Grundannahme hinter Latent Semantic Indexing ist nicht völlig plausibel, da auch ein Faktor oder Term, der wenig zur gesamten globalen Struktur der Matrix beiträgt, für einzelne Anfragen ausschlaggebend sein kann. Trotzdem zeigt LSI in empirischen Untersuchungen gute Resultate und hat sich auch für Massendaten bewährt.

Das folgende Beispiel veranschaulicht die Berechnung der neuen Matrix anhand einer Menge von vier Dokumenten mit sechs Termen. Es wurde mit einer experimentellen Software der Firma Bellcore berechnet, die auf dem SVD-Algorithmus von Lanczos beruht (cf. Berry et al. 1993). Dieser Algorithmus entspricht dem in TREC verwendeten. Tabelle 2-1 zeigt die ursprüngliche Dokument-Term-Matrix. Das LSI-Verfahren berechnet dafür die folgenden vier Singular-Values: 1,682 1,507 0,907 0,268. Die folgende Tabelle 2-2 zeigt die reduzierte Matrix, die sich daraus ergibt.

Tabelle 2-1: Beispiel für eine kleine Dokument-Term-Matrix

	Term1	**Term2**	**Term3**	**Term4**	**Term5**	**Term6**
Dok1	1	1	2			
Dok2		2	1	1		
Dok3				1	1	1
Dok4		1				

Tabelle 2-2: Beispiel: die mit LSI reduzierte Matrix

	LSI-Dim1	**LSI-Dim2**	**LSI-Dim3**	**LSI-Dim4**
Dok1	-0,435	0,724	0,533	0,025
Dok2	-0,493	0,281	-0,772	-0,283
Dok3	-0,747	-0,620	0,230	0,058
Dok4	-0,089	0,101	-0,256	0,956

In diesem Raum lassen sich nun auch wieder die Terme darstellen. Dabei zeigt sich, wie Latent Semantic Indexing die Verteiltheit der Repräsentation erhöht. Während in der originalen Matrix sowohl Term- als auch Dokument-Vektoren hauptsächlich aus Nullen bestehen und die Information in wenigen Einsen lokalisiert ist, verteilt sie sich bei der LSI-Matrix über alle Dimensionen. Anders ausgedrückt verteilt sich die vorher lokale Repräsentation eines Terms auf mehrere andere Dimensionen.

Durch Berücksichtigung von nur zwei oder drei der Dimensionen aus der Tabelle 2-2 erhält man eine noch stärker reduzierte Matrix. Die zweidimensionale Matrix lässt sich grafisch darstellen. Die vier Dokumente sind in Abbildung 2-7 vier Punkte in einem zweidimensionalen Koordinatensystem.

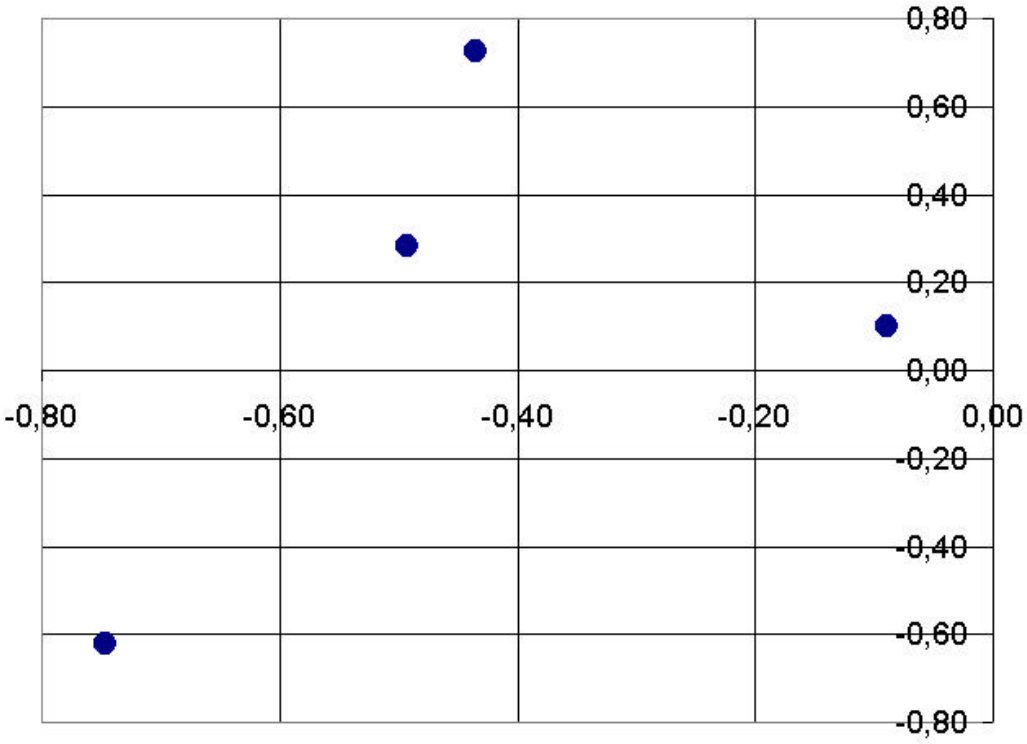


Abbildung 2-7: Die vier Dokumente aus Tabelle 2-1 und Tabelle 2-2 in einem zweidimensionalen LSI-Raum

Dieses Beispiel zeigt, dass sich Latent Semantic Indexing bei einer sehr starken Reduktion auch für die Visualisierung von Dokumentmengen eignet. Die Qualität dieser Visualisierung muss mit anderen Verfahren wie etwa den Kohonen-Karten (cf. Abschnitt 4.4) verglichen werden

2.1.2.5 Weitere Modelle

Zu diesen drei wichtigsten Modellen im Text-Retrieval gibt es zahlreiche Erweiterungen, von denen hier nur einige kurz erwähnt werden sollen.

Auf dem theoretischen Fundament der mathematischen Logik baut das logische Modell der IR-Prozesse auf (cf. Fuhr 1995). Die Anfrage wird darin als Satz im Sinne der Logik betrachtet, der durch bestimmte Aussagen bewiesen wird. Im IR-Modell sind die Aussagen die Dokumente.

Das Boolesche Modell wird durch verschiedene Ansätze erweitert. Fuzzy-Retrieval-Modelle greifen auf die Fuzzy Logik (cf. Abschnitt 2.2.1) zurück und lockern den exakten Abgleich zwischen Anfrage und Dokument. Die Fuzzy Logik erlaubt Zugehörigkeitsgrade zu Mengen. Damit ist keine eindeutige Entscheidung nötig, ob ein Dokument zum Ergebnis gehört oder nicht. Vielmehr führen die Zugehörigkeitsgrade zu einem Ranking-System.

Daneben gibt es weitere Extended Boolean Modelle, die das klassische Boolesche Modell ergänzen und auch die Aufteilung des Dokumentenbestandes in relevant und nicht relevant vermeiden (cf. Womser-Hacker 1997).

Die Modelle auf Basis neuronaler Netze bilden den Schwerpunkt dieser Arbeit und werden in Kapitel 4 ausführlich besprochen.

2.1.3 Ähnlichkeitsberechnung

IR-Systeme gehören entweder zum *exact match* (Boolesches Modell) oder zum *best match* (Ranking-Verfahren) Paradigma. Beim Retrieval berechnen fast alle Modelle die Ähnlichkeit zwischen der Anfrage- und den Dokument-Repräsentationen und die ähnlichsten Dokumente bilden das Ergebnis. Das Boolesche Modell ist ein Extremfall eines Ähnlichkeitsmodells, das nur die Ähnlichkeitswerte Null und Eins zulässt, also den Bestand in relevante und nicht relevante Dokumente einteilt. Innerhalb der *best match* Verfahren haben sich in den letzten Jahren probabilistische Verfahren etabliert, die ausgefeilte statistische Methoden benutzen. Aus Benutzersicht sollte die berechnete Ähnlichkeit auf semantischem oder pragmatischem Wissen über die Texte beruhen, da Benutzer Texte suchen, die ihr Problem inhaltlich lösen. Tatsächlich suchen heutige Systeme aber nach Texten, in denen sich aus den Vorkommenshäufigkeiten der Suchwörter ein hoher Ähnlichkeitswert ergibt. Dahinter steht die Annahme aus der inhaltlichen Erschließung, an der Vorkommenshäufigkeit ließe sich die Wichtigkeit eines Wortes in einem Text ablesen (cf. Abschnitt 2.1.1).

Die Wahl einer Ähnlichkeitsfunktion aus der großen Menge von mathematischen Maßen stellt ein Problem für IR-Systeme dar. Diese Entscheidung fällt meist aus heuristischen Faktoren. Die folgende Übersicht zeigt einige der Ähnlichkeitsfunktionen, die üblicherweise im Text-Retrieval zum Einsatz kommen:

Inneres Produkt-Maß:
$$\ddot{A}_I(W_i, W_j) = \sum_{k=1}^{n} Term_{ik} \cdot Term_{jk}$$

Kosinus-Maß:
$$\ddot{A}_C(W_i, W_j) = \frac{\sum_{k=1}^{n} Term_{ik} \cdot Term_{jk}}{\sqrt{\sum_{k=1}^{n} Term_{ik}^{\,2} \sum_{k=1}^{n} Term_{jk}^{\,2}}}$$

Tanimoto-Maß: $$\ddot{A}(W_i, W_j) = \frac{\sum_{k=1}^{n} Term_{ik} \cdot Term_{jk}}{\sum_{k=1}^{n} Term_{ik}^{\,2} + \sum_{k=1}^{n} Term_{jk}^{\,2} - \sum_{k=1}^{n} Term_{ik} \cdot Term_{jk}}$$

cf. Bollmann/Konrad 1979:284

Dice-Maß: $$\ddot{A}(W1, W2) = \frac{2 \cdot \sum_{k=1}^{t} (Term_{ik} \cdot Term_{jk})}{\sum_{k=1}^{t} Term_{ik} + \sum_{k=1}^{t} Term_{jk}}$$

cf. Salton/McGill 1983

Manchmal bilden empirische Tests die Entscheidungsgrundlage für die eine oder andere Funktion. Eine eingehende Analyse der Eigenschaften von Ähnlichkeitsmaßen von Jones/Furnas 1987 zeigt, dass Formeln unterschiedliche Sensitivität für Eigenschaften der Objekte aufweisen und z.B. „within- and between-object term weight relationships“ (Jones/Furnas 1987:423) unterschiedlich stark gewichten. Demnach führen Ähnlichkeitsmaße zu unterschiedlichen Ergebnissen, je nachdem ob die Differenz der Gewichte stärker zwischen Objekten oder innerhalb der Objekte schwankt, ob die Dokument-Term-Matrix also stärkere Unterschiede in den Zeilen oder den Spalten aufweist. Die Autoren betonen, dass diese Analyse eine empirische Untersuchung nicht ersetzt. Abhängig von der Benutzersicht können verschiedene Eigenschaften eines Korpus entscheidend für eine benutzeradäquate Ähnlichkeitsbewertung sein.

Bei formal deutlich abgrenzbaren Aufgaben kommen im Information Retrieval spezielle Ähnlichkeitsfunktionen zum Einsatz. Z.B. gibt es in TREC einige Ansätze für die Behandlung kurzer Anfragen (cf. Gauch/Wang 1997 und Wilkinson et al. 1996). Außerhalb des Text-Retrieval entstehen spezielle Ähnlichkeitsfunktionen, so z.B für Image Retrieval (z.B. für Diagramme von Fabrikanlagen cf. Wakimoto et al. 1995 oder für Gesichter cf. Narasimhalu/Leong 1995) und Multimedia IR (cf. Aigrain et al. 1996, Gupta/Jain 1997).

Auch Case-Based-Reasoning (CBR) kann als ein Spezialfall von IR aufgefasst werden. CBR speichert Problembeschreibungen und –lösungen als Objekte. Aufgabe des Retrieval ist es, einen möglichst ähnlichen Fall zu einem aktuellen Problem zu finden. CBR befasst sich weiterhin mit der Anpassung der Lösung auf den neuen Fall. Die Ähnlichkeitsfunktion ist eine

der kritischen Komponenten in einem CBR-System und in diesem Rahmen wurden zahlreiche Verfahren für diesen speziellen Anwendungsfall entwickelt (für einen Überblick cf. Mántaras/Plaza 1997).

Dies zeigt, wie problematisch die Auswahl einer Ähnlichkeitsfunktion ist. Es ist offensichtlich, dass die optimale Ähnlichkeitsfunktion für Teilbestände einer Kollektion unterschiedlich sein kann. Eine optimale Ähnlichkeitsfunktion nähert die kognitive Ähnlichkeitsbeurteilung des Benutzers möglichst gut an. Dies kann eine rein formale Analyse der Dokument-Kollektion und der Anfragen kaum erreichen. Die meisten Maße behandeln alle Terme gleich, was nicht als plausibel erscheint. Es ist davon auszugehen, dass zwischen den einzelnen Termen komplexe Zusammenhänge und Abhängigkeiten bestehen, die sich in der menschlichen Relevanzbewertung widerspiegeln. Grimmer/Mucha 1998 erklären, dass eine Gewichtung von Merkmalen oder Objekten bei Ähnlichkeitsfunktionen im Bereich Data Mining denkbar ist. Im Information Retrieval ist dies zwar nicht üblich, kann jedoch gerade bei heterogenen Objekten oder Objekten mit heterogenen Repräsentationen von Vorteil sein.

Als hypothetisches Beispiel soll die Erkennung von handgeschriebenen Postleitzahlen und Ortsnamen auf Briefen betrachtet werden. Im Kontext eines Information Retrieval Systems stellt ein Brief die Anfrage dar und die gespeicherten Namen und Postleitzahlen sind die Dokumente. Jedes Element bedingt eine spezifische Ähnlichkeitsfunktion. Die Ortsnamen lassen mit Verfahren für Wörter bzw. Strings behandeln, während die Zahlen eine Mischung aus spezielle String-Verfahren und Fakten-Retrieval erfordern. Das Gesamtergebnis sollte die Ähnlichkeiten aus den beiden verschiedenen Repräsentationen des Objekts mit unterschiedlichen Gewichten kombinieren, da die Zuverlässigkeit der Teilergebnisse unterschiedlich ausfällt.

Ein Blick auf die psychologische Forschung zum Thema Ähnlichkeit wirft ein weiteres gravierendes Problem auf. Tversky 1977 zeigt, dass die formalen Eigenschaften Transitivität und Symmetrie für menschliche Ähnlichkeitsurteile nicht zutreffen müssen. Zwar werden in den letzten Jahren mathematische Maße diskutiert, die ohne Transitivität auskommen, aber fast alle Maße sind symmetrisch. Tversky 1977 zeigt, dass bei kognitiven Urteilen das Phänomen des Prototypen eine Rolle spielt. So wurde in seinen Experimenten die Ähnlichkeit zwischen *Rußland* und *Kuba* höher eingeschätzt als zwischen *Kuba* und *Rußland*. Die menschliche Ähnlichkeitsbewertung muss also nicht immer symmetrisch sein. Die von Tversky 1977 vorgeschlagene Familie von Ähnlichkeitsfunktionen löst das Problem für den Anwendungsfall Information Retrieval nicht, da darin nur binäre Eigenschaften zugelassen sind. Wünschenswert wäre ein Modell, das die Ähnlichkeitsbeziehungen in einer Do-

mäne nur aufgrund menschlicher Urteile lernt und das auf keinerlei formalen Voraussetzungen beruht.

Die Berechnung der Ähnlichkeit ist aufwendig, da sie für jedes Dokument durchgeführt wird. Bentz et al. 1989 und Hagström 1996 stellen eine Methode vor, die nur auf Teile einer Objekt-Repräsentation zugreift, um effizient Ähnlichkeiten zu berechnen (cf. Abschnitt 4.2.1).

2.1.4 Evaluierung

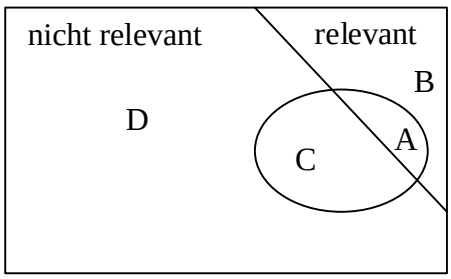
Das Ziel von Information Retrieval Systemen ist die Zufriedenheit der Benutzer. Ein Benutzer wiederum ist zufrieden, wenn die nachgewiesenen Dokumente helfen, sein Problem zu lösen.

Die Auswirkungen verschiedener Modelle, ihrer Parameter und ihrer Kombination auf das Ergebnis lassen sich a priori nicht vorhersagen. Deshalb nimmt die Evaluierung einen zentralen Platz in der IR-Forschung ein. Die folgenden Abschnitte stellen einige Standard-Verfahren sowie die große Evaluierungsstudie TREC (Text Retrieval Conference) vor.

2.1.4.1 Qualitätsmaße

Grundlage der Evaluierung von Information Retrieval Ergebnissen ist die Relevanz. Die Relevanz ist die Einschätzung eines Dokuments durch einen Benutzer und damit ein dynamisches und subjektives Phänomen. Ihre objektive Messung ist problematisch. Empirische Untersuchungen nutzen meist die Relevanz-Entscheidungen von Experten als Maßstab. Mizzaro 1997 referiert die Forschung zur Relevanz.

Von der Relevanz leiten sich eigentlichen Messgrößen ab. In der Regel sind die von einem System gefundenen Dokumente keineswegs alle relevant und die Ergebnismenge enthält auch keineswegs alle potenziell relevanten Dokumente des Korpus. Das erste Phänomen wird durch die Größe Recall, das zweite durch die Precision gemessen. Abbildung 2-8 stellt die beiden Maße vor.



	Im Ergebnis	Nicht im Ergebnis
Relevant	A	B
Nicht relevant	C	D

Die Gesamtmenge besteht aus relevanten und nicht relevanten Dokumenten, getrennt durch die schräge Linie. Das System präsentiert einen Ausschnitt aus der Gesamtmenge als Ergebnis (gekennzeichnet durch das Oval).

$$\mathrm{Re}\,call = \frac{\mathrm{A}}{\mathrm{A}+\mathrm{B}}$$

$$\mathrm{Pr}\,ecision = \frac{\mathrm{A}}{A+C}$$

Abbildung 2-8: Recall und Precision

Am Recall ist problematisch, dass der Nenner nur schwer bestimmbar ist. Er beinhaltet die Anzahl relevanter Dokumenten in der gesamten Kollektion, die nur bei kleinen Kollektionen von Juroren bestimmt werden kann. Bei größeren Datenmengen wird diese Zahl meist geschätzt. Vergleichende Tests benutzen häufig die Pooling-Methode, bei der die relevanten Ergebnis-Dokumente mehrerer Systeme gleich der Anzahl der relevanten Dokumente in der Kollektion gesetzt werden (cf. Harman 1995).

Recall und Precision lassen sich nicht gleichzeitig optimieren. Der Recall kann einfach durch eine Vergrößerung der Ergebnismenge gesteigert werden, dabei sinkt aber die Precision. Recall und Precision eignen sich zunächst für Information Retrieval Modelle, die eine Aufteilung des Dokumentenbestandes in relevant und nicht relevant vornehmen, wie das Boolesche Modell. Für Ranking-Systeme wird in der Regel ein Recall-Precision-Graph erstellt. Da der Benutzer typischerweise die Ergebnismenge von oben bis zu einem bestimmten Punkt abarbeitet, vergrößert sich schrittweise die Ergebnismenge. Für jedes Dokument mehr lassen sich ein neuer Recall und eine neue Precision berechnen, die in ein Koordinatensystem eingezeichnet werden. Um Systeme anhand einer Zahl vergleichbar zu machen, wird meist die Precision auf neun Recall-Niveaus von 0,1 bis 0,9 gemittelt (*frozen rank method*). Auch neuerere Arbeiten zur Evaluierung der Ergebnisse von Internet-Suchmaschinen nutzen Recall und Precision (cf. Hawking et al. 1999).

Gerade bei Ranking-Verfahren kann der Benutzer beliebig viele Dokumente aus der Ergebnisliste betrachten. Je mehr Dokumente er evaluiert, desto mehr relevante Dokumente befinden sich darunter. Würde ein Benutzer alle Doku-

mente der Kollektion betrachten, dann fände er alle relevanten Dokumente. Der Recall erreicht den optimalen Wert von Eins. Die Precision wäre in diesem Extremfall aber sehr schlecht, da der Benutzer auch alle nicht relevanten Dokumente sehen würde. Bei Ranking-Verfahren hängt der Recall und die Precision also vom Verhalten des Benutzers ab.

Neben diesen einfachen Maßzahlen gibt es auch komplexere Größen wie das E-Maß von van Rijsbergen, das Recall und Precision gewichtet kombiniert:

$$e = 1 - \frac{(b^2 + 1) \cdot precision \cdot recall}{b^2 \cdot precision + recall}$$

Womser-Hacker 1989:49

Neuere Ansätze beziehen weitere Faktoren mit ein, so etwa die Erfolgsfaktorenanalyse aus dem Informationsmanagement (cf. Bayraktar/Womser-Hacker 1998). Dabei fließen bei der Auswahl eines geeigneten Information Retrieval Systems auch Faktoren wie Benutzerfreundlichkeit, Service des Herstellers oder Funktionsumfang mit ein.

2.1.4.2 Text Retrieval Conference (TREC)

Da Forscher für ihre IR-Tests verschiedenste Textkollektionen benutzten, waren die Ergebnisse bisher oft nicht vergleichbar. Die TREC (Text REtrieval Conference) Initiative hat diese Situation verbessert. Die TREC-Konferenzen bieten eine einheitliche Testumgebung und sind als gemeinsme Plattform zum Leistungsvergleich organisiert. Die Beiträge werden jährlich publiziert (cf. Harman 1993/94/95/96, Voorhees/Harman 1997/98/99), einen Überblick gibt Womser-Hacker 1997. Die Initiative findet großen Anklang; so beteiligen sich an TREC 6 bereits 58 Retrieval-Systeme (cf. Voorhees/Harman 1998).

TREC bietet den Teilnehmern eine große Dokument-Kollektion, Anfragen und übernimmt für eingereichte Ergebnisse die Auswertung. Als Trainingsdaten stehen die Kollektionen der Vorjahre zur Verfügung, die intellektuelle Relevanzurteile enthalten. Die Ergebnisse der Systeme werden untereinander verglichen. Als Datengrundlage bietet TREC hauptsächlich Zeitungs- und Nachrichtentexte. Viele Experimente außerhalb von TREC haben jedoch gezeigt, dass ein Verfahren bei anderen Daten zu anderen Ergebnisse führen kann. Bereits aus diesem Grund erlaubt auch TREC keine endgültige Entscheidung über das *beste* Retrievalsystem.

Der wichtigste Bestandteil der TREC-Experimente ist das Ad-hoc Retrieval, bei dem die Standard-Situation im Information Retrieval der Ausgangspunkt

ist. Ein Benutzer stellt eine Anfrage, die Ergebnisse aus einer großen Menge von Text-Dokumenten liefert. Daneben gibt es Routing-Aufgaben, die einem automatisierten Filter entsprechen. Feststehende Routing-Aufgaben treffen dabei auf ein Strom von Dokumenten, aus dem die relevanten gefiltert werden. Die Veranstalter erstellen die Anfragen (im TREC-Jargon Topics) in drei Detaillierungsebenen. Neben Überschrift und Kurzbeschreibung in einem Satz gibt es eine sog. Langbeschreibung. Die Teilnehmer entscheiden sich für eine Fassung und arbeiten damit. Getrennte Bewertungen erfolgen für Retrieval-Systeme, die mit intellektuell aus den Topics erstellten Anfragen arbeiten.

Um die Anzahl der relevanten Dokumente für die Bestimmung des Recall abzuschätzen wird die Pooling-Methode angewandt, bei der Juroren alle Ergebnis-Dokumente der verschiedenen Systeme bewerten, aber nicht die gesamte Kollektion durchsuchen.

Neben dem Standard-Retrieval-Experimenten und den Routing-Experimenten haben sich weitere Teile, sogenannte Tracks etabliert, darunter ein Cross-Language-Track, ein Interactive Track, bei dem Relevanz-Feedback wichtig ist und ein Track mit sehr großer Datenmenge. In Abschnitt 4.8 werden die TREC-Ergebnisse der bis dahin vorgestellten Retrieval Systeme auf der Basis neuronaler Netze diskutiert.

Der Schwerpunkt bei TREC liegt auf einer reinen Systemsichtweise, die viele pragmatische Faktoren der Suche ausblendet. Gerade die Ergebnisse von TREC unterstreichen aber nochmals die Wichtigkeit dieser Faktoren. So hat TREC gezeigt, dass sich die Qualität der besten IR-Systeme in dieser systemlastigen Evaluierung kaum unterscheidet, dass allerdings die Ergebnisse unterschiedlich sind. Jedes Verfahren bringt also andere relevante Dokumente, jedes bringt aber etwa gleich viele. Große Hoffnungen setzen die Forscher daher in Fusionsverfahren, die Ergebnisse mehrerer Verfahren so kombinieren, dass die relevanten Dokumente der einzelnen Verfahren im Gesamtergebnis einen hohen Rang erhalten (cf. z.B. Voorhees et al. 1995, Lee 1995, Bartell et al. 1994).

2.1.5 Beispiel für ein Text-Retrieval-System

Derzeit sehr populäre Beispiele für Text-Retrieval-Systeme sind die Internet-Suchmaschinen. Sie suchen nach HTML-Seiten im Internet, die teilweise auch multimediale Elemente enthalten. Nach wie vor überwiegt im Internet jedoch Text und die multimedialen Elemente werden weitgehend wie Text behandelt, so dass die Einordnung als Text-Retrieval-Systeme gerechtfertigt ist. Einen Überblick über Information Retrieval im Internet bietet Bekavac 1999.

Als Beispiel wurde das System von Northern Light[1] gewählt, die zur Zeit zu den größten Suchmaschinen gehört. Nach eigenen Angaben hat Northern Light am 11.4.2000 ca. 229 Millionen Seiten erfasst[2]. Nach einer Schätzung von Notess 2000 umfasst die Kollektion von Northern Light tatsächlich zwischen 204 und 237 Millionen Seiten indexiert.

Die Grundfunktionalität aller Internet-Suchmaschinen ist weitgehend identisch. Sie indexieren im Internet publizierte Seiten. Hilfsprogramme (Crawler, Spider) suchen ausgehend von bekannten Seiten nach weiteren Dokumenten, indem sie die Hypertext-Verbindungen verfolgen. Die Suchmaschine indexiert die Seiten und speichert die Repräsentation und die Adresse der Seite.

Für den Benutzer ähneln sich die Internet-Suchmaschinen in den Hauptfunktionen stark. Sie bieten eine Eingabezeile für die Suchbegriffe (cf. Abbildung 2-9). Dabei muss keine Syntax beachtet werden, es stehen aber Operatoren zur Verfügung, die eine Feldsuche (z.B. nur im Titel) oder andere Funktionen (z.B. Suche nur im Bereich eines Web-Servers) ermöglichen. Die Suchverfahren sind durchweg Ranking-Verfahren, jedoch ist in den meisten Internet-Suchmaschinen alternativ die Formulierung einer Booleschen Anfrage möglich. Bei Northern Light kann die Boolesche Eingabe direkt in der Eingabezeile erfolgen, während andere Suchmachinen eine zusätzliche Seite für Boolesche Anfragen vorhalten. Die genauen Suchverfahren und Algorithmen veröffentlichen die Betreiber in der Regel nicht. Northern Light gibt in der Hilfe zur Suche[3] lediglich sehr allgemeine Hinweise darauf, wie das System die Relevanz der Dokumente ermittelt. Faktoren für die Gewichtung sind demnach die Vorkommenshäufigkeit, das Vorkommen im Titel und die Poularität einer Internet-Seite, die anhand der darauf verweisenden Verbindungen errmittelt wird. Die Rolle der Frequenz im Information Retrieval ist bekannt (cf. Abschnitt 2.1.1) und auch die höhere Gewichtung von Termen, die im Titel vorkommen, ist üblich. Die Popularität von Dokumenten ist ein für das Internet spezifischer Einflussfaktor.

Die Startseite von Northern Light bietet neben der Eingabezeile für die Such-Terme noch eine Auswahlmöglichkeit für die Quelle der Dokumente an. Weiterhin kann ein Benutzer mehrere spezielle Sucharten anwählen wie etwa eine Suche nach Nachrichten-Meldungen und eine Suche nach Dokumenten für Investoren. Die Rolle der Werbung, über die sich die meisten Suchmaschinen finanzieren, ist bei Northern Light etwas zurückgenommen und

---

[1] http://www.northernlight.com/

[2] http://www.northernlight.com/docs/intelligent_stats.html

[3] http://www.northernlight.com/docs/gen_help_faq.html#q14 und
http://www.northernlight.com/docs/gen_help_faq_webmaster.html#rank

nimmt wenig Platz ein. Mehr Platz belegen zusätzliche Mehrwertdienste wie aktuelle Schlagzeilen und Börsennachrichten. Solche Dienste sind auch bei anderen Anbietern üblich. Damit wollen die Suchmaschinen sich als Eingangsseite zum Internet (Portal) für möglichst viele Benutzer etablieren.

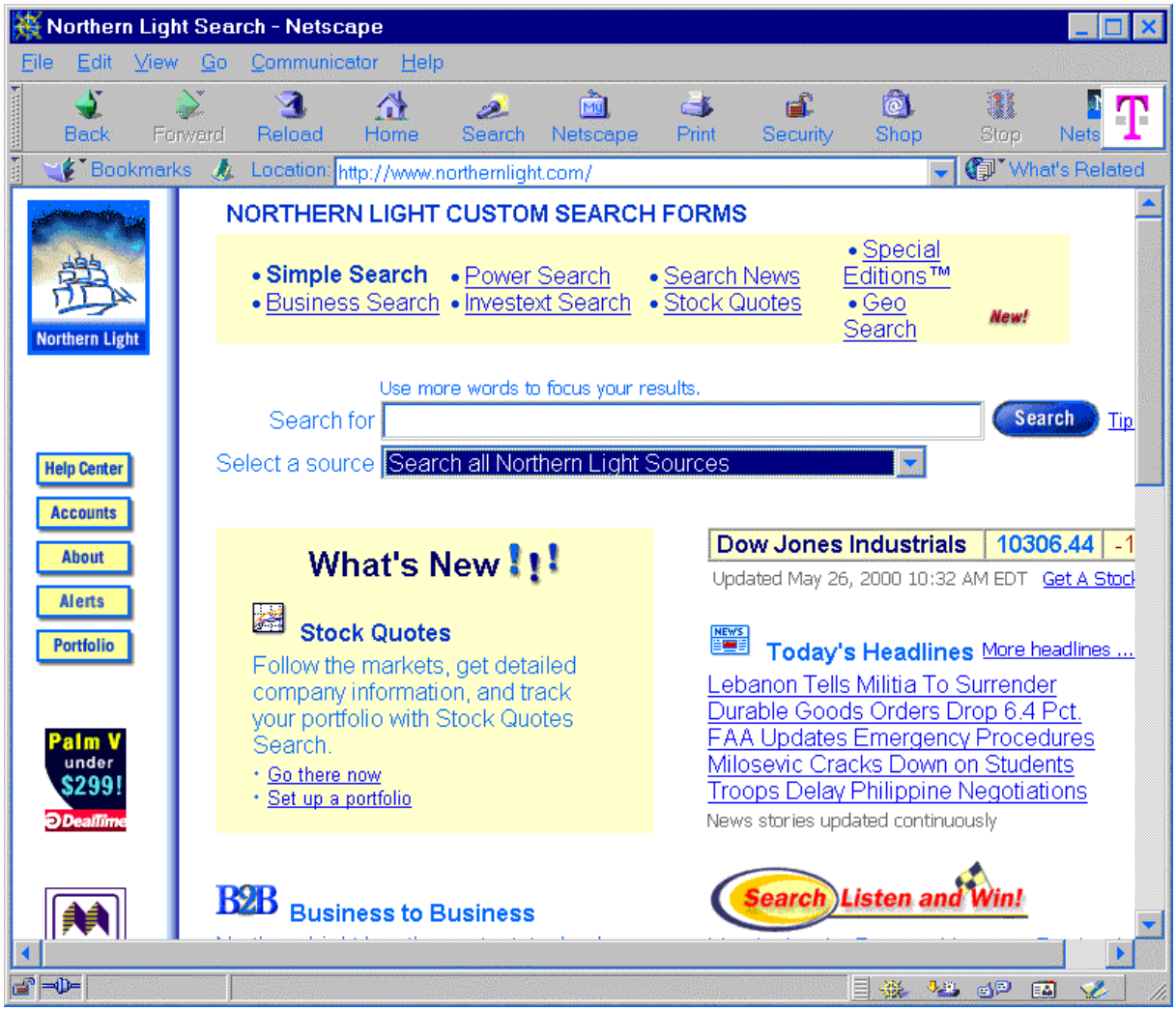


Abbildung 2-9: Startseite von Northern Light

Die nach System-Relevanz geordnete Ergebnisanzeige von Northern Light in Abbildung 2-10 verdeutlicht einen der größten Vorteile der Internet-Suchmaschinen, den direkten Zugriff auf das gefundene Dokument. Zwar enthält die Schmaschine nur die Repräsentation des Dokuments, sie bietet aber einen Link auf das Ergebnis-Dokument, dem der Benutzer folgen kann. Allerdings sind viele Dokumente aufgrund der Dynamik des Internets nicht mehr an der ursprünglichen Adresse verfügbar. Nach Notess 2000a führen 5,7% der Verbindungen von Northern Light ins Leere.

Als Entscheidungshilfe für den Benutzer zeigt die Suchmaschine den Beginn des Textes der Seite an. Ein weiterer von Northern Light angebotener Mehr-

wert sind die in Abbildung 2-10 auf der linken Seiten sichtbaren Cluster von Dokumenten, in denen gefundenen Dokumente eingeordnet wurden.

Abbildung 2-10: Ergebnisanzeige in Northern Light

Eine Besonderheit von Northern Light sind die Special Collections, zu denen das vierte Ergebnisdokument in Abbildung 2-10 gehört. Diese Dokumente sind nicht frei im Internet verfügbar, sondern es handelt sich um kostenpflichtige Artikel aus Zeitschriften und anderen potentiell hochwertigen Quellen. Damit bietet Northern Light mit einer Anfrage den Zugriff auf das Internet und kommerzielle Datenbanken. Während andere Suchmaschinen

ausschließlich kostenfreie Suchen im Internet ermöglichen, erhöht Northern Light die Zahl der durchsuchten Dokumente und schafft möglicherweise einen neuen Informationsmarkt für die kostenpflichtigen Zeitschriften-Texte. Es ist natürlich durchaus denkbar, dass Northern Light kostenpflichtige Dokumente stets höher gewichtet und sie so häufiger den Benutzern anbietet.

Notess 2000b berichtet, dass die wichtigsten Suchmaschinen bei mehreren Tests nur wenig Überschneidung in ihren Ergebnissen aufwiesen. Allerdings benutzte Notess 2000b mit fünf Anfragen nur relativ wenig Daten, um die Überschneidung zwischen den Ergebnissen zu mesen. Dies deckt sich mit den Ergebnissen aus TREC, bei dem die verschiedenen Systeme relativ wenig Überschneidung bei den Treffern aufwiesen (cf. Abschnitt 2.1.4.2). Während bei TREC aber zumindest die Kollektion identisch ist, kommt bei den Suchmaschinen ein weiterer Faktor hinzu, der zu einer geringen Überlappung der Ergebnisse führt. Die Grundmenge an Dokumenten ist unterschiedlich und hängt von den jeweiligen Crawlern ab.

Andere Beispiele für Text-Retrieval-Systeme, die für lokale Dokument-Kollektionen eingesetzt werden, sind z.B. der FULCRUM SearchServer[1], Knowledge Retrieval von der Firma Verity[2] und der Intelligent Miner for Text von IBM[3].

2.2 Fakten-Retrieval

Unter Fakten werden in diesem Kontext stark strukturierte Daten verstanden, wie sie typischerweise in Datenbanken vorliegen. Die prototypischen Vertreter bilden numerische Fakten. Der Übergang zu textuellen Daten ist fließend, da auch numerische Fakten in natürlicher Sprache formuliert werden können. Betrachtet man Fakteninformationssysteme im realen Einsatz, so spielt Vagheit bei ihnen ebenso eine große Rolle wie bei unstrukturierten Texten.

Am Beispiel eines Informationsprozesses zur Buchung einer Reise lässt sich dies schnell zeigen. Der Preis, den ein Benutzer als Bedingung setzt, ist selten als exakte Grenze zu betrachten. Vielmehr sind viele Reisende bereit, bei einem passendem Angebot auch einige Prozent mehr zu bezahlen. Auch der Zielort ist häufig vage. Für viele Urlaubssuchende aus dem Norden und der Mitte Europas z.B. ist der Mittelmeerraum oder ein Land darin das vage Ziel. Das konkrete Reiseziel kristallisiert sich häufig bei der Informationssuche im

[1] http://www.hummingbird.com/products/dkm/km/searchserver

[2] http://www.verity.com/products/enterprise

[3] http://www-4.ibm.com/software/data/iminer/fortext/ibm_tse.html

Bekanntenkreis oder Reisebüro heraus. Viele Buchungssysteme ignorieren jedoch den vagen Charakter dieser Bedingungen und erlauben nur exakte Suchparameter wie etwa einen konkreten Ort.

Allgemein betrachtet entsteht auch in Fakteninformationssystemen die Anfrage im Laufe des Interaktionsprozesses und steht nicht von vornherein fest. Die Informationsbedürfnisse der Benutzer ergeben sich in einer Problemsituation und lassen sich daher nicht exakt in eine Anfrage überführen. Erst Zwischenergebnisse führen zu einer Konkretisierung der Anfrage und so zum endgültigen Ergebnis.

Für Fakteninformationssysteme sind also auch vage Verfahren nötig, die hinausgehen über die Boolesche Logik, wie sie die meisten Datenbank-Managementsystemen (DBMS) anbieten. Solche Systeme, die die Perspektive der Benutzer in den Vordergrund stellen, können auch dem Information Retrieval zugerechnet werden.

In solchen Fakteninformationssystemen bewährt sich oft auch das Ranking-Paradigma. Da diese Anwendungen in der Regel nicht im Kontext der IR-Forschung genannt werden, werden sie hier ausführlicher besprochen. Zunächst wird die Fuzzy Logik als Extension der klassischen zweiwertigen Logik eingeführt, da sie eine Grundlage vieler derartiger Systeme bildet. Dazu gehören Erweiterungen der klassischen Datenbankmodelle und ihrer Abfragesprachen, die der folgende Abschnitt vorstellt. Die Fuzzy Logik wird gemeinsam mit neuronalen Netzen und genetischen Algorithmen (cf. Abschnitt 4.7.1) zum Paradigma des Soft-Computing gerechnet (cf. z.B. Zadeh 1994). Da Fakten-Retrieval häufig auf Fuzzy Logik zurückgreift, wird es hier exemplarisch als Möglichkeit der vagen Modellierung vorgestellt. Neuronale Netze, die in dieser Arbeit als vage Modellierungsmethode im Text-Retrieval und für Heterogenitätsbehandlung diskutiert werden, stellt Kapitel 3 vor.

Abschließend stehen zwei Beispiele für Fakteninformationssysteme im Mittelpunkt, welche die Berücksichtigung vager Suchprozesse illustrieren.

2.2.1 Repräsentationen mit Fuzzy Logik

Die Grundidee der Fuzzy Logik besteht darin, im Gegensatz zur zweiwertigen aristotelischen Logik beliebige Wahrheitswerte zwischen Null und Eins zuzulassen. Eine Aussage wie „Adenauer war populär“ kann z.B. einen Wahrheitswert von 0,8 erhalten. Durch Übertragung dieser Grundidee auf die Mengenlehre entstehen Fuzzy Mengen (*Fuzzy Sets*), zu denen Elemente mit einem bestimmten Grad (in der Regel μ) gehören. Das Prinzip der Fuzzy Logik wird dabei auf Aussagen zu Mengen angewandt. Der Satz „Adenauer gehört zur Menge der populären Menschen“ ist demnach nicht wahr oder

falsch, sondern besitzt einen bestimmten Wahrheitswert. Fuzzy Logik und Fuzzy Mengen beruhren somit auf dem gleichen Fundament und werden in dieser Arbeit synonym gebraucht. Fuzzy Mengen kennzeichnet eine Tilde (z.B. Ã).

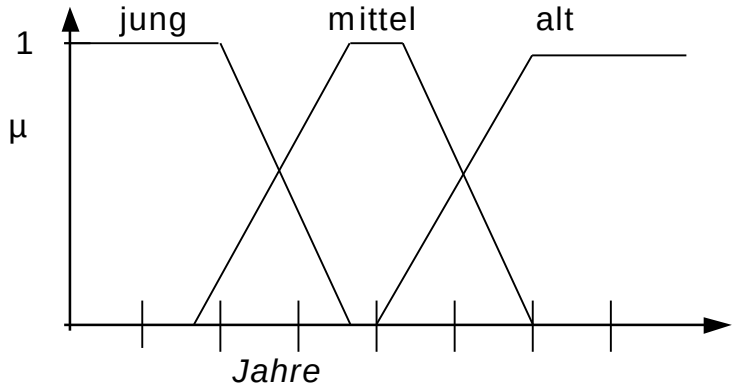


Abbildung 2-11: Zugehörigkeitsfunktionen von *jung*, *mittel* und *alt*: $\mu_{\tilde{A}}$ (Jahre)

Natürlichsprachliche Konzepte wie etwa *hohes Alter* können mit Fuzzy Mengen besser modelliert werden als mit der klassischen Mengenlehre. Die klassische Mengenlehre gibt eine exakte Grenze an, ab der z.B. jemand *alt* ist, was immer unbefriedigend ist. In der Fuzzy Logik lässt sich dieser Übergang fließend modellieren, was eher dem menschlichen Denken entspricht. Diese Fuzzy Definition bildet die Zugehörigkeitsfunktion μ. Einführungen in Fuzzy Logik bieten Mayer et al. 1993 und Nauck et al. 1994. Eine umfassende Darstellung der mathematischen Grundlagen findet sich in Zimmermann 1995.

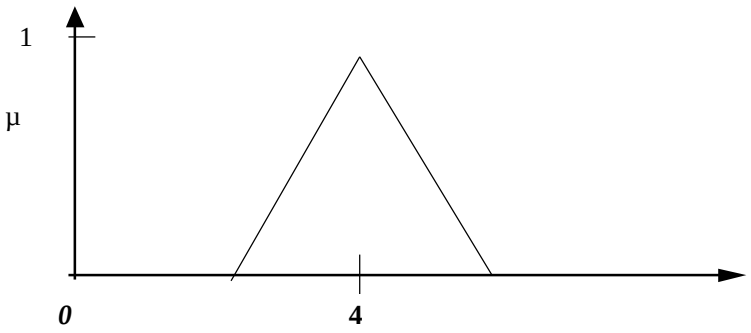


Abbildung 2-12: Zugehörigkeitsfunktion von „ungefähr 4“

Im Rahmen der Fuzzy Logik sind Begriffe wie *alt* oder *jung* linguistische Variablen einer Basisvariable (hier: Alter). Weitere Modifizierer wie *sehr*, *mehr*

oder weniger oder *nicht* können hinzu treten. Eine ähnliche Art der Modellierung bietet sich für Zahlen an. Z.B. kann die unscharfe Zahl *ungefähr vier* durch folgende Zugehörigkeitsfunktion definiert werden.

Um logische Aussagen zu verbinden und Schlussfolgerungen zu ziehen, sind Operatoren nötig, die atomare Aussagen verknüpfen. In der klassischen Logik und der Booleschen Algebra sind dies besonders die Operatoren UND und ODER. Sie entsprechen nicht ihren natürlichsprachlichen Pendants, sondern sind formal definiert. Dies führt v.a. bei Anfängern zu Schwierigkeiten bei der Benutzung, da die natürlichsprachliche Semantik die formal definierte Semantik überlagert (cf. Cooper 1988). In der Fuzzy Logik existieren eine Vielzahl von Operatoren, die an die Stelle von UND und ODER treten.

Die einfachsten Operatoren sind der Minimum- und der Maximum-Operator, wobei der Minimum-Operator dem AND-Operator und der Bildung der Schnittmenge entspricht. Die Zugehörigkeitsfunktion des Minimum-Operators lautet folgendermaßen:

$$\mu\ (Min\ (\tilde{A}, \tilde{N}))\ (x) = Minimum\ (\mu_{\tilde{A}}\ (x), \mu_{\tilde{N}}\ (x)\ )$$

Zimmermann 1995:28

Bsp.: Ein Land X_1 gehört mit 0,3 zur Menge der politisch stabilen Länder Ã und mit 0,8 zur Menge der attraktiven Märkte Ñ. Der Minimum-Operator soll durch Verknüpfung dieser beiden Einflussgrößen ein Maß für Investitionsentscheidungen liefern.

$\mu\ (Min\ (\tilde{A}, \tilde{N}))\ (X_1) = Minimum\ (0{,}3\ ;\ 0{,}8\ ) = 0{,}3$

Der Minimum-Operator liefert den kleineren der zwei Ausgangswerte, während der größere Wert keinen Einfluss hat. Dies führt zu Problemen, wie ein zweites Beispiel verdeutlicht:

Bsp.: Land X_2 gehört mit 0,4 zur Menge der politisch stabilen Länder Ã und mit 0,4 zur Menge der attraktiven Märkte Ñ.

$\mu\ (Min\ (\tilde{A}, \tilde{N}))\ (X_2) = Minimum\ (0{,}4\ ;\ 0{,}4\ ) = 0{,}4$

Der Minimum-Operator bewertet Land X_2 höher als Land X_1, obwohl X_1 einen wesentlich attraktiveren Markt verspricht. Da die politische Stabilität sich nur wenig unterscheidet, würde Land X_2 von einem Menschen wohl höher bewertet. Der Minimum-Operator ist häufig unangemessen. Das gleiche gilt für den Maximum-Operator, der analog definiert ist.

Wie das Beispiel zeigt, geht bei Verwendung des Minimum-Operators nur einer der beiden verknüpften Werte in das Ergebnis ein, während der Mensch beide Werte berücksichtigt und eine Kompensation zwischen ihnen zulässt. Ist eine Bedingung sehr gut erfüllt, so akzeptiert man bei einer anderen Bedingung eine größere Abweichung. Um dies in der Fuzzy Logik abzubilden, wurden zahlreiche sogenannte kompensatorische Operatoren definiert. Das algebraische Produkt z.B. berücksichtigt beide Werte:

$$\mu\ (Alg.\ Prod.\ (\tilde{A}, \tilde{N}))\ (x) = \mu\tilde{A}\ (x) \quad \mu\tilde{N}\ (x)$$

Zimmermann 1995:28

Die Bewertung mit dem Algebraischen Produkt ergibt im obigen Beispiel eine Reihenfolge, die eher dem menschlichen Urteil entspricht:

Bsp.: μ (Alg. Prod. ($\tilde{A}$, $\tilde{N}$)) (X_1) = $\mu_{\tilde{A}}$ (X_1) $\mu_{\tilde{N}}$ (X_1) = 0,8 · 0,3 = 0,24
μ (Alg. Prod. ($\tilde{A}$, $\tilde{N}$)) (X_2) = $\mu_{\tilde{A}}$ (X_2) $\mu_{\tilde{N}}$ (X_2) = 0,4 · 0,4 = 0,16

Das Algebraische Produkt führt also zu einer höheren Bewertung von Land X_2, die angemessener erscheint. Neben dem Algebraischen Produkt existieren die Gamma-Operatoren, bei denen sich der Grad der Kompensation parametrisieren lässt (cf. Zimmermann 1995). Wie stark die Kompensation sein soll, hängt stark von der Anwendung und der Semantik der entsprechenden Operation ab (cf. Womser-Hacker 1997a).

Eine wichtige Rolle im Entwurf eines Fuzzy-Systems spielt die Erarbeitung des Wissens aus den vorhandenen Quellen. Dazu gehören die Festlegung der linguistischen Variablen, die Zahl der zugelassenen Terme und Modifizierer, der Zugehörigkeitsfunktionen und die Auswahl oder Parametrisierung der Operatoren. Dieses Knowledge Engineering muss in Zusammenarbeit mit den Benutzern erfolgen. Verschiedene Konzepte hierfür diskutiert Womser-Hacker 1997a.

2.2.2 Vages Fakten-Retrieval als Erweiterung von Datenbanksystemen

Fuzzy Erweiterungen für Datenbankabfragesprachen wie SQL schlagen bereits zahlreiche experimentelle Systeme vor. Einen ausführlichen Überblick über die Verbindung von Fuzzy Logik und Datenbank-Managementsystemen (DBMS) bieten Petry 1996 und Medina/Cubero et al. 1994. Eine Darstellung

von Fuzzy-Systemen im Rahmen der IR-Forschung bietet Womser-Hacker 1997.

Bereits kurz nach der Formulierung der Fuzzy Set Theory durch Lofti Zadeh (Zadeh 1965) entstanden erste Überlegungen zu ihrem Einsatz im Information Retrieval. Tahani 1977 entwirft ein allgemeines Modell und befasst sich mit Implementierungsfragen. Sommer 1978 beschreibt den Entwurf eines Immobilien-Systems, das einen vagen Abgleich zwischen dem gewünschten und den gespeicherten Objekten zulässt.

Ein Ansatz ohne Fuzzy Logik beschreiben Rabitti/Savino 1990. Ihr System MULTOS ermöglicht das Retrieval von Geschäftsdokumenten, wobei sich die Bedingungen auf einzelne Teile eines typischen Geschäftsbriefs beziehen können wie Datum, Logo oder Produktbeschreibung. Eine Erweiterung um vage Bedingungen ermöglicht u.a. die tolerante Suche nach Zahlen („Der Preis ist ungefähr DM 5000.-“) und erlaubt die Gewichtung von Bedingungen.

Yager/Larsen 1993 stellen ein System vor, das eine Anfrage mit vagen Kriterien in eine exakte SQL-Anfrage umsetzt. Diese exakte Anfrage fasst die Bedingungen sehr weit, um auch die Elemente mit den niedrigsten Zugehörigkeitswerten zu erhalten. Die Anfrage wird an eine relationale Datenbank geschickt und die Ergebnisse werden anhand der ursprünglichen Anfrage sortiert.

Bosc/Pivert 1991 untersuchen die von ihnen entwickelte Abfragesprache SQLf, die auf der Basis von SQL Fuzzy Retrieval zulässt. Sie stellen fest, dass Äquivalenzen, wie sie in SQL zwischen syntaktisch unterschiedlichen Formulierungen vorkommen, in SQLf weitgehend weiter bestehen.

Neben vagen Anfragen wird auch auf der Datenseite Fuzzy Logik zur Modellierung unsicheren Wissens eingesetzt. Medina/Pons et al. 1994 stellen mit GEFRED ein generelles Modell für ein Fuzzy-Relationales Datenbanksystem vor. Es erlaubt die Repräsentation verschiedener Arten von Unsicherheit wie Lücken und Fuzzy Mengen. Über das gesamte Modell kann mit vagen Attributen abgefragt werden.

Allein die Erweiterungen von DBMS um Fuzzy-Komponenten und vager Attribute reicht noch nicht aus, um ein System benutzerfreundlicher zu gestalten. Die erhöhte Komplexität einer Sprache mit vagen Operatoren führt eher zum Gegenteil. Wichtig ist die detaillierte Analyse von Benutzeranforderungen und ihre Abbildung auf die formalen Konzepte.

2.2.3 Beispiele für Fakten-Retrieval-Systeme

Während bei Retrieval Systemen für Texte und für andere Objekte wie etwa Multimedia-Dokumente, inzwischen weitgehend Konsens besteht, dass vage Modellierungen erforderlich sind, ist dies für Fakteninformationssystemen noch nicht immer der Fall. Deshalb werden in den folgenden Abschnitten zwei entsprechende Systeme diskutiert. Da die Vagheitsmodellierung sich stark an den Bedürfnissen der Benutzer orientiert, um so menschengerechte und aufgabenangemessene Systeme zu schaffen, wird der Anwendungskontext ausführlich vorgestellt.

2.2.3.1 Werkstoffinformationsystem WING

Womser-Hacker 1997a beschreibt das System FUZZY-WING, das im Rahmen des Projekts WING-IIR (Werkstoffinformationssystem mit natürlichsprachlicher/graphischer Benutzungsoberfläche - Intelligentes Information Retrieval, cf. Krause/Womser-Hacker 1997) an der Universität Regensburg entwickelt wurde und das tolerante Suchen und Retrieval mit vagen Bedingungen in einem Werkstoffinformationssystem zulässt.

Im Projekt WING arbeitete eine Forschergruppe von 1989 bis 1996 in Zusammenarbeit mit mehreren Industriepartnern an einem benutzerfreundlichen Zugang zu Werkstoffdaten (cf. Krause/Womser-Hacker 1997). Ausgangspunkt war die Frage, welche Form der Mensch-Maschine-Interaktion in dieser Domäne optimal sei. Getestet wurden hierarchische Suchbäume, ein formalsprachlicher Zugang mit und ohne grafische Unterstützung, ein natürlichsprachlicher Zugang, Query-by-example, Hypertext-Verknüpfungen und ein grafischer, am Gegenstandsbereich orientierter Zugang. Der grafische Zugang führte insgesamt zu den besten Ergebnissen, jedoch waren in vielen spezifischen Situationen andere Formen der Interaktion überlegen. So eignet sich z.B. der natürlichsprachliche Zugang besonders als Einstieg für ungeübte Benutzer und Query-by-example zeigte sich bei Abfragen über nur eine Tabelle als sehr effektiv. Dies führte zur Realisierung des Prototypen WING-M1, der für die verschiedenen Aufgaben die jeweils besten Elemente einsetzte und in Benutzertests empirisch überprüft wurde. Dabei zeigten sich zahlreiche Probleme bei der Benutzung des Systems, die v.a. in folgende Kategorien fielen (cf. Marx/Schudnagis 1997:55f.):

- Flexibilität bei der Navigation: Wechsel zwischen einzelnen Suchtypen, die in eigenen Fenstern realisiert sind erwies sich als schwierig.
- Dialogsteuerung: Die Suchtypen bestanden aus sequentiell abzuarbeitenden Teilschritten, die durch das Drücken von Buttons beendet werden mussten. Die Beschriftung der Buttons mit inhaltlichen Begriffen führte zu

Schwierigkeiten. Zudem führte das gewählte Design die Benutzer nicht in der richtigen Reihenfolge zur Bearbeitung der Aufgaben.

- Die natürlichsprachliche Komponente wurde trotz ihrer großen Vorteile für Anfänger und Laien selbst bei Orientierungslosigkeit der Benutzer kaum eingesetzt.

Diese Schwächen sind von vielen anderen Systemen her bekannt. Der Rückgriff auf vorhandenes Wissen zur Softwareergonomie führt bei der Suche nach Lösungen nicht weiter. Zum einen handelt es sich dabei um rezeptartige Vorschläge in den sogenannten Styleguides (cf. Lynch/Horton 1999). Diese detaillierten Regeln führen schnell zu inhärenten Widersprüchen und sind somit nicht für den konzeptuellen Entwurf von BOF geeignet. Den Styleguides gegenüber stehen softwareergonomische Normen und kognitionspsychologische Kenntnisse auf sehr hohem Abstraktionsniveau. Diese sind zu allgemein, als dass sie sich bei der Lösung konkreter Designprobleme anwenden ließen.

Das WOB-Modell (auf der Werkzeugmetapher basierende, strikt objektorientierte graphisch-direktmanipulative Benutzungsoberflächen) ist ein Mittelmodell zwischen inkonsistenten Detailforderungen auf der Ebene der Styleguides und den abstrakten Normen und kognitionspsychologischen Theorien. Beim WOB-Modell handelt es sich um ein Bündel softwareergonomischer Vorschläge, die für Benutzer zu einer effizienten und natürlichen Handhabung von Informationssysteme führen sollen (cf. Krause 1997). Den Kern des WOB-Modells bildet die doppelte Interpretierbarkeit, die eine unterschiedliche Interpretation der gleichen Software durch Anfänger und fortgeschrittene Benutzer erlaubt. Die objektorientierte Benutzungsoberfläche präsentiert sich einem Anfänger als Formularsystem mit Dialogleitlinie. Der fortgeschrittene Benutzer erkennt nach und nach, dass die Formulare Ansichten von Werkzeugen sind, die sehr flexibel einsetzbar und parametrisierbar sind. Das WOB-Modell unterstützt kognitive Information Retrieval Strategien wie das iterative Retrieval, bei der Benutzer aufgrund der Evaluierung von Zwischenergebnissen die Anfrage verändert. Eine komprimierte und modifizierbare Anzeige der Anfrage im Ergebnismodus unterstützt iteratives Retrieval auch aus der Ergebnisansicht. Die komprimierte Ansicht kann als natürlichsprachliche oder formalsprachliche Repräsentation der Anfrage oder die Veränderung grafischer Elemente des Suchbildschirms realisiert werden. Generelle softwareergonomische Prinzipien ergänzen diese spezifischen Ansätze. Dazu gehören intelligente Komponenten, stärkere Visualisierung und die dynamische Anpassung, die bereits getroffene Entscheidungen des

Benutzers und ihre Auswirkungen an allen relevanten Stellen zur Vereinfachung der Interaktion ausnutzt (cf. Krause 1997).

Auf der Basis des WOB-Modells entstand der zweite Prototyp WING-M2, der bei Benutzertests zu wesentlich besseren Ergebnissen führte als WING-M1. In diesem Rahmen entstand das System FUZZY-WING. Die emprischen Untersuchungen zu Informationsbedürfnissen im Bereich Werkstoffinformation zeigten, dass vage Informationsbedürfnisse eine wichtige Rolle spielen. Werkstoffingenieure suchten in beispielhaften Anfragen nach Werkstoffen mit mittlerem Elastizitäts-Modul bei hoher Temperatur. Um solche Benutzerbedürfnisse zu unterstützen, wurde zunächst das vage Wissen in Kooperation mit Domänenexperten modelliert und Zugehörigkeitsfunktionen für Fuzzy Variablen definiert.

Der Benutzer kann mit einer vorgegebenen Liste von vagen Attributen Suchbedingungen spezifizieren (cf. Abbildung 2-13), ohne konkret angeben zu müssen, was z.B ein hoher Elastizitäts-Modul (E-Modul) numerisch bedeutet. Das System präsentiert als Ergebnis dann eine nach Zugehörigkeitsfunktion geordnete Liste von Werkstoffen, die in der vagen Menge der spezifizierten Bedingungen enthalten sind.

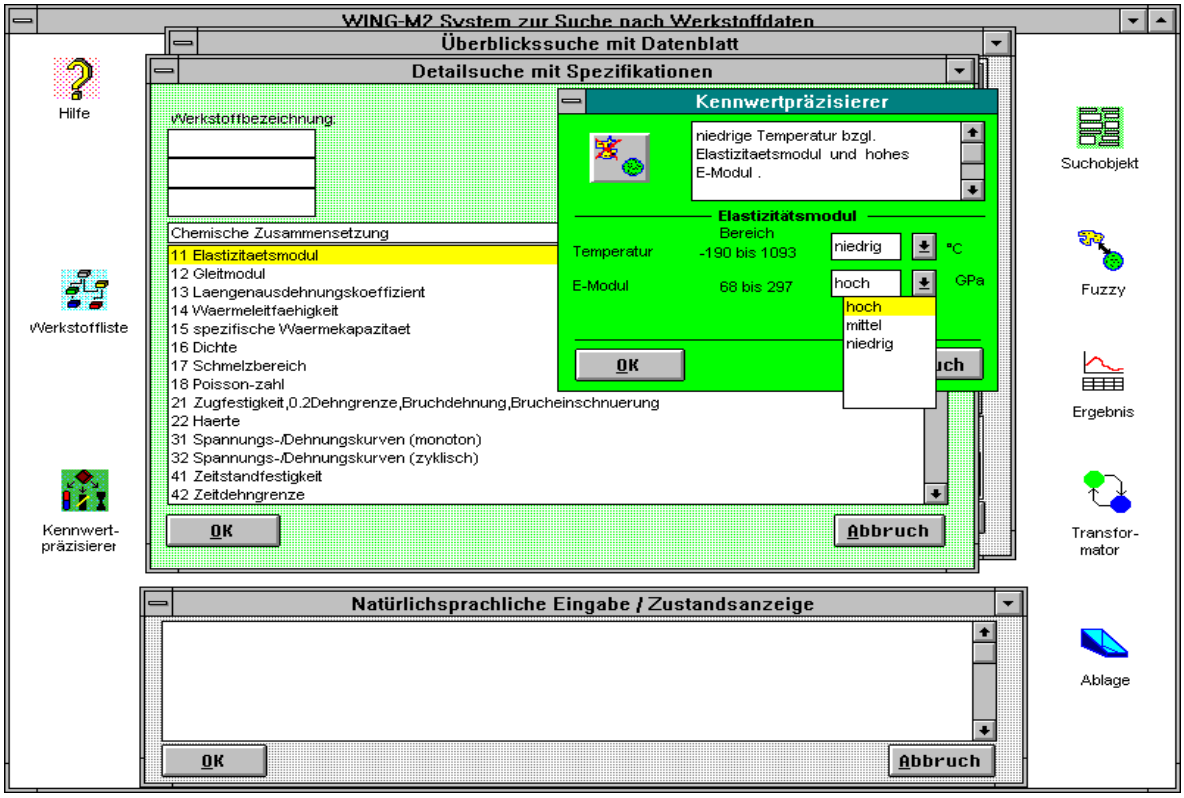


Abbildung 2-13: FUZZY-WING (aus: Womser-Hacker 1997a:196)

Der Benutzer kann auch exakte Bedingungen eingeben und sie vage und damit tolerant interpretieren lassen. Dies ist sinnvoll, um leere Antwortmengen bei Suchen mit Werkstoffprofilen zu vermeiden.

Eine andere Suchstrategie ist die Suche nach Daten zu bekannten Werkstoffen. Auch hier treten häufig leere Ergebnisse aufgrund von Datenlücken auf, da die Messungen zum Teil sehr aufwendig und teuer sind. In diesem Fall kann die Bedingung Werkstoff tolerant interpretiert und gelockert werden. Statt dem eingegebenen Werkstoff kann das System einen ähnlichen Werkstoff vorschlagen (cf. Ludwig/Mandl 1997, cf. auch Abschnitt 5.3.4.2).

WING zeigt die enge Verbindung zwischen Information Retrieval und Mensch-Maschine-Interaktion. Das softwareergonomische Design eines Informationssystems und damit die benutzergerechte und aufgabenadäquate Gestaltung bedingt auch den Entwurf entsprechender Retrieval-Modelle und einer angemessenen Retrieval Funktionalität. Dies gilt auch für das Fakten-Retrieval-System ELVIRA.

2.2.3.2 Verbandsinformationssystem ELVIRA

Ebenfalls auf der Basis des WOB-Modells wurde das Client-Server-System ELVIRA (Elektronisches Verbandsinformations-, Retrieval- und Analysesystem, gefördert vom Bundesministerium für Wirtschaft, Fördernummer IV C2-003060/22, cf. Scheinost et al. 1998) für die Recherche von statistischen Zeitreihen entwickelt. Der Schwerpunkt bei der Entwicklung von ELVIRA lag auf der softwareergonomischen Gestaltung der Benutzungsoberfläche. Die Benutzer soll möglichst schnell die für ihn relevanten Daten aus der großen Gesamtmenge recherchieren, ohne die interne Struktur aller Datenbestände zu kennen.

Inzwischen wird das ursprünglich für den Zentralverband der Elektrotechnik- und Elektronikindustrie, Frankfurt (ZVEI) entwickelte System von zwei weiteren Industrieverbänden den Mitgliedsunternehmen angeboten (Hauptverband der Deutschen Bauindustrie, Wiesbaden, HVB; Verband Deutscher Maschinen- und Anlagenbau, Frankfurt, VDMA). Nach einer Vorstellung der Benutzungsoberfläche von ELIVRA werden Anforderungen und Lösungsansätze für die Integration vager Retrieval-Komponenten vorgestellt. Kapitel 5 zur Heterogenität im Information Retrieval bespricht Probleme der heterogenen Datengrundlage von ELVIRA (cf. Abschnitt 5.1.1).

2.2.3.2.1 Die Benutzungsoberfläche von ELVIRA

Die wichtigste Herausforderung bei der softwareergonomischen Gestaltung von ELVIRA war die Menge und Komplexität der darzustellenden Daten. Die enthaltenen statistischen Zeitreihen werden anhand mehrerer hierarchischer Nomenklaturen erfasst und die Daten besitzen unterschiedliche Dimensionalität. ELVIRA löst diese Probleme v.a. durch eine Adaptierung der BOF an jede einzelne Anfrage, bei der der Gebrauch von Bildschirmplatz optimiert wird. Dazu wird das Prinzip der dynamischen Anpassung erweitert und auf Inhalt und sogar Größe von Bedienelementen übertragen (cf. Krause et al. 1998: 54ff.).

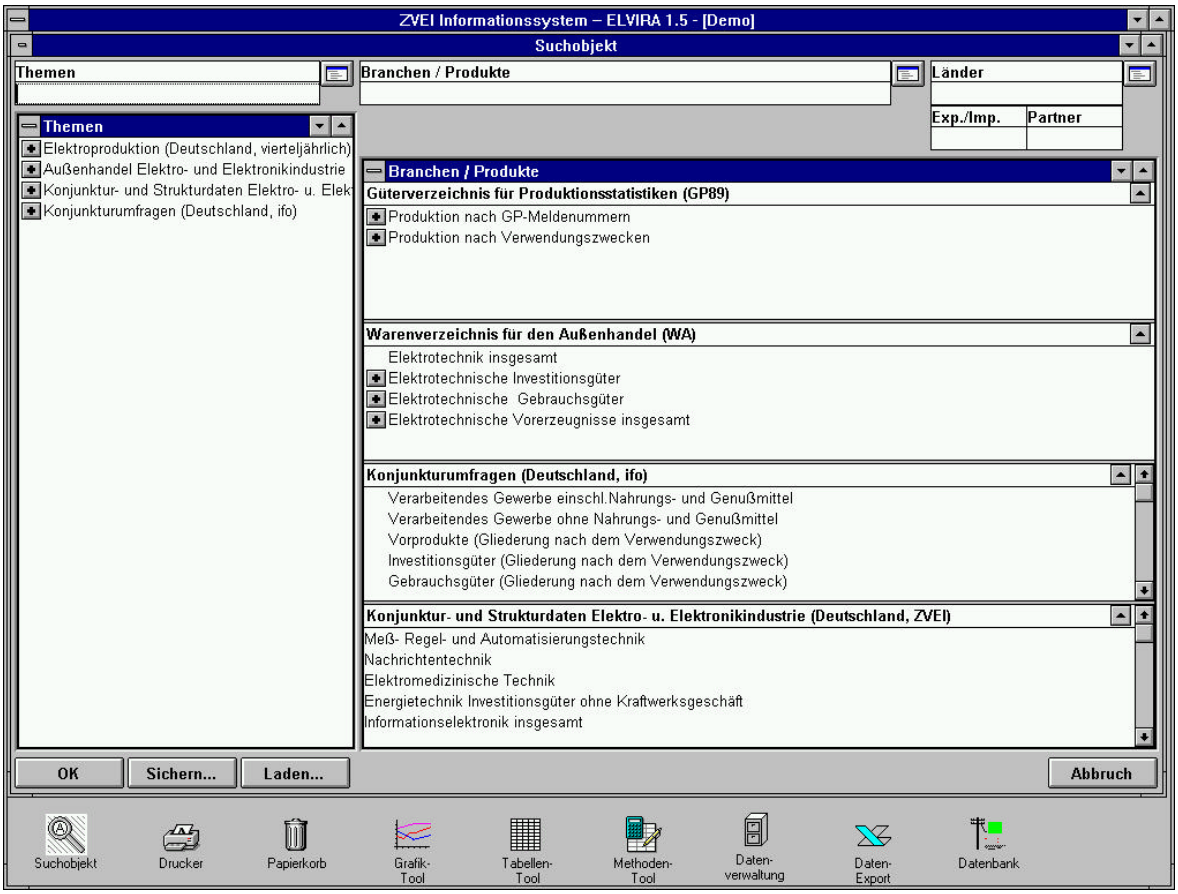


Abbildung 2-14: Eingangsbildschirm von ELVIRA

Diese Lösung demonstriert, wie dynamische Anpassung und eine modifizierbare Zustandsanzeige die Benutzbarkeit erheblich verbessern. Im Ausgangszustand sind im Hauptsuchfenster von ELVIRA Browser für zwei von drei Term-Kategorien sichtbar (cf. Abbildung 2-14). Jede Kategorie sieht eine Zustandsanzeige vor. Die Browser stellen hierachische Listen dar, aus denen der Benutzer die gewünschten Deskriptoren durch Anklicken auswählt. Der

Browser für *Produkte und Branchen* enthält vier verschiedene Nomenklaturen für die Elektroindustrie, wobei jede die spezifische Sichtweise eines Informationsanbieters repräsentiert. Der Länderbrowser ist zunächst nicht sichtbar, da vertikal nur genug Platz für zwei Browser zur Verfügung steht und Länder für den Einstieg kaum genutzt werden. Durch diese maximale Vorlageleistung kann der Benutzer seine Deskriptoren flexibel in der gewünschten Reihenfolge auswählen. Diese Anordnung erlaubt eine hohe Flexibilität und ermöglicht gleichzeitige Anfragen zu verschiedenen Themen.

Die dynamische Anpassung in ELVIRA wirkt gleichzeitig horizontal und vertikal. Sobald der Benutzer Einträge in einem Browser auswählt, passen die anderen Browser ihren Inhalt an und zeigen nur Terme die noch relevant sind. Sowohl einzelne Terme als auch ganze Nomenklaturen werden ausgeblendet wie Abbildung 2-15 zeigt. Der Benutzer hat das Thema *Produktion* gewählt und der Browser für *Produkte und Branchen* enthält nur noch die zwei der vier Listen, für die Produktion erfasst wird. Somit erhält er mehr Platz für die verbleibende Information. Dieses Prinzip wirkt von jedem Ausgangspunkt. Wird zuerst auf ein Produkt geklickt, so passt sich die Themenliste an. Dieser Mechanismus entlastet den Anwender, der die Zusammenhänge zwischen Themen und verschiedenen Einteilungen der Elektroindustrie nicht lernen muss. Auch die Anzahl potenzieller Anfragen mit leerer Ergebnismenge reduziert sich erheblich Die vertikale Anpassung in ELVIRA verbindet jeden Browser mit der Zustandsanzeige darüber, die immer die bisher in dieser Kategorie ausgewählten Terme enthält. Obwohl die selektierten Terme in der Liste invers dargestellt sind, gelangen sie beim Scrollen in den teils sehr langen Listen in den nicht sichtbaren Bereich (cf. Abbildung 2-15). Um die Merkleistung für die momentan formulierte Anfrage zu verringern, bietet ELVIRA die Zustandsanzeige. Jeder angeklickte Term wird dort in eine eigene Zeile übertragen.

Die Zahl der selektierten Einträge kann nicht von vornherein vom Entwickler festgelegt werden. Eine feste Anzahl von Zeilen in der Zustandsanzeige reicht in manchen Fällen nicht aus und führt in anderen Fällen zu einer stark suboptimalen Platzausnutzung. Die dynamische Anpassung wird daher auf die Größe der Zustandsanzeige angewandt.

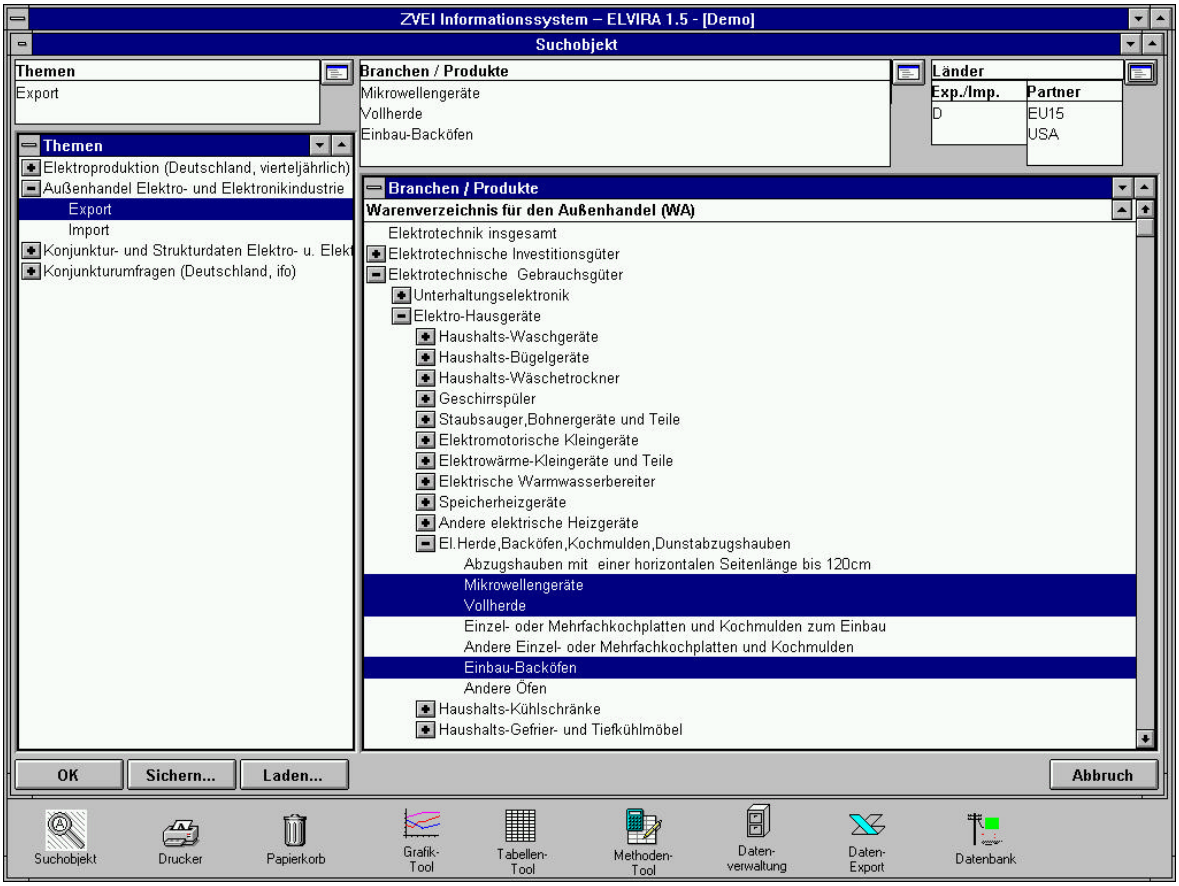


Abbildung 2-15: Anfrage in ELVIRA

Die Zustandsanzeige wächst bei jeder Selektion dynamisch (cf. Abbildung 2-15). Um den Bildschirmplatz optimal zu nutzen, verändern die Browser ihre Größe entsprechend, sie schrumpfen, wenn die Zustandsanzeige wächst und wachsen umgekehrt, um frei werdenden Platz auszunutzen. Diese effiziente Lösung bietet einen optimalen Kompromiß zwischen den sich widersprechenden Forderungen nach maximaler Vorlageleistung und dem Platzbedarf der Zustandsanzeige. Zwei Prinzipien des WOB-Modells führen so zu einer ausgewogenen Lösung in ELVIRA.

Darüberhinaus wird die Zustandsanzeige als Eingabefeld ausgenutzt, in der ein Benutzer ihm bekannte Abkürzungen für Einträge eintippen kann. Jede Zustandsanzeige besitzt immer eine leere Zeile, die als Eingabefeld dient.

Die modifizierbare Zustandsanzeige in ELVIRA ermöglicht den reduzierten Eingangsbildschirm. In der Ergebnisanzeige werden die Browser ausgeblendet und freiwerdender Platz für die Ergebnis-Dokumente ausgenutzt. Die Zustandsanzeigen bleiben erhalten, so dass die Anfrage weiterhin sichtbar ist. Um iterative Retrieval strategien zu unterstützen, bleibt die Zustandsanzeige modifizierbar wie im Eingangsbildschirm.

Das dynamische Design von ELVIRA wurde in Benutzertests in den Firmen und in einer Praxisphase von drei Jahren mit Fragebögen und Nutzertreffen validiert. Das gewählte Konzept scheint für die Aufgabe angemessen. Jedoch zeigte sich schnell, dass Zeitreihen alleine nicht alle Informationsbedürfnisse der Marktforscher abdecken. Die Integration von Textdaten führt zur Problematik der Heterogenitätsbehandlung, die in Kapitel 5 und in Abschnitt 5.1.1 für ELVIRA behandelt wird.

2.2.3.2.2 Ranking von Faktendaten

Das Ordnen von Suchergebnissen nach Relevanz (Ranking) ist eine Technik, die typischerweise im Text-Retrieval verwendet wird. Dort hat sich gezeigt, dass ein exakter Vergleich von Suchbedingungen und Dokumenten nicht immer zu befriedigenden Ergebnissen führt. Die meisten Systeme benutzen daher vage Methoden, um die Ähnlichkeit zwischen Anfrage und Dokumenten zu berechnen. Bei der Entwicklung von ELVIRA zeigte sich, dass auch in Wirtschaftsinformationen Vagheiten und Unsicherheiten vorkommen. Die folgenden Informationsbedürfnisse ergaben sich aus empirischen Untersuchungen:

1. Anfragen mit vagen Parametern
   Wie in WING (cf. Abschnitt 2.2.3.1) ist es auch im Kontext von ELVIRA sinnvoll, Anfragen mit vagen Parametern zuzulassen.
2. Ranking von Anteilen unter einem bestimmten Oberbegriff
   Um die eigene Branche im größeren Kontext einzuordnen und ihre Entwicklung zu beurteilen, ist es sinnvoll sie innerhalb der übergeordneten Branche zu betrachten.
   Beispiel: Ein Marktforscher im Bereich *Textilmaschinen* hat festgestellt, dass der ifo-Konjunkturtest für seine Branche eine Verbesserung des Geschäftsklimas ausweist. Er will wissen, ob dies für den gesamten Maschinenbau so ist und wie sich seine Branche im Vergleich zu anderen Bereichen des Maschinenbaus entwickelt.
3. Analyse von Importmärkten
   Beim Außenhandel taucht häufig die Frage nach den größten Importländern zu einem Produkt auf. Damit lässt sich der Markt in diesem Land abschätzen. In einem zweiten Schritt interessieren dann die größten Länder, die in diesen Markt exportieren.
   Beispiel: Ein deutscher Unternehmer will *Integrierte Schaltungen* auf dem amerikanischen Markt anbieten. Dazu muss er zunächst wissen, wie groß der Importmarkt in den USA ist und ob er zu den größten in der Welt ge-

hört. Ist der Markt nach seiner Einschätzung groß genug, sind die Anteile der größten Wettbewerber interessant.

Die erste Fragestellung löst ein Prototyp einer Fuzzy Retrievalkomponente. Das System erlaubt vage Anfragen in der Güterproduktionsstatistik, einem Datenbestand, der den Benutzern von ELVIRA sehr vertraut ist. Ausgangsobjekte sind ca. 500 Produktgruppen aus der Produktionsstatistik. Der Verband erfasst für sie den Umfang der Produktion und die Anzahl der betrieblichen Einheiten, die in diesem Bereich produzieren. Dabei gilt grundsätzlich: je mehr produziert wird, desto mehr betriebliche Einheiten melden in diesem Bereich. Diese Abhängigkeit stellt Abbildung 2-16 dar.

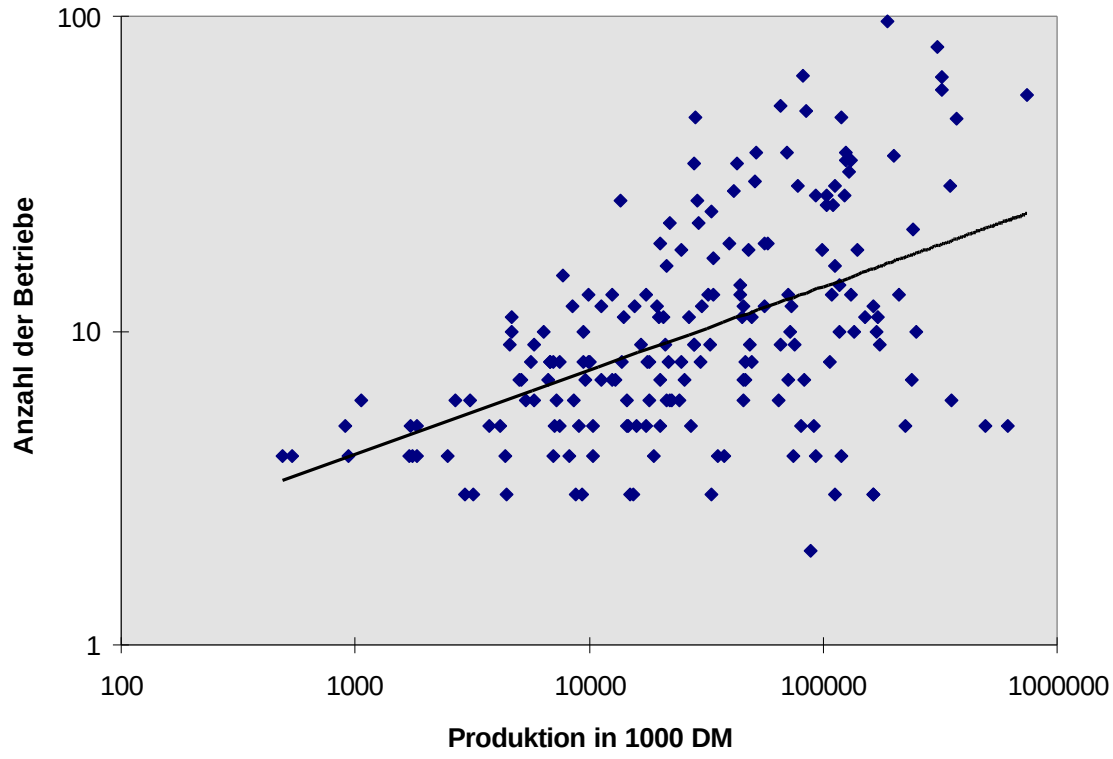


Abbildung 2-16: Verteilung für 200 Produktgruppen der Elektroindustrie mit Trendlinie

Die Streuung ist allerdings hoch, so dass es Produkte gibt, für die die Produktion sehr hoch ist und die nur von wenigen Firmen produziert werden (z.B. Hausgeräte). Dies weist auf monopolistische Verhältnisse hin. Andererseits gibt es Produkte, bei denen wenig produziert wird, aber relativ viele Firmen daran beteiligt sind. Dann liegt ein kleiner Markt vor, den sich viele Unter-

nehmen teilen. Der Prototyp der Fuzzy Retrievalkomponente erlaubt dem Benutzer den Zugriff auf solche Zusammenhänge.

Die Interpretation der Zahlen sind aufgrund der Natur der statistischen Daten teilweise problematisch. So entsprechen betriebliche Einheiten nicht immer Unternehmen und sobald nur drei oder weniger Firmen in ein Segment melden, werden die Daten aus Gründen des Datenschutzes nicht publiziert, da sich ansonsten Rückschlüsse auf einzelne Firmen ziehen ließen.

Der Prototype bildet in ELVIRA ein Objekt, das die Oberfläche für die Fuzzy-Anfrage enthält. Der Benutzer wählt in zwei Gruppen von Radio-Buttons bezüglich der Variablen *Produktion in DM* und *Anzahl der Betriebe* aus. Als Bedingung kann er jeweils *hoch* und *niedrig* wählen. Ein Gamma-Operator verknüpft die Bedingungen (cf. Abschnitt 2.2.1).

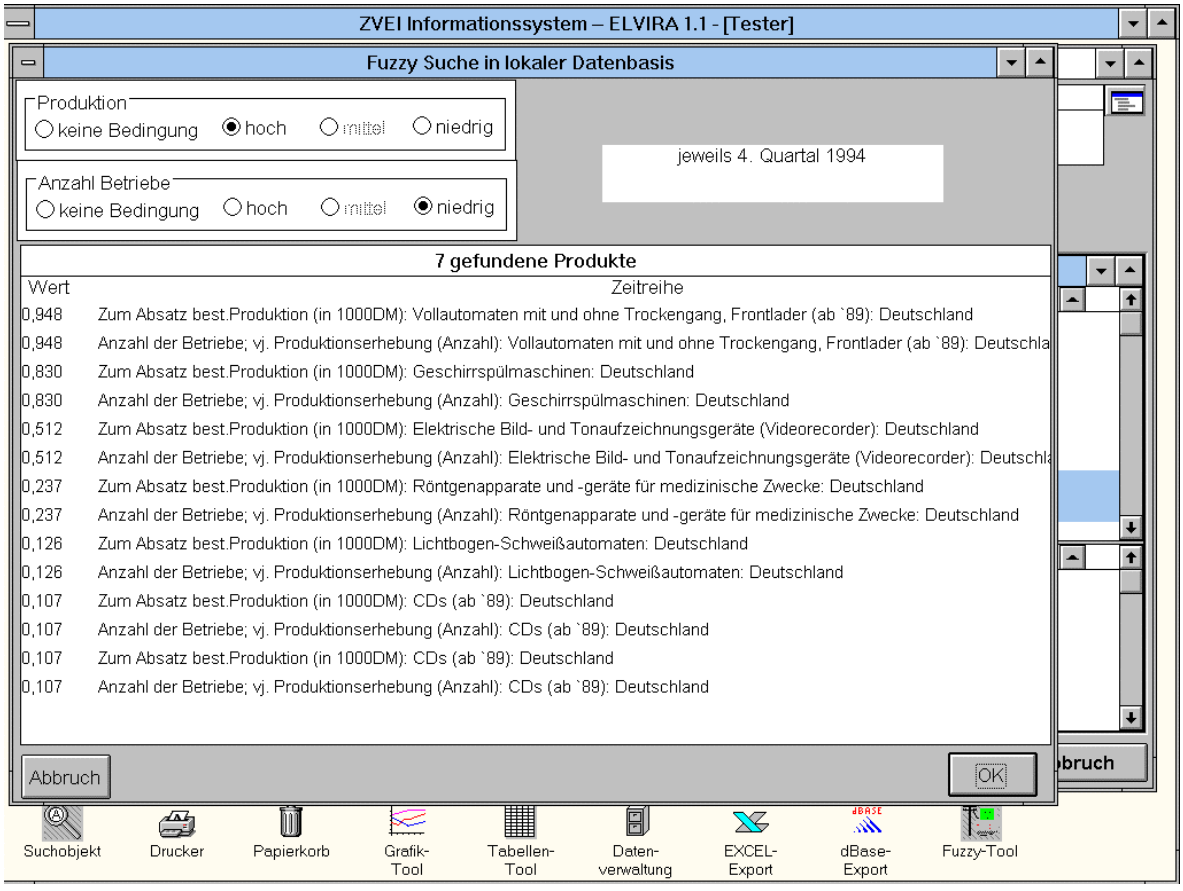


Abbildung 2-17: Fuzzy-Komponente mit Anfrage und Ergebnis

Nach Klick auf OK wird der Ergebnisbereich des Fensters mit einer geordneten Liste gefüllt. Dabei steht rechts eine Zeitreihenüberschrift und links davon ein Faktor, der angibt, wie gut dieses Produkt die Bedingungen erfüllt. Er kann als Fuzzy-Zugehörigkeitswert zu der vom Benutzer definierten unschar-

fen Menge und damit als Ranking-Wert betrachtet werden. Für jedes Produkt erscheint sowohl die Zeitreihe Produktion in DM als auch Anzahl der Betriebe in der Ergebnisliste (cf. Abbildung 2-17). Diese Zeitreihenobjekte können nun etwa per drag-and-drop auf das Grafik-Tool gezogen und so dargestellt werden (cf. Abbildung 2-18).

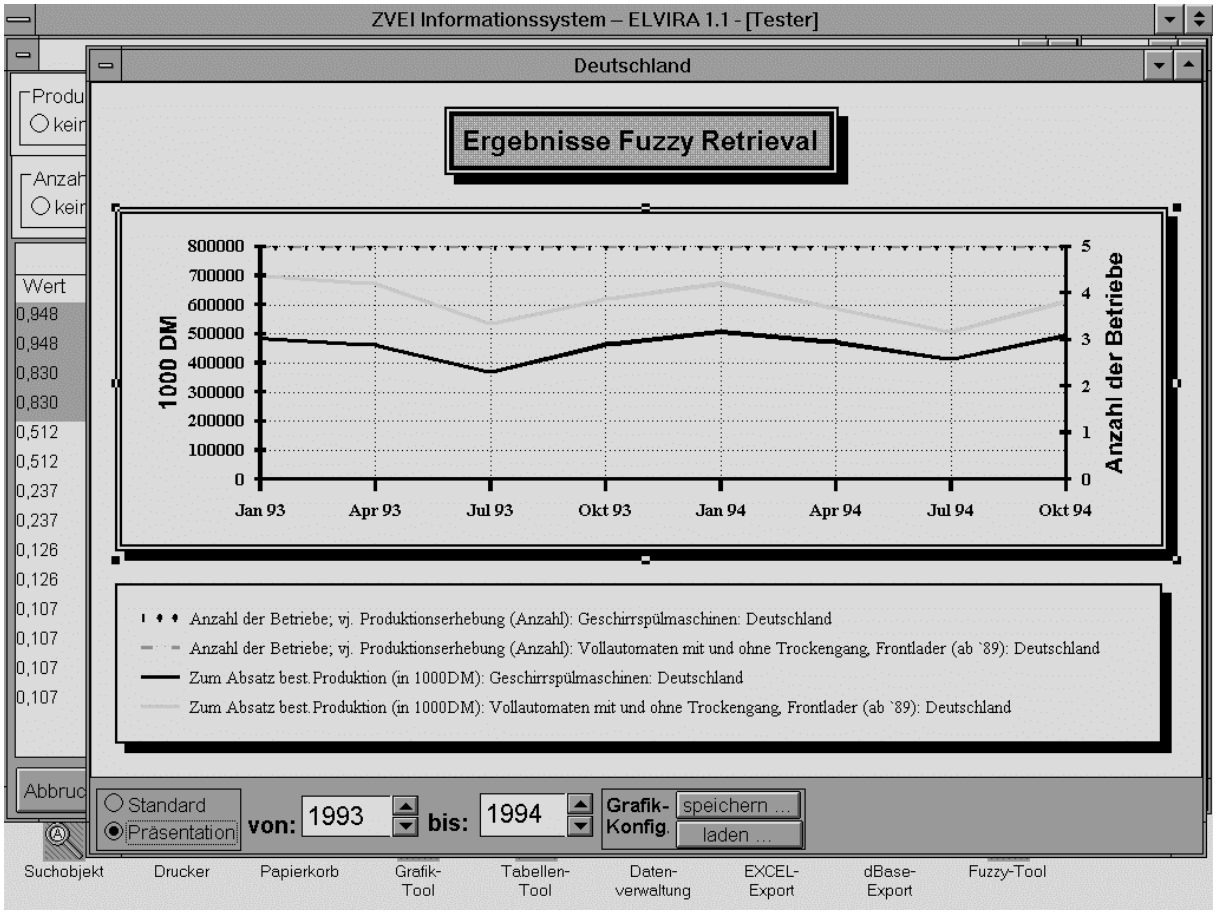


Abbildung 2-18: Grafische Darstellung von gefundenen Zeitreihen im ELVIRA Grafik-Tool

Diese Funktionalität ist besonders interessant für Benutzer, die sich einen Überblick über die gesamte Elektroindustrie verschaffen möchten. Dabei müssen sie nicht wissen, was z.B. *Anzahl der Betriebe = hoch* in Zahlen bedeutet. Die Erweiterung auf andere Variablen ist möglich. Ebenso besteht die Möglichkeit, die Suche auf einen Teil der Produktnomenklatur, wie etwa *Antriebstechnik* einzuschränken (cf. Mandl 1998).

Die Fragestellungen 2 und 3 lassen sich in ELVIRA zwar bearbeiten, erfordern aber umständliche Bedienschritte. Ein zusätzliches Werkzeug erlaubt die Lösung in wenigen Schritten. Für die erste Fragestellung muss eine Zeitreihe ausgewählt werden. Falls die Branche der Zeitreihe noch Unterebenen hat, werden diese angezeigt. Für die Auswahl nur einer Zeitreihe ist die Zustands-

anzeige des ELVIRA-Suchobjekts nicht nötig. Deshalb wurde ein spezielles Werkzeug implementiert, in dem durch Auswahl aus einem Themenbrowser und aus einem Branchenbrowser eine Zeitreihe spezifiziert wird (cf. Abbildung 2-19). Als Land wird Deutschland eingesetzt. Alternativ kann diese Einstellung des Tools auch über Drag-and-Drop erfolgen. Da nur eine Zeitreihe ausgewählt wird, wurden die Browser mit Zustandsanzeige als einfache DropDown-Listboxen mit hierarchischer Anzeige realisiert. Das Ergebnis besteht aus einer Liste von Branchen. Die Reihenfolge wird aufgrund des angegebenen Sortierkriteriums festgelegt (cf. Abbildung 2-20). Hierbei sind absolute Größe und Veränderungsrate sinnvoll, wobei bisher nur absolute Größe realisiert ist.

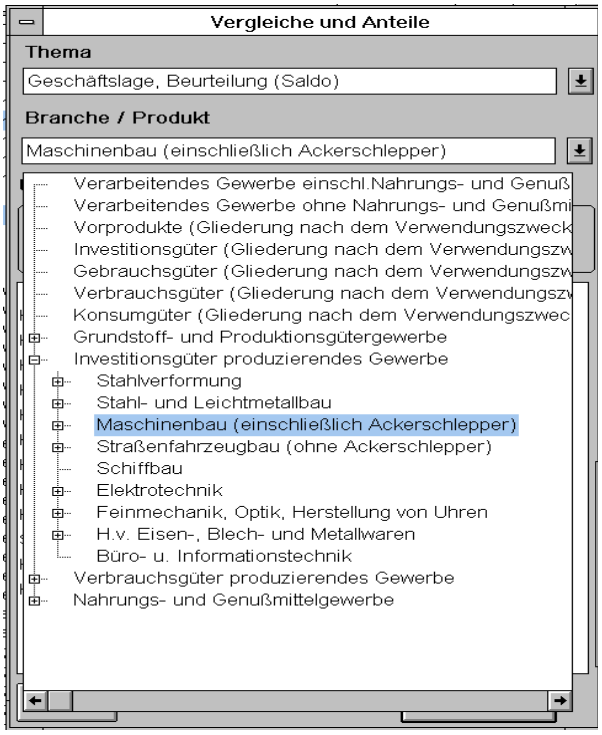


Abbildung 2-19: Auswahl einer Branche

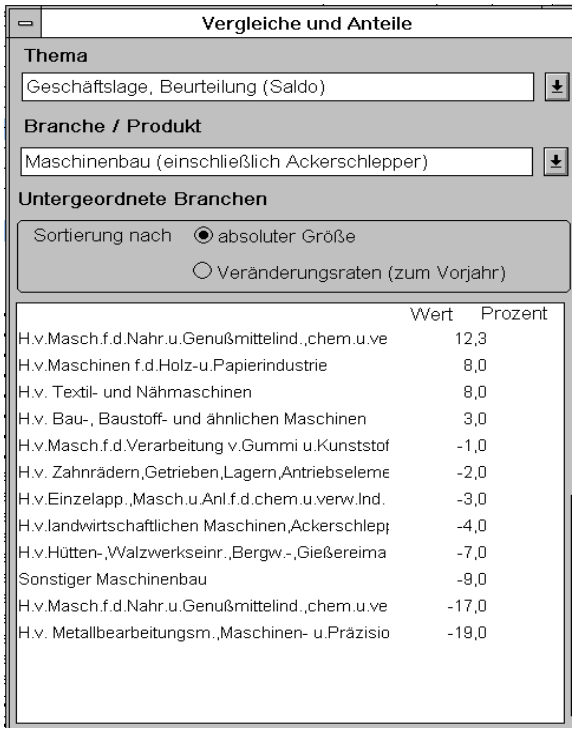


Abbildung 2-20: Sortierte Anzeige der untergeordneten Branchen

Das System zeigt nicht die vollständigen Zeitreihen, da Vergleiche nur einen zeitlichen Querschnitt erfordern. Zieht ein Benutzer das Bedienelement mit der sortierten Liste auf die Grafik, so erstellt ELVIRA eine Balken- oder Tortengrafik.

Der Prototyp enthält auch die Suche nach den größten Ländern für den Import oder Export eines Produkts. Sobald ein Außenhandelsthema und eine Branche

aus dem Warenverzeichnis für die Außenhandelsstatistik (WA) ausgewählt wurden, erscheinen rechts zwei zusätzliche Listen. Die obere zeigt die größten Berichtsländer für die Kombination von Thema und Branche. Der Benutzer markiert ein Land, das ihn näher interessiert. Darauf werden in der zweiten Liste die größten Partnerländer gezeigt, die an diesem Warenstrom beteiligt sind. Auch dieses Tool bewerteten Benutzer in informellen Tests sehr positiv. In einer kommerziellen Implementierung müsste das Problem der Suche nach den größten Ländern technisch gelöst werden. Hierzu können Heuristiken eingesetzt werden, bei denen zunächst alle Länder mit allgemein großem Handelsvolumen durchsucht werden. Weiterhin kann ein Suchalgorithmus die hierarchische Gliederung der Länder ausnutzen, so dass zunächst nur die Werte für die Kontinente geprüft werden. Dann verfolgt das System nur die größten Knoten weiter (cf. Mandl/Stempfhuber 1998:156).

Die Entwicklung von ELVIRA demonstriert, wie das pragmatische Postulat der Informationswissenschaft (cf. Kuhlen 1999) im IR eingelöst werden kann und geht damit auch konform mit der Definition der Fachgruppe IR (cf. erster Abschnitt in Kapitel 2). Sowohl die Möglichkeit vager Fragestellungen als auch die Gestaltung der Mensch-Maschine-Interaktion müssen aus Sicht des Benutzers und seiner Anforderungen bearbeitet werden.

2.3 Ansätze zur Verbessserung von Information Retrieval Systemen

Die Systeme WING und ELVIRA begegnen Problemen bei der Benutzung von Informationssystemen durch aufgaben- und benutzerangepasste Gestaltung. Darin liegt häufig großes Potenzial zur Verbesserung. Daneben haben u.a. die Ergebnisse von TREC gezeigt, dass auch die Qualität beim Retrieval noch verbessert werden kann. Die Suche nach neuen Modellen oder die Verbesserung bestehender Modelle ist nach wie vor sinnvoll. Die folgenden Abschnitte fassen einige der auffälligsten Schwächen und Ansatzmöglichkeiten für Verbesserungen zusammen.

Die Schwächen beginnen bereits bei der im Text-Retrieval üblichen Indexierung. Einzelne Terme können nicht die Semantik eines Textes widerspiegeln und damit nicht ein Dokument repräsentieren. Wie in Abschnitt 2.1.3 gezeigt, ist die Ähnlichkeitsberechnung ein Problem, da die üblicherweise verwendeten Modelle die Komplexität der menschlichen Ähnlichkeitsberechnung formal nicht abbilden können. Weitere Schwächen bestehender IR-Systeme liegen in den Bereichen Adaptivität und Heterogenität, wie die folgenden Abschnitte zeigen.

2.3.1 Adaptivität

Eine Möglichkeit, die Prozesse im Information Retrieval kognitiv adäquater zu gestalten, besteht in Adaptivität, der selbständigen Anpassung der Systeme und Algorithmen an die Eigenschaften und Präferenzen eines Benutzers oder an die Anforderungen einer spezifischen Retrievalsituation. Adaptierbarkeit hingegen ist die Möglichkeit des Benutzers, das System seinen Anforderungen anzupassen.

Adaptivität ist in Information Retrieval Systemen meist sehr gering ausgeprägt. Wenn Systeme überhaupt lernen, dann meist vor dem eigentlichen Einsatz. Während der Nutzung findet dann keine Anpasssung mehr statt.

2.3.1.1 Relevanz-Feedback

Ein häufig untersuchter Ansatz zur Erhöhung der von Adaptivität im Information Retrieval ist Relevanz-Feedback., das sich u.a. in TREC sehr bewährt hat. Wie bereits bei den Modellen in Abschnitt 2.1.2.2 und 2.1.2.3 diskutiert, beginnt Relevanz-Feedback mit der Bewertung einiger Dokumente durch den Benutzer. Aus den Repräsentationen der positiv und negativ beurteilten Dokumente modifiziert das System die Anfrage (cf. Harman 1992). Relevanz-Feedback gilt als eines der besten Verfahren zur Verbesserung der Qualität von IR-Systemen ist (cf. Womser-Hacker 1997:147ff.).

Relevanz-Feedback bezieht die Adaptivität auf eine Anfrage. In einigen Modell-Varianten führt die Bewertung des Benutzers jedoch zu Änderungen der Repräsentationen und damit zu Änderungen am Modell. Allerdings sind die so erreichten Modifikationen gering, da immer nur die Werte für Terme verändert werden, die in den als relevant gekennzeichneten Dokumenten vorkommen.

2.3.1.2 Das MIMOR-Modell

Das MIMOR-Modell (Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval, cf. Womser-Hacker 1997) ist ein Ansatz zur Erhöhung der Adptivität auf einer Meta-Ebene. MIMOR basiert auf Ergebnissen von TREC und arbeitet mit Mehrfachindexierung. Die TREC-Studien (cf. Abschnitt 2.1.4.2) haben u.a. gezeigt, dass die besten IR-Verfahren sich in der Qualität nur unwesentlich unterscheiden. Ihre Treffermengen weisen jedoch nur wenig Überschneidung auf. Das bedeutet, dass alle Verfahren etwa gleich viele relevante Dokumente finden, aber eben verschiedene. Sogenannte Mehrfachindexierungs- und Fusionsansätzen versuchen dies auszunutzen und die relevanten Dokumente mehrerer Ansätze zu verei-

nigen. Das Problem besteht darin, die relevanten Dokumente in der Gesamtergebnismenge zu finden und stärker zu gewichten (cf. z.B. Voorhees et al. 1995, Lee 1995, Bartell et al. 1994).

MIMOR lernt, die Fusion durch Relevanz-Feedback zu steuern und kombiniert zwei wichtige Ergebnisse aus TREC zu einer Strategie. Bei der Fusion lernt das System, die Ergebnisse der einzelnen Systeme zu kombinieren. Jedes System trägt mit einem bestimmten Gewicht zum Gesamtergebnis bei. Dieses Gewicht erhöht sich, wenn ein einzelnes System häufig positiv bewertete Dokumente auf die gesamte Ergebnisliste bringt.

In Womser-Hacker/Mandl 1999 wird die Adaptivität von MIMOR auf Benutzer- und Dokumenteigenschaften ausgeweitet. Cluster von Dokumenten mit gemeinsamen Eigenschaften erhalten spezifische Gewichte. So kann MIMOR positive Eigenschaften von Systemen für bestimmte Dokumente berücksichtigen.

MIMOR profitiert von einer künstlich geschaffenen Heterogenität. Es nutzt verschieden erzeugte Repräsentationen der gleichen Objekte.

2.3.2 Heterogenität

Für Benutzer ist es wünschenswert und sinnvoll, Daten verschiedener Medialität und aus unterschiedlichen Quellen, die eventuell auch unterschiedlich inhaltlich erschlossen sind, mit einer Anfrage zu suchen. Dies ist z.B. bei sogenannten Data Warehouses wichtig, in denen Organisationen ihre gesamten Daten speichern, verwalten, integrieren und auswerten (cf. Anahory/Murray 1997).

Dadurch entsteht systemseitig Heterogenität, die Information Retrieval Systeme vor große Probleme stellt. Da diese Problematik von besonderer Bedeutung ist, wird sie ausführlich in Kapitel 5 behandelt. Vorher werden in Kapitel 3 Grundlagen neuronaler Netze eingeführt, die u.a. zur Behandlung von Heterogenität eingesetzt werden.

2.4 Fazit: Grundlagen des Information Retrieval

Die geschilderten Probleme von Information Retrieval Systemen lassen sich unter dem Schlagwort *mangelnde Toleranz* zusammenfassen.

- Heterogenität erfordert Toleranz gegenüber verschiedenen Datenquellen und Datentypen.

- Adaptivität bedeutet Toleranz gegenüber Benutzereigenschaften.
  Die starre mathematische Ähnlichkeitsberechnung sollte durch an den Eigenschaften der menschlichen Ähnlichkeitsbeurteilung ausgerichtete Rechenverfahren abgelöst oder ergänzt werden. Die Realisierung höherer Toleranz sollte auf bewährte Verfahren zur Vagheitsbehandlung zurückgreifen. Dazu gehören insbesondere neuronale Netze, die das folgende Kapitel bespricht.

Demnach soll ein neues IR-Modell wie das in Kapitel 1 vorgestellte COSIMIR-Modell oder das Transformations-Netzwerk versuchen, tolerant auf alle Anforderungen zu reagieren. Wie die Ergebnisse aus TREC (cf. Abschnitt 2.1.4.2) zeigen, muss ein neues IR-System nicht unbedingt die beste Leistung zeigen, um als erfolgreich zu gelten. Vielmehr führt oft auch ein Beitrag zu einem Fusionsverfahren zu einer Verbesserung des Gesamtergebnisses. Großes Potenzial für Verbesserungen verspricht der Bereich Soft-Computing, der die notwendige vage Informationsverarbeitung leistet.

3 Grundlagen neuronaler Netze

Künstliche neuronale Netze sind lernfähige Systeme, die Information tolerant und robust verarbeiten. Wie ihr natürliches Vorbild - die Nervensysteme von Lebewesen -bestehen sie aus zahlreichen einfachen, miteinander verknüpften Prozessoren. Über ihre Verknüpfungen tauschen diese Prozessoren oder Neuronen numerische Informationen in Form von Aktivierung aus. Folgendes Zitat formuliert dieses Prinzip sehr prägnant:

> "Die Informationsverarbeitung geschieht durch eine große Anzahl von relativ einfachen Prozessoren, die in einem dichten Netzwerk miteinander verbunden sind. Diese Prozessoren (auch Units genannt) arbeiten lokal, jeder für sich allein, und kommunizieren mit anderen Units nur via Signale, die sie über die Verbindungen senden." (Dorffner 1991: 16)

Künstliche neuronale Netze eignen sich für die Simulation menschlicher Fähigkeiten im Bereich Perzeption und Kognition, bei denen starre, regelverarbeitende Systeme nicht zum Erfolg führen. Im Folgenden werden sie einfach als neuronale Netze bezeichnet, da in dieser Arbeit zwischen natürlichen und künstlichen Netzen keine Verwechslungsgefahr besteht. In der Regel ist immer von künstlichen neuronalen Netzen die Rede.

Einige Beispiele für die Anwendung neuronaler Netze sind Steuerung autonomer Fahrzeuge (cf. Zell 1994:541ff.), Erkennung von handgeschriebenen Postleitzahlen (cf. de Waard 1994), Vorhersage von Aktienkursen (cf. Refenes/Azema-Barac 1994), Analyse von Herztönen (cf. Alonso-Betanzos et al. 1999) und Bestimmung der Toxität von Flüssigkeiten (cf. Grauel et al. 1999).

Dieses Kapitel bietet eine kurze Einführung in die Architektur und Funktionsweise neuronaler Netze. Nach einer Darstellung des natürlichen Vorbilds und der kognitionswissenschaftlichen Aspekte folgt eine Vorstellung von Backpropagation-Netzwerken. Backpropagation ist der am häufigsten eingesetzte Netzwerktyp und ein typischer Vertreter lernfähiger Systeme. Das COSIMIR-Modell und das Transformations-Netzwerk beruhen auf dem Backpropagation-Modell. Daran schließt eine systematische Einführung der Grundlagen neuronaler Netze und verschiedener Modelle an. Dabei liegt der Schwerpunkt auf den Modellen, die für den state-of-the-art Neuronale Netze im Information Retrieval (cf. Kapitel 4) wichtig sind. Im weiteren Verlauf

wird das Backpropagation-Modell erneut aufgegriffen und ausführlicher diskutiert.

Als Einführung in neuronale Netze eignet sich Scherer 1997. Einen ausführlichen systematischen Überblick bietet Zell 1994. Rojas 1993 diskutiert formale und algorithmische Grundlagen.

3.1 Natürliche neuronale Netze

Das Vorbild für die Informationsverarbeitung in (künstlichen) neuronalen Netzen ist die Architektur biologischer Nervenzellenverbände, die aus sehr vielen stark vernetzten Einheiten besteht. Diese Neuronen können Impulse von anderen Neuronen über Nervenbahnen aufnehmen und eventuell selbst *feuern*, also elektrische Impulse an andere Neuronen weiterleiten. Die folgende Darstellung greift auf Scherer 1997 und Zell 1994 zurück.

Nervenzellen bestehen aus dem Zellkern, der Nervenfaser (Axon) und den Dendriten, länglichen Verbindungsleitungen. Signale kommen über die Dendriten in der Zelle und laufen über das Axon an andere Zellen weiter. Die Verbindungsstelle zwischen Axon und Dendrit ist die Synapse. Entlang der Nervenbahnen Axon und Dendriten verlaufen Signale oder Impulse in elektronischer Form durch Veränderung des Potenzials zwischen den Zelleninneren und dem Zellenäußeren. Die Stärke eines Signals ist dabei immer gleich. Die Stärke eines Reizes wird also nur durch die Frequenz kodiert.

An den Synapsen verbreitert sich das Axon. Es endet kurz vor der nächsten Nervenzelle, wodurch ein Spalt entsteht. Diese Lücke überbrückt das elektrische Signal auf chemischem Weg durch Auslösung sogenannter Neurotransmitter. Diese Moleküle werden am Ende des Axons ausgelöst und auf der anderen Seite des Spalts von Rezeptoren aufgenommen. In der anderen Nervenzelle läuft das Signal in elektrischer Form weiter. Erreichen die in einer Zelle ankommenden Signale einen Schwellenwert, so schickt sie einen Impuls ausgehend vom Ausgangspunkt des Axons entlang einer Nervenbahn. Die Veränderung der Durchlässigkeit von Synapsen bildet eine der Grundlagen für das Lernen von Lebewesen.

Messungen zur Geschwindigkeit dieser biologischen Vorgänge zeigen, dass das Gehirn im Vergleich zu einem seriellen Computer äußerst langsam arbeitet (cf. Kinnebrock 1992:11). Jedoch aktiviert das Gehirn immer sehr viele Neuronen gleichzeitig. Aufgrund dieser Parallelverarbeitung übertrifft das menschliche Gehirn den Computer bei den meisten kognitiven Fähigkeiten.

3.2 Kognitionswissenschaftliche Aspekte

Auch wenn die künstlichen Neuronen stark von ihrem natürlichem Vorbild abweichen, so sind neuronale Netze ein plausibleres Modell für Kognition als ein serieller Rechner. Deshalb haben neuronale Netze auch in der Kognitionswissenschaft ein neues Paradigma geschaffen: den Konnektionismus.

Bis zum Erfolgszug der neuronalen Netze seit Mitte der 80er Jahre galt das *Physical Symbol System* (PSS) als angemessenes Modell für Kognition und damit auch für Künstliche Intelligenz. Das PSS besagt, dass Symbolverarbeitung notwendig und hinreichend für Intelligenz ist. Nach der PSS-Hypothese, wie sie in Newell/Simon 1976 manifestiert ist und etwa in Newell et al. 1989 wieder aufgegriffen wird, besteht Kognition aus der Manipulation von Symbolen. Intelligenz entsteht demnach durch die Verarbeitung von Zeichen, die symbolhaft auf etwas anderes verweisen.

Dagegen treten in neuronalen Netzen Prozesse auf, die sich einzeln nicht symbolisch interpretieren lassen, sondern erst im Zusammenhang mit vielen anderen Verarbeitungsschritten sinnvoll sind. Im Gegensatz zum PSS wird Information sub-symbolisch verarbeitet.

Smolensky 1988 formuliert einen integrativen Ansatz. Er führt das sub-symbolische Paradigma ein und setzt es als "intuitive processor" dem "conscious rule interpreter" gegenüber (Smolensky 1988:4f.). Demnach besteht Kognition nicht im Abarbeiten von symbolischen Regeln, sondern die Vorgänge im Gehirn wie auch die in neuronalen Netzen liefen auf einer tieferen Ebene ab. Insbesondere intuitives Expertenwissen lasse sich nicht als Regelsystem modellieren.

Zwar ergibt sich an der Oberfläche oft regelfolgendes Verhalten, tatsächliche Intelligenz entstehe aber nur unter dem Level der Symbole. Somit sind die Abläufe in neuronalen Netzen auch nicht interpretierbar wie etwa ein Computerprogramm in einer prozeduralen Programmiersprache, und viele der Neuronen repräsentieren keineswegs Entitäten der realen Welt.

Smolensky 1988 sieht das sub-symbolische Paradigma als Basis kognitiver Vorgänge. Intuition lässt sich sub-symbolisch erklären und durch das Zusammenspiel zahlreicher einfacher Prozesse entstehen auf einer höheren Ebene regelfolgende Systeme (cf. Smolensky 1988, Dorffner 1991).

3.3 Das Backpropagation-Modell: ein erster Überblick

Um komplexe Probleme zu lösen, arbeiten serielle Computer meist mit einer Menge von Regeln. So kennt ein Schachcomputer die erlaubten Züge und einige Strategie-Regeln und berechnet daraus die besten Züge.

Viele Probleme sind jedoch nicht so gut formalisierbar wie Schach. Besonders bei Expertenentscheidungen ist oft schwer oder unmöglich, die zugrundliegenden Regeln widerspruchsfrei zu formulieren. Dies gilt z.B. für finanzielle oder medizinische Diagnoseprobleme. Ein Finanzexperte schätzt aufgrund seiner Expertise zwar die Kreditwürdigkeit eines Kunden gut ein, er kann aber nur schwer allgemeingültige Regeln für diesen Vorgang angeben. Der Experte trifft seine Entscheidung durch eine ganzheitliche Beurteilung der Daten zu einer Person. Ebenso verlässt sich ein Arzt bei einer Diagnose kaum auf ein einzelnes Symptom oder eine einzige Messung. Vielmehr bewertet er die relevanten Daten ganzheitlich.

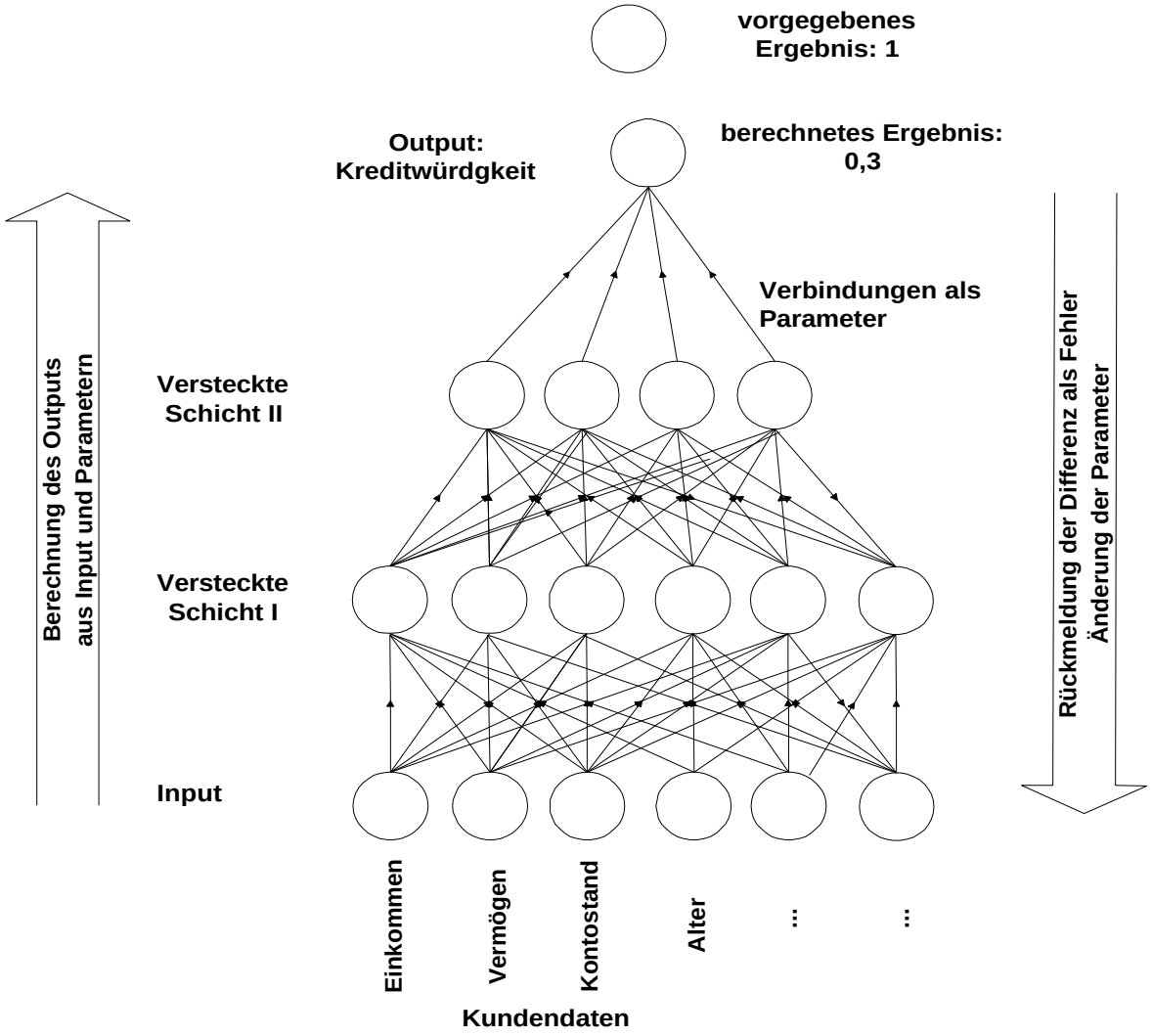


Abbildung 3-1: Funktionsweise eines Backpropagation-Netzwerks

Experten, denen es schwer fällt, ihr intuitives Wissen in Regeln zu formulieren, erwerben es durch Erfahrung, also durch die Kenntnis vieler

Beispiele. Durch das Analysieren von Beispielen ist das Wissen auch wieder zugänglich. In dieser Form erhalten auch viele lernende Systeme Expertenwissen. Anstatt wie bei einem typischen Schachprogramm Regeln anzugeben, sind diese Systeme mit zahlreichen Beispielen für die gewünschte Abbildung von Symptomen auf Entscheidungen ausgestattet.

Backpropagation-Netzwerke sind ein typischer Vertreter solcher lernender Systeme. Sie lernen anhand von Beispielen komplexe Funktionen und können dabei ähnlich wie Experten ihr Wissen nicht in Form von Wenn-dann-Regeln ausgeben. Der Benutzer eines Netzwerks weiss also in den meisten Fällen nicht, *warum* ein Netzwerk eine Entscheidung trifft. Diesen Nachteil nimmt man aber in Kauf, wenn ansonsten keine Modellierung möglich scheint. Backpropagation-Netzwerke bestehen aus zahlreichen Neuronen, die in Schichten angeordnet sind. Eine Schicht dient dem Input, dann folgen Schichten für die Berechnung von Zwischenstufen und schließlich gelangt die Aktivierung in eine Output-Schicht. Input und Output bilden definierte Schnittstellen zur Welt, dort werden Daten angelegt und abgelesen. Bei einen Netz zur Beurteilung der Kreditwürdigkeit repräsentieren die Input-Neuronen z.B. die Daten eines Kunden. Ein Neuron steht z.B. für den momentanen Kontostand, ein anderes z.B. für das Alter. Bei der Eingabe eines neuen Kunden wird das entsprechende Neuron auf den jeweiligen Wert gesetzt.

Welches Wissen hier eingeht, hängt vom Anwendungsfall ab. Oft entschieden Experten, welche Daten für das Abbildungsproblem erforderlich sind. Der Output steht für das gewünschte Ergebnis. So kann im Beispiel die Kreditwürdigkeit zwischen Null und Eins liegen, wobei ein höherer Wert eine hohe Kreditwürdigkeit bedeutet.

Zwischen den Neuronen der verschiedenen Schichten befinden sich gewichtete Verbindungen. Sie bilden die Parameter des Netzes, die anfangs zufallsgesteuert initialisiert und beim Lernen richtig eingestellt werden. Das Lernen verläuft in zwei Schritten. Zunächst berechnet das Netz nach Eingabe der Daten ein Ergebnis im Output. Da noch nichts gelernt wurde, ist dieses Ergebnis sicher falsch. D.h., es stimmt nicht mit dem Wert überein, den ein Experte als Beispiel vorgegeben hat. Die Differenz zwischen Ergebnis und Vorgabe misst den Fehler des Netzes.

Im zweiten Schritt wird dieser Fehler vom Output in Richtung Input, also gewissermaßen rückwärts ins Netz gespeist. Nun verändern sich die Werte der Verbindungen. Sie stellen sich so ein, dass sich der Fehler für dieses Trainingsbeispiel etwas verringert. Dies wird nun für alle Beispiele häufig wiederholt. Bei Erfolg zeigt sich, dass der Fehler immer kleiner wird und das Netz schließlich die gewünschte Funktion lernt.

Ein Backpropagation-Netzwerk ist demnach eine Funktion mit sehr vielen Parametern, die aus einem Input einen Output berechnet. Diese Paramter werden zu Beginn zufallsgesteuert eingestellt. Anders als bei einem Schachprogramm ist weder die Anzahl, die Bedeutung noch der *richtige* Wert dieser Parameter bekannt. Durch die kontinuierliche Präsentation von Beispielen, stellt das Netz die Parameter so ein, dass sich aus den Input-Daten der richtige Output ergibt. Jede Verbindung kann als eine Mikro-Regel betrachtet werden, die jedoch für sich alleine keinen Sinn ergibt. Nur im Zusammenspiel aller Verbindungen entsteht die richtige Funktion.

Nach dem Erlernen einer Funktion anhand von vorliegenden Beispielen, erfolgt der Einsatz des Netzes. Nun berechnet es aus unbekannten Input-Daten den Output ab. Im Beispiel bestimmt es die Kreditwürdigkeit neuer Kunden.

3.4 Aufbau und Funktionsweise

Die Bestandteile neuronaler Netze fasst ein erstmals von Rumelhart/McClelland 1986 vorgestellter Rahmen zusammen, der sich gut für die Kategorisierung eignet. Die folgende Darstellung orientiert sich an diesem Rahmen.

3.4.1 Neuronen

Jedes Modell besteht aus einer Menge von Prozessoren, die in der Gehirnmetapher den Neuronen entsprechen. Diese Prozessoren werden auch Knoten oder Units genannt. Sie gruppieren sich häufig in Mengen oder Schichten mit ähnlichen Eigenschaften. Neuronen repräsentieren Konzepte, Eigenschaften oder *Micro-Features* und sind symbolisch nicht interpretierbar.

- Aktivierungszustände
  Die Neuronen oder Knoten besitzen Aktivierungszustände (a_n in Abbildung 3-2). Jedes Neuron hat zu einem Zeitpunkt eine bestimmte numerische Aktivierung. Die Aktivierungszustände aller Knoten und damit der Aktivierungszustand des gesamten Netzes beschreibt bei einem Netz mit n Neuronen ein n-stelliger Vektor. In der Praxis werden die Aktivierungszustände jedoch häufig auf das Intervall von 0 bis 1 normalisiert. In der Gehirnmetapher entspricht der Aktivierungszustand des Netzes dem Kurzzeitgedächtnis. Die Aktivierung entspricht meist dem Output (o_n in Abbildung 3-2) eines Neurons.

- Aktivierungsfunktion
  Jedes Neuron berechnet lokal seine Aktivierung (a_n in Abbildung 3-2) nach einer definierten Aktivierungsfunktion aus der alten Aktivierung und aus dem Input, den das Netz liefert. Teilweise verwenden die Modelle Schwellwertfunktionen. Ein Neuron leitet in der Regel nur dann Aktivierung weiter („feuert“), wenn der gesamte Input (net-input$_n$ in Abbildung 3-2) eine bestimmte Schwelle überschreitet. Für viele Netzwerkmodelle wie das Backpropagation-Modell (cf. Abschnitt 3.5.4) sind komplexe Aktivierungsfunktionen erforderlich.

- Ausbreitungsfunktion
  Diese Funktion errechnet das Netzeingangssignal eines Knoten aus den Ausgabesignalen der mit ihm verbundenen Knoten und den Gewichten der dazwischenliegenden Verbindungen. Das Eingangssignal für alle Units lässt sich wieder als n-stelliger Vektor interpretieren und wird meist einfach als Summe der Produkte zwischen Gewichtung und Output-Signal berechnet.

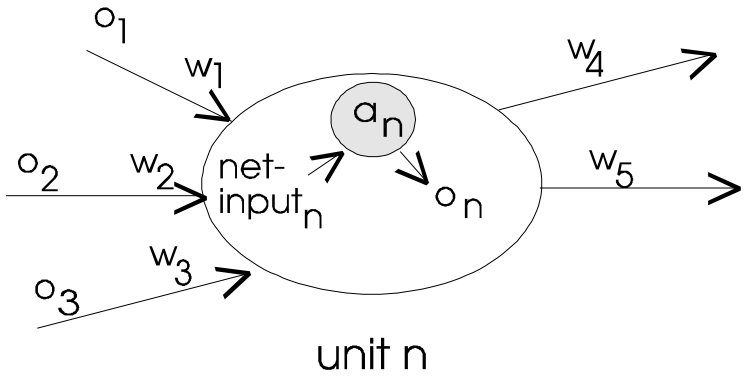


Abbildung 3-2: Die Funktionsweise eines künstlichen Neurons: (cf. Dorffner 1991: 17)

o_i:	Output der Unit i
w :	Gewicht einer Verbindung
net-input$_n$:	Input in Unit n
a_n:	Aktivierung der Unit n

3.4.2 Vernetzung

Die Neuronen sind durch gewichtete Verbindungen vernetzt. Bei einem Netz mit n Neuronen entsprechen diese Gewichtungen einer n x n - Matrix. Die Verbindungsmatrix legt die Architektur des Netzwerks fest. Im Hopfield-Modell sind alle Werte der n x n - Matrix belegt. In vielen anderen Modellen sind die Neuronen in Schichten gruppiert, wobei innerhalb der Schichten keine Verbindungen erlaubt sind. Manche Architekturen beschränken Verbindungen auf eine Richtung, andere verfügen über bidirektionale Links. Beim Backpropagation-Netzwerk laufen die Verbindungen nur in eine Richtung, damit steht für die Gewichte auch nur eine Hälfte der n x n – Matrix zur Verfügung.

Die Werte der Verbindungsgewichte können in der Regel sowohl positive als auch negative Werte annehmen.

3.4.3 Lernregel

Die wichtigste Eigenschaft neuronaler Netze ist die Selbstorganisation oder das Lernen. Dabei verändert das Netz die Stärke der Verbindungen und damit ihre Durchlässigkeit. Die Aktivierungsausbreitung berechnet aus einem Input den Output, während das Lernen diese Abbildungsfunktion verändert.

Lernverfahren sind das wichtigste Unterscheidungsmerkmal für neuronale Netze. Die häufigsten Typen sind überwachte und unüberwachte Verfahren. Bei unüberwachten Modellen analysiert das Netz nur die Daten selbst, während beim überwachten Lernen der gewünschte Output als Teacher benutzt wird. Dieser kann in Form von beispielhaften Zuordnungen vorliegen.

Ein dritter Typ ist das Reinforcement Lernen, das zwischen den beiden anderen liegt. Der gewünschte Output wird nicht exakt, sondern nur als graduelle oder vage Entscheidung vorgegeben. Das Netz erfährt gewissermaßen nur die Richtung, in die es sich entwickeln soll.

Die Hebb'sche Lernregel ist eine typische Vertreterin des unüberwachten Lernens. Sie besagt, dass häufig benutzte Verbindungen gestärkt und selten benutzte geschwächt werden (*Use it or lose it!*). Bei überwachten Lernverfahren muss die erwünschte Ausgabe für einige Eingaben bekannt sein. Aus der Differenz zwischen tatsächlichem und erwünschtem Output ergibt sich ein Fehler. Anhand dieses Fehlers versucht das Netz nun seine Verbindungen so einzustellen, dass der Fehler kleiner und die gewünschte Ausgabe somit besser erreicht wird. Die entsprechende Lernregel wird als Delta-Regel bezeichnet, da die Differenz die entscheidende Rolle spielt. Die allgemeinste Form bestimmt die notwendige Änderung an der Gewichtungsstärke zwischen

zwei Units und berücksichtigt neben dem Fehler die Aktivierung des empfangenden Neurons vor der Verbindung:

$$\text{Allgemeine Delta - Regel}$$

$$\Delta w_{ij} = h \; \text{Output}_i \; (Teacher_j - Aktivierung_i)$$

h Lernrate

cf. Zell 1994:85

Überwachtes Lernen ähnelt Näherungsverfahren, die Funktionen nach Vorgabe einiger Punkte möglichst genau eingrenzen. Das am häufigsten benutzte Modell ist der Backpropagation-Algorithmus (cf. 3.5.4).

3.4.4 Schnittstelle zur Umgebung

Neuronale Netze besitzen mit den Input- und Output-Neuronen Schnittstellen nach außen. Bei der Eingabe wird die Aktivierung der Input-Neuronen von außen gesetzt. Die Aktivierungszustände von Output-Neuronen bilden die Ausgabe.

Input-Daten werden meist vorverarbeitet und durch lineare oder nicht-lineare Transformationen dabei auf das Intervall zwischen Null und Eins normalisiert. Die adäquate Vorverarbeitung entscheidet oft über die Qualität eines Netzes.

3.5 Modelle

Der im letzten Abschnitt aufgespannte Rahmen erlaubt als Ausprägungen die verschiedensten neuronalen Netzwerkmodelle.

3.5.1 Kohonen-Netze

Kohonen-Netze sind ein unüberwachter Lernalgorithmus, der Strukturen in Daten aufdeckt. Sie werden auch als selbstorganisierende Karten (Self Organizing Maps, SOM) bezeichnet (cf. Kohonen 1984, Scherer 1997:93ff., Zell 1994:179ff.). Kohonen-Netze bestehen aus zwei Schichten, einer Eingabe-Schicht und einer Kohonen- oder Ausgabe-Schicht, in der eine topologische Karte der Daten entsteht. Die Anzahl der Neuronen in der Eingabe-Schicht

ergibt sich aus der Dimensionalität der Eingangsdaten. Die Struktur der Kohonen-Schicht wird vom Anwender vorgegeben. Eine zwei- oder dreidimensionale Anordnung unterstützt die spätere Visualisierbarkeit der Daten. Abbildung 3-3 zeigt ein Kohonen-Netzwerk mit zweidimensionaler Anordnung der Output-Neuronen. Jede Unit der Eingabe-Schicht ist mit allen Units der Kohonen-Schicht verbunden, so dass jedem Kohonen-Neuron ein Gewichtsvektor von der Größe des Eingabe-Vektors zugeordnet ist. Zudem sind die Kohonen-Neuronen untereinander verbunden.

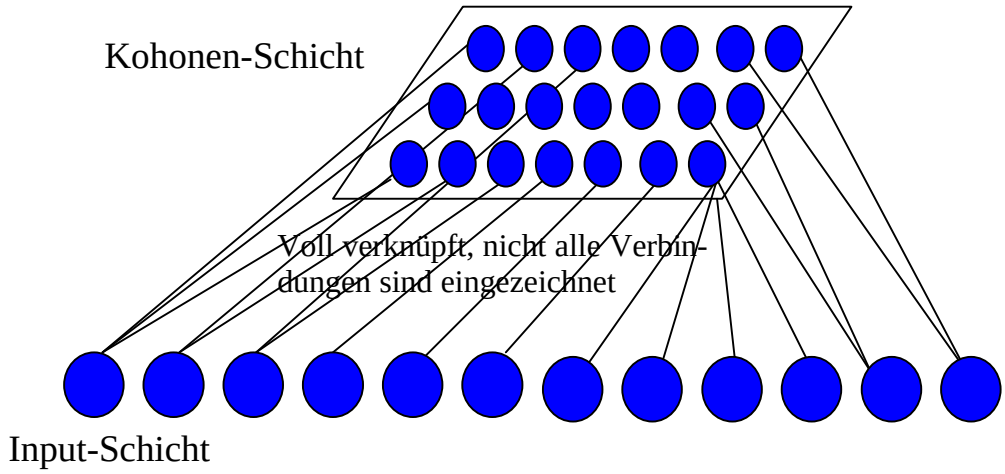


Abbildung 3-3: Schematisches Kohonen-Netzwerk

Die Eingabe-Vektoren werden mit den Gewichtsvektoren verglichen. In der Regel ist die Euklidische Distanz das Ähnlichkeitsmaß. Das Neuron mit der geringsten Distanz oder der höchsten Ähnlichkeit zum Eingabemuster gewinnt und erhält die gesamte Aktivierung. Die Gewichtungen des Gewinner-Neurons in die Eingabe-Schicht werden so modifiziert, dass die Ähnlichkeit weiter steigt. Geometrisch betrachtet verschiebt der Algorithmus den Gewichtsvektor in Richtung des Eingabevektors. Soweit arbeitet das Kohonen-Netzwerk wie andere Clustering-Verfahren. Um die topologische Struktur zu erzeugen, verändern sich auch die Gewichtsvektoren der Nachbar-Neuronen des Gewinners. Dies erfordert eine Definition von Nachbarschaft, die verschiedene Funktionen wie etwa die Gauss-Funktion oder der Kosinus implementieren. Diese Funktionen liefern ein Maß für die Entfernung jedes Neurons in der Kohonen-Schicht vom Gewinner-Neuron, das die Intensität der Gewichtsänderung beeinflusst. Je näher ein Neuron dem aktivierten Neuron ist, desto stärker wird sein Gewichtsvektor adaptiert. Die Vektoren sehr naher Neuronen werden somit immer in ähnliche Richtungen verschoben. Dadurch entstehen Cluster, in die ähnliche Muster abgebildet werden.

Kohonen-Netzwerke werden u.a. in der Spracherkennung für Sprechererkennung, für Bildverarbeitung in der Medizin und für die Extraktion der Eigenschaften von Chromosomen eingesetzt (cf. Kohonen 1997a).

3.5.2 Adaptive Resonance Theory (ART)

Adaptive Resonance Theory (ART) ist eine Familie von mathematisch elaborierten Netzwerken, die Inputmuster in Cluster einteilen. Ein besonderes Merkmal von ART gegenüber Kohonen-Netzen ist die zweifache Überprüfung der Ähnlichkeit zwischen Input und Zielcluster. ART I verarbeitet nur binäre Vektoren. ART II ist eine Erweiterung für Muster mit reellen Zahlen. ART III zieht biologische Vorgänge beim Übergang eines Nervensignals auf eine andere Zelle mit in die Modellierung ein und versucht so, kognitive Vorgänge adäquater zu modellieren. Parameter sind etwa die prä- und postsynaptischen Mengen von Neurotransmittern. ARTMAP ist eine überwacht lernende Variante des Clustering-Verfahrens, bei der die korrekte Einteilung der Muster vorgegeben wird. Mehr Informationen zu ART bietet Zell 1994.

Die Grundidee von ART beruht auf der Beobachtung, dass alle anderen Lernverfahren nicht angemessen auf Änderungen in einer Lernmenge reagieren. Dies ist biologisch nicht plausibel, da natürliche Nervensysteme neu auftretenden Muster flexibel verarbeiten, sie in den Wissensbestand integrieren und die bisherigen Erfahrungen dabei nicht vollständig überschreiben.

Zunächst klassifiziert ART einen Input-Vektor im Wesentlichen wie im Kohonen-Netz in einer Winner-take-all-Schicht. Dann überprüft eine weitere Schicht von Neuronen (die Vergleichsschicht), inwieweit die Zuordnung angemessen ist, indem sie mit einer anderen Ähnlichkeitsfunktion prüft, ob die Ähnlichkeit einen vorgegebenen Schwellenwert übersteigt. Erreicht die Ähnlichkeit nicht das geforderte Maß, wird das Muster zur erneuten Klassifikation an die Erkennungsschicht zurückverwiesen. Findet das ART-Netz kein befriedigendes Cluster, so fügt es ein neues Neuron in der Erkennungsschicht hinzu, dessen Cluster für den Eingabe-Vektor zuständig ist. Wie im Kohonen-Netz wird nach jeder Zuordnung der Clustervektor in Richtung des Eingabemusters verändert.

Dieses Verfahren erlaubt es, bereits gelernte Zuordnungen von Mustern zu Clustern nicht zu überschreiben und sie so nicht zu *vergessen*.

3.5.3 Assoziativspeicher

Hopfield-Netze sind unüberwacht lernende Netzwerke, die meist als assoziativer Speicher dienen. Sie reagieren auf unvollständige oder gestörte Muster mit sinnvollen Ausgaben (cf. Zell 1994:197ff., Scherer 1997:125ff.). Ein assoziativer Speicher ruft ein gespeichertes Muster direkt über den Inhalt ab und nicht über einen Index wie traditionelle Datenbanken. Hopfield-Netze verarbeiten meist binäre Muster.

Hopfield-Netze bestehen aus einer untereinander voll vernetzten Schicht von Neuronen und besitzen symmetrische Verbindungen. Alle Neuronen dienen sowohl als Input als auch als Output. In der Retrievalphase wird ein Muster angelegt und einem der gespeicherten Muster zugeordnet. Dabei läuft die Aktivierungsausbreitung wie in anderen Netzen und das Netz konvergiert zu einem Zustand, der einem der gespeicherten Muster entspricht. Als Stoppkriterium wird die Änderung der Aktivierungsmatrix zwischen zwei Schritten berechnet und mit einem Schwellwert verglichen.

Um zu erreichen, dass das Netz immer zu einem der gespeicherten Muster konvergiert, stellt ein Algorithmus die Verbindungen entsprechend ein. Das Hopfield-Netz weicht hierbei von den meisten anderen Netzwerk-Modellen ab und lernt die Gewichte nicht schrittweise, sondern berechnet sie aus den zu speichernden Mustern. Dazu ist eine Energie-Funktion definiert, die das Energie-Niveau des Netzes ausdrückt:

$$E = -\frac{1}{2}\sum_{i}\sum_{J\neq i} w_{ij} akt_i akt_j + \sum_i akt_i \Theta_i$$

Scherer 1997:129

Dieses Niveau wird durch die Aktivierungsausbreitung minimiert. Dabei summiert jedes Neuron die gewichteten Output-Werte jedes anderen Neurons, mit dem es in Verbindung steht. Dieser Input löst die binäre Aktivierungsfunktion ein, wenn ein bestimmter Schwellenwert überschritten wird.

Die Initialisierung stellt sicher, dass jedes Minimum dieser Energie-Funktion eines der Muster repräsentiert.

$$w_{ij} = \sum_s w_{ij}^s = \sum_s x_i^s x_j^s$$

s Index über die Muster

Scherer 1997:131

Das Hopfield-Netz konvergiert nicht immer zu einem gespeicherten Muster, sondern läuft oft in ein lokales Minimum der Energie-Funktion. Da das Lernverfahren immer versucht, die Energie zu minimieren, kann es das lokale Minimum nicht mehr verlassen. Dem versucht eine Variante des Hopfield-Netzwerks entgegenzuwirken, die sogenannte Boltzmann-Maschine (cf. Zell 1994: 207 ff.).

Boltzmann-Maschine und Hopfield-Netz unterscheiden sich nur durch die Aktivierungsfunktion und das Lernverfahren, das bei der Boltzmann-Maschine nicht deterministisch sondern stochastisch abläuft. Die Lernfunktion enthält einen zufallsgesteuerten Faktor. Damit wird die Energie des Netzwerks in einem Schritt nicht unbedingt verkleinert, sondern kann auch steigen. Erreicht das Verfahren ein lokales Minimum, dann erreicht die Hopfield-Regel keine Verbesserung mehr. Der zufallsgesteuerte Beitrag zur Aktivierungsfunktion ermöglicht auch die Erhöhung des Fehlers bei einem einzelnen Lernschritt. Steckt ein Netz in einem lokalen Fehlerminimum, erlaubt der Boltzmann-Lernalgorithmus durch die vorübergehende Erhöhung des Fehlers das Verlassen des Minimums. Diese Hoffnung ist durchaus plausibel, da die Wahrscheinlichkeit einer Erhöhung des Fehlers mit der Anzahl der Trainingsschritte steigt.

Das Lernverfahren minimiert wie beim Hopfield-Netz die gesamte Energie im Netz. Die Boltzmann-Netze entwickeln aber die Energie-Metapher weiter und übertragen sie auf das Auskühlen eines glühenden Metalls. Das Lernverfahren wird dieser Metapher folgend als *simulated annealing* bezeichnet. Das Kühlen eines Metalls erfolgt bei einem Kristallisationsvorgang sehr langsam, um allen Molekülen die Möglichkeit zu geben, sich optimal auszurichten.

Die Energie des Netzes berechnet sich nach der Summe aller Aktivierungen und aus den paarweise zwischen den Neuronen herrschenden Energien, die sich aus den beiden Aktivierungen und dem Gewicht der Verbindung ergibt:

$$E = -\sum w_{ij}\, o_i\, o_j + \sum q_i\, o_j$$

o_i *Output = Aktivierung von Neuron i*

w_{ij} *Gewicht der Verbindung zwischen Neuron i und j*

q_i *Schwellenwert von Neuron i*

Zell 1994:208

Die Wahrscheinlichkeit, dass ein Neuron aktiviert ist, sinkt mit der im System enthaltenen Gesamtenergie. Demnach wird ein Neuron mit folgender Wahrscheinlichkeit aktiviert:

$$p_k = p(o_k \equiv 1) = \frac{1}{1 + e^{-\Delta E / T}}$$

ΔE *Energiedifferenz zwischen zwei Zuständen*

T *Temperatur−Parameter*

Zell 1994:209

Der Parameter T wird zwischen den Schritten langsam reduziert. Das Lernverfahren für die Boltzmann-Maschine unterscheidet zwischen sichtbaren und versteckten Neuronen. Dabei sind sichtbare Neuronen am Input und Output beteiligt, während versteckte Neuronen nur der internen Verarbeitung dienen. Beim Lernen wird zunächst ein Mustervektor angelegt, wobei die Aktivierung der Eingabe-Neuronen konstant bleibt. Nachdem ein Energiegleichgewicht im gesamten Netz erreicht ist, spielen die Aktivierungen der versteckten Neuronen die entscheidende Rolle für das Lernen. Nun verändern sich die Verbindungsgewichte so, dass diese Aktivierungen auch ohne externen Input das gleiche Aktivierungsmuster in den sichtbaren Neuronen induzieren. Dazu modifiziert eine lokale Lernregel die Verbindungen zwischen sichtbaren und versteckten Neuronen.

Die Forschung im Bereich Assoziativspeicher befasst sich v.a. mit den Fragen, wie gut das Retrieval bei unvollständigen Mustern arbeitet und mit ihrer Speicherkapazität (cf. Jagota et al. 1995, Schwenker et al. 1996). Eine interessante Modifikation stellen Perfetti/Massarelli 1997 vor. Das Verfahren lernt nicht die Verbindungsmatrix, sondern eine davon durch Datenkompression abgeleitete Matrix. Die Autoren erzeugen dazu einen Raum über die Eigen-Werte der Verbindungsmatrix und transformieren diese nach dem Training wieder in die Verbindungen. Dadurch wird das Lernproblem verkleinert. Das Verfahren beruht auf den gleichen Grundlagen wie Latent Semantic Indexing (cf. Abschnitt 2.1.2.4.3).

3.5.4 Backpropagation-Netze

In den 60er Jahren galt das Perzeptron als ein erfolgversprechendes Netz, das beliebige Beziehungen zwischen Input- und Output-Mustern lernen kann.

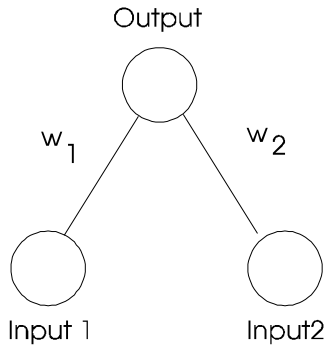


Abbildung 3-4: Aufbau eines Perzeptrons (McClelland/Rumelhart 1988:124)

Ein Perzeptron besteht aus zwei verbundenen Schichten. Beim Lernen wird an der Input-Schicht ein Vektor angelegt und daraus der Output berechnet. Dieser Ausgabevektor wird mit dem gewünschten Zielvektor verglichen. Durch Anwendung der Delta-Regel (cf. Abschnitt 3.4.3) versucht das Perzeptron die durch die Beispiele vorgegebenen Funktionen zu lernen. Alle Muster liegen dabei mehrmals am Input des Perzeptron an. Eine Phase, in der alle Muster einmal präsentiert werden, heisst Epoche. Dieses einfache Verfahren findet immer eine Lösung, wenn eine existiert. Die Lösung besteht in einer Menge von Gewichtungen, die dafür sorgen, dass das Netz den Input richtig auf den Output abbildet (cf. McClelland/Rumelhart 1988:123).

3.5.4.1 Vom Perzeptron zum Backpropagation-Netzwerk

Gegen Ende der 60er Jahre analysierten Minsky/Papert (1969) in ihrem Buch *Perceptrons* formal die Mächtigkeit des Lernverfahrens und konnten zeigen, dass das Perzeptron nur linear trennbare Funktionen lernen kann. Da bereits eine so einfache Funktion wie das logische XOR (Entweder-Oder) nicht linear trennbar ist, lernt das Perzeptron nur eher einfache Funktionen.

Lineare Trennbarkeit bedeutet anschaulich, dass sich in einem zweidimensionalen Koordinatensystem eine Gerade zwischen die Punkte der verschiedenen zu lernenden Klassen ziehen lässt. Abbildung 3-5 zeigt, dass zwar die logischen Funktionen AND und OR linear trennbar sind, aber nicht XOR.

Vermutlich sind zahlreiche reale Probleme nicht linear trennbar (cf. Zell 1994:100f.). Minsky/Papert (1969) hatten bereits erkannt, dass eine Zwischen-Schicht im Perzeptron diese Beschränkung aufhebt. Cybenko (1989) zeigt, dass eine versteckte Schicht prinzipiell ausreicht, um jede kontinuierliche Funktion anzunähern.

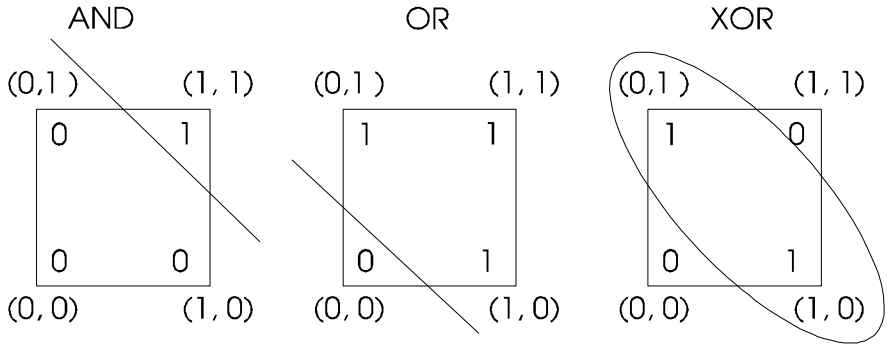


Abbildung 3-5: Lineare Trennbarkeit (McClelland/Rumelhart 1988:124)

Jedoch existierte für ein dreistufiges Netzwerk keine Lernregel. In den 70er und 80er Jahren wurde ein Verfahren entdeckt, das den Fehler auch über eine Zwischen-Schicht zurückverfolgt. Dieser Backpropagation-Algorithmus löst auch nicht linear trennbare Probleme wie etwa die XOR-Funktion. Ein entsprechendes Backpropagation-Netz besteht also aus mindestens drei Schichten, wobei die Schichten zwischen Input und Output versteckte oder *Hidden Layer* heißen.

Für Gewichtsveränderungen von der Zwischen-Schicht in Richtung Input-Schicht wird der gewünschte Output aus der Summe der Produkte aus Fehler und entsprechender Gewichtung in der darüber liegenden Schicht gemittelt. Das Fehler-Signal für alle Neuronen, die nicht Output-Neuronen sind, folgt aus den Fehlern der oberen Schichten:

$$d_{pj} = f_j'(net_{pj}) \sum_k d_{pk} w_{kj}$$

Rumelhart et al. 1986b:329

Ein Faktor ist die Ableitung der Aktivierungsfunktion, die die Vorwärtsschritte definiert. Für ein Backpropagation-Netzwerk muss die Aktivierungsfunktion damit an jeder Stelle ableitbar sein. Im Perzeptron wird dagegen die neurobiologisch plausible Schwellenwertfunktion benutzt, welche die Ableitungsbedingung nicht an jeder Stelle erfüllt. Backpropagation-Netzwerke setzen z.B. die Sigmoid-Funktion ein:

$$aktivierung_i = \frac{1}{1 + e^{-netinput_i \cdot const.}}$$

cf. Rumelhart et al. 1986b:329

Konvergiert der Fehler in einem Netz während des Trainings, so hat es die gewünschte Abbildung gelernt. Der Fehler zwischen Output und gewünschtem Ergebnis dient als Maß für die Qualität der Abbildung.

3.5.4.2 Probleme und Lösungsansätze

Der Backpropagation-Algorithmus besitzt einige Schwächen, die bei der praktischen Anwendung berücksichtigt werden müssen.

- Die Trainingszeiten sind oft sehr lang.
- Es ist nicht garantiert, dass der Algorithmus das globale Fehlerminimum findet. Da der Fehler bei jedem Schritt minimiert wird, ist es möglich, dass nur ein lokales Minimum erreicht wird (cf. Zell 1994:112f.). Das gleiche Problem tritt beim Hopfield-Modell auf (cf. Abschnitt 3.5.3).
  Dies lässt sich für die Veränderung nur eines Gewichts gut zweidimensional veranschaulichen. In Abbildung 3-6 steuert der Algorithmus nur berab, und bleibt im lokalen Minimum B stecken, obwohl das gobale Minium des Fehlers bei C eine wesentlich bessere Lösung darstellt.

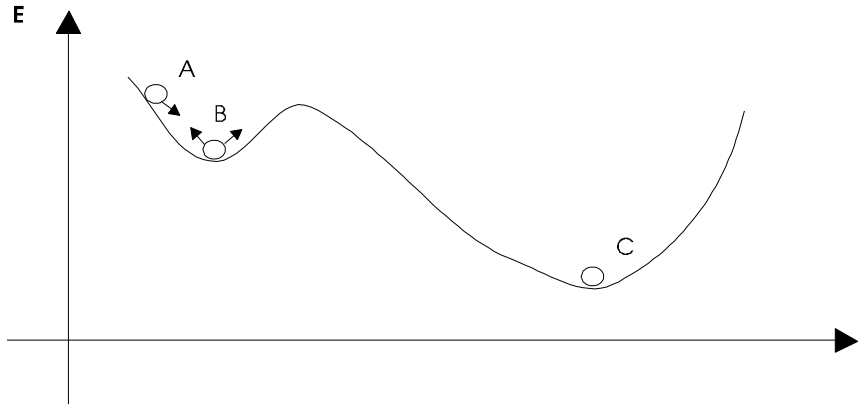


Abbildung 3-6: Fehlerverlauf in Abhängigkeit von einem Gewicht (cf. Dorffner 1991:111)

- Die Entwicklung und der Lernprozess erfordern zahlreiche heuristische Parametersetzungen, von denen im Folgenden einige diskutiert werden:

  - Ende des Trainings (*Overlearning)*

    Verfolgt man die Fehlerentwicklung in der Trainings- und Testmenge während des gesamten Trainings, so ergibt sich etwa folgendes Bild:

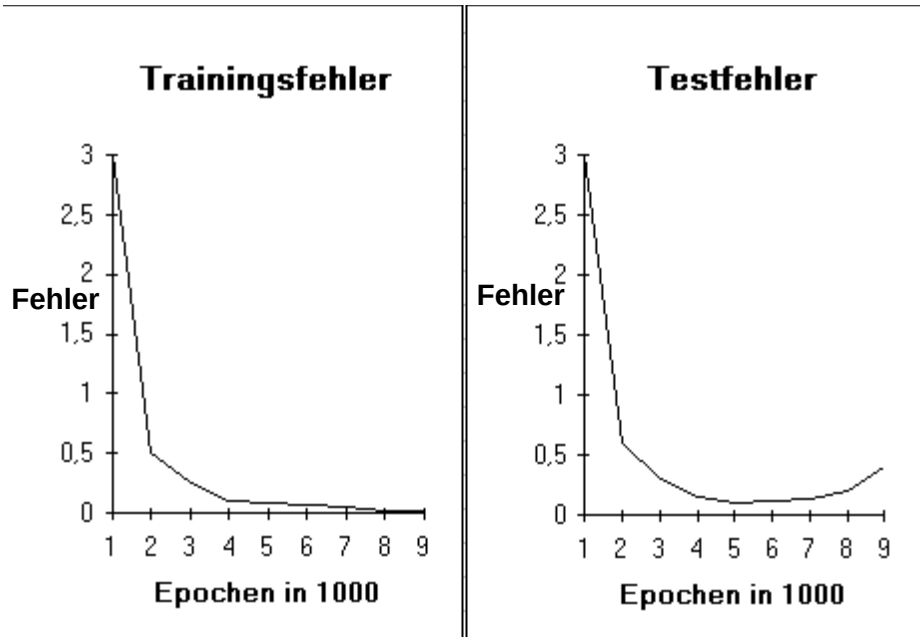


Abbildung 3-7: Fehler in Trainingsmenge und Testmenge in Abhängigkeit von der Anzahl der Trainings-Epochen

Der Fehler in der Trainingsmenge konvergiert mit steigender Epochenzahl. Der Testfehler sinkt nur bis zu einem bestimmten Zeitpunkt und steigt dann wieder an. Durch längeres Training sinkt die Generalisierungsfähigkeit. Diesen Effekt nennt man *overlearning*. Eine plausible Erklärung dieses Phänomens ist, dass das Netz sich bei längerer Lernzeit immer mehr auf die Eigenheiten in den Trainingsfällen spezialisiert. Das Training wird daher meist bei einem Minimum in der Testmenge beendet.

- Anzahl der Neuronen
  Die Daten bestimmen die Zahl der Units in Input- und Output-Schicht. Die Zahl der versteckten Neuronen dagegen wird durch Experimentieren optimiert. Sie hängt von der Komplexität der anzunähernden Funktion ab. In der Regel wird sie zwischen der Zahl der Input- und Output-Neuronen liegen.

- Die Anzahl der Verbindungen
  Die Zahl der Verbindungen hängt natürlich stark von der Zahl der Neuronen ab. Meistens sind Feed-Forward-Netze nur zwischen den Schichten vernetzt. Es besteht aber die Möglichkeit, Verbindungen einzuführen, die Schichten überspringen (*shortcut connections*) und

die z.B. vom Input direkt zum Output laufen und eine versteckte Schicht auslassen. Andererseits kann man auch geringere Grade der Vernetzung festlegen, indem man Maße für die maximale Zahl von Verbindungen angibt, die von einer Unit ausgehen oder bei dieser ankommen (*fan-in, fan-out*). Allein zufälliges Eliminieren von Gewichten führte in Experimenten von le Cun 1989 zu Verbesserungen der Performanz. Die Einführung eines *weight decay*, der bei jedem Schritt alle Verbindungen leicht schwächt, unterstützt die Suche nach unwichtigen Verbindungen, die dann aus dem Netz entfernt werden (cf. z.B. Moody 1992).

3.5.4.3 Weitergehende Parameter

Zahlreiche Verbesserungen und Varianten für den Backpropagation-Algorithmus versuchen, diesen Schwächen zu begegnen. Die meisten versuchen, die Konvergenz durch mathematische Verfahren zu beschleunigen. Verschiedene empirische Studien zeigen, dass die Auswirkungen der einzelnen Parameter sehr stark vom Problem abhängen (cf. z.B. Cherkassky/Vassilas 1989, Jervis/Fitzgerald 1993, Schiffmann et al. 1993, Cichocki/Unbehauen 1993).

- Einsatz alternativer Aktivierungsfunktionen
  Die Aktivierungsfunktion steuert die Aktivierung eines Neurons in Abhängigkeit von der ankommenden Aktivierung. Eine Veränderung kann die gesamte Dynamik eines Netzes verändern.
- Änderungen an den Variablen der Aktivierungsfunktion
  Viele Aktivierungsfunktionen benutzen Variablen, die ihren Verlauf beeinflussen. Veränderungen betreffen wiederum das gesamte Netzwerk.
- Initialisierung der Gewichte
  Alle Gewichte eines Netzes werden vor dem Training zufällig initialisiert. Das Training beginnt damit an einem zufälligen Punkt der Fehlerfunktion. Der wichtigste Parameter ist dabei das Intervall für die Initialisierungsgewichte. In der Regel wird das Intervall [-1; 1] benutzt. Meist wird ein Netz mehrfach initialisiert und mit den gleichen Parametern trainiert. Das beste Ergebnis wird dann verwendet.
- Multitask-Learning
  Bei Multitask-Learning wird einem Backpropagation-Netzwerk zusätzliche Information im Output gegeben. Das Netz lernt neben der erwünschten Abbildung noch weitere Abbildungen, die für den Anwendungsfall aber

nicht primär interessant ist. Damit erhält der Lernalgorithmus mehr Fehlerinformation, um die Verbindungen einzustellen (cf. Caruana 1995/97). Dies widerspricht der Intuition zusätzliche Information als Input einzuleiten.

- In mehreren empirischen Untersuchungen konnte dadurch die Qualität für die erwünschte Abbildung verbessert werden. So lernte z.B. ein Netz von Bartlmae 1998 die Kreditwürdigkeit von Länder zu bestimmen. Das Lernen zusätzlicher Informationen über diese Länder im gleichen Netz erhöhte die Qualität für die eigentliche Aufgabe.

- Varianten des Lernverfahrens

  - *Learning Rate Adaption*
    Die Erhöhung der Lernrate ist die einfachste Möglichkeit, Backpropagation zu beschleunigen, sie führt aber häufig zu unerwünschten Nebenwirkungen. Die Lernrate wird selten größer 0,3 gewählt. Es könnte zum einen zu Oszillationen (cf. Cichocki/Unbehauen 1993:146) aufgrund konkurrierender Muster kommen. Diese Probleme haben zu Überlegungen geführt, variable Lernraten zuzulassen, die zu Beginn des Lernens hoch sind und gegen Ende hin abnehmen (cf. Cichocki/Unbehauen 1993:144).

  - Stochastische Lernverfahren
    Mathematisch weniger gut zu erfassen sind Methoden, die stochastische Elemente in den Backpropagation-Prozess einbringen. Dies kann zum einen bei der Aktivierung und zum andern beim Lernen geschehen. Am einfachsten wird dies durch zufallsgesteuerte Elemente in der Aktivierungs- und Lernfunktion realisiert. Durch eine derartige *noise injection* kann das Lernverfahren eventuell aus einem lokalen Minimum entkommen (cf. Dorffner 1991:271). Dies ähnelt dem Algorithmus der Boltzmann-Maschine (cf. Abschnitt 4.2).

- Momentum
  Die Einführung eines zusätzlichen Terms versucht, Backpropagation zu beschleunigen. Das sehr häufig eingesetzten Backpropagation mit Momentum addiert bei jeder Gewichtsveränderung ein Bruchteil der letzten Gewichtsveränderung hinzu. Dadurch soll die aktuelle Lerngeschwindigkeit und –richtung beibehalten werden. Durch den Momentum-Term

würden Oszillationen verhindert und flache Strecken in der Fehlerfunktion könnten schneller durchschritten werden (cf. Zell et al. 1995:115).

- Automatische Anpassung der Architektur
  Die optimale Architektur für die Lösung eines Problems mit einem Backpropagation-Netz lässt sich nur experimentell finden. Somit liegt es nahe, dies zu automatisieren. Mehrere Lernverfahren verändern die Architektur während des Lernens und versuchen, sie dem Problem optimal anzupassen.
  - *Cascade Correlation* ist ein Meta-Algorithmus, bei dem nur Input- und Output-Schicht definiert werden müssen. Der Algorithmus trainiert Netze bis zu einem bestimmten Fehler, fügt selbst versteckte Neuronen hinzu und trainiert das Netz erneut. Wenn möglich löscht das Verfahren Neuronen auch wieder (cf. Zell 1994:161ff.).

  - Pruning bezeichnet Techniken, die Verbindungen aus einem neuronalen Netzwerk löschen (cf. Zell 1994:319ff.). Meist wird Pruning auf bereits trainierte Netze angewandt. Bei allen Verfahren wird über eine mathematische Analyse der Einfluss einer Verbindung abgeschätzt und Verbindungen, die unter einen bestimmten Schwellenwert liegen, werden gelöscht. Auch Pruning erfordert also heuristische Parametersetzungen (cf. Zell et al. 1995:333f.).

3.6 Simulationssoftware

Künstliche neuronale Netze werden meist nicht als Hardware mit physikalisch vorhandenen Neuronen implementiert, sondern durch Software simuliert. Anstelle der parallelen Verarbeitung tritt eine serielle Implementierung, bei der ein zentraler Prozessor die Berechnungen für jedes Neuron nacheinander ausführt. Dies beeinflusst lediglich die Geschwindigkeit des Netzes, während die Vorteile wie Robustheit, Fehlertoleranz und bei Backpropagation-Netzwerken subsymbolische Verarbeitung von Information erhalten bleiben. Einen Überblick über vorhandene Software bietet Searle 2000. Für die in Kapitel 7 beschriebenen Experimente wurden die im folgenden beschriebenen zwei Simulatoren benutzt.

3.6.1 Stuttgarter Neuronaler Netzwerk Simulator

Der Stuttgarter Neuronale Netzwerk Simulator (SNNS) läuft als X-Windows-Anwendung unter UNIX (Z.B. SUN-OS und LINUX). Es verfügt über eine graphische Benutzungsoberfläche. Der Funktionalitätsumfang ist insbesondere in Hinblick auf die implementierten Netzwerkmodelle enorm. Die oben besprochenen Modelle sind alle mit zahlreichen Varianten enthalten.

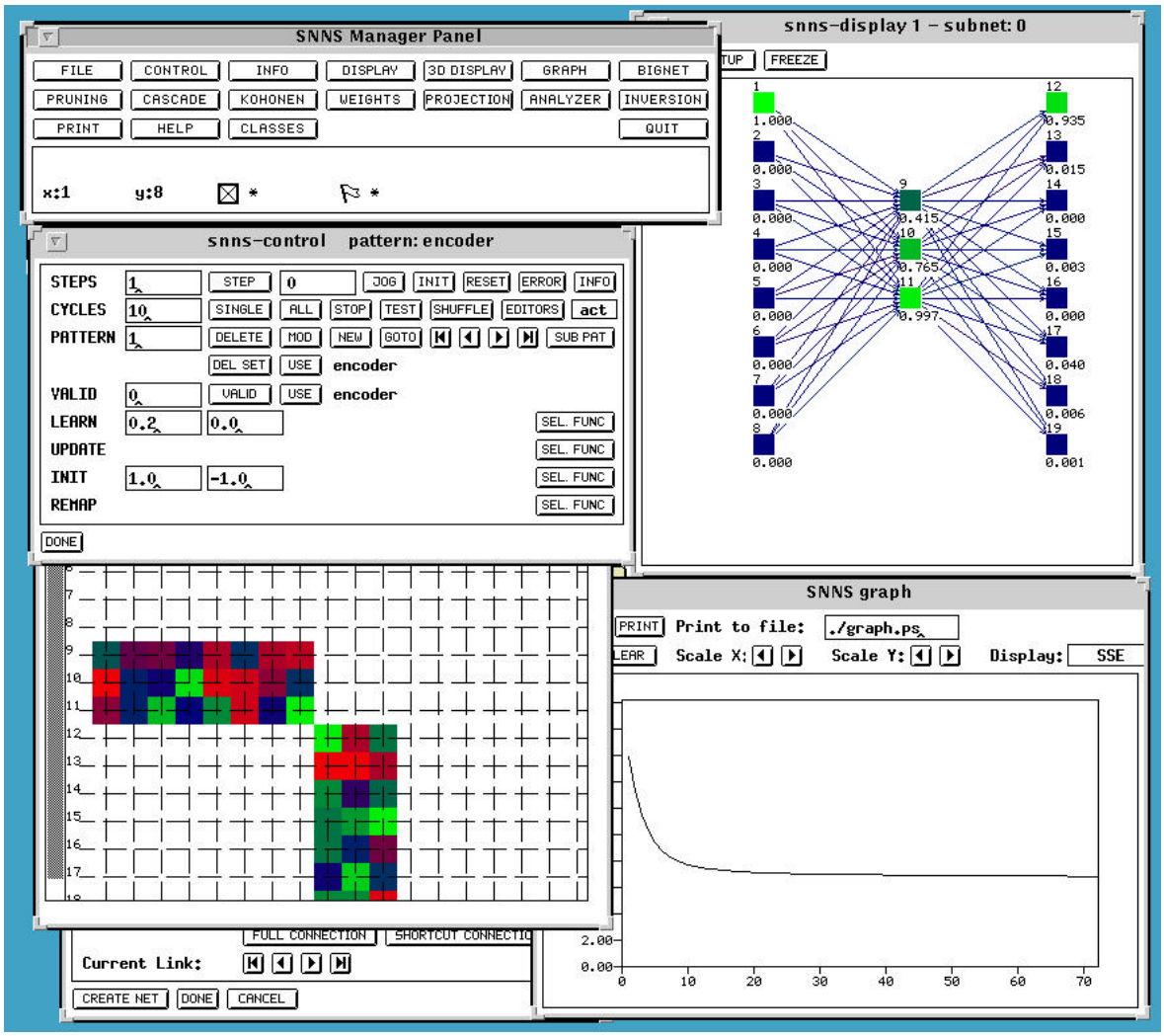


Abbildung 3-8: Die Benutzungsoberfläche von SNNS

SNNS erlaubt mehrere Möglichkeiten der Visualisierung. Das Netzwerk selbst erscheint in 2D oder 3D, wobei Farben die Aktivierung der Knoten und die Stärke der Verbindungen kodieren. Abbildung 3-8 zeigt außer dem Hauptfenster oben links ein Steuerfenster für die Auswahl der Aktivierungs- und Trainingsfunktion und die Steuerung des Lernprozesses. Weiterhin zeigt Abbildung 3-8 eine Visualisierung des Netzwerks, eine Visualisierung der

Verbindungsstärken und den Verlauf der Fehlerfunktion in einem Koordinatensystem.

Neben der graphischen Schnittstelle gibt es noch die Möglichkeit, interaktiv im Batch-Modus zu arbeiten und das C-Kernel-Interface zu benutzen. Trainierte Netze können zudem als C-Code exportiert und in anderen Programmen eingesetzt werden. SNNS entstand ursprünglich an der Universität Stuttgart und steht für Forschungszwecke kostenlos zur Verfügung[1]. Ein umfangreiches Handbuch erläutert die zahlreichen Netzwerktypen, ihre Parameter und die Bedienung der Software (cf. Zell et al. 1995).

3.6.2 DataEngine

DataEngine[2] ist ein kommerzielles Produkt, das unter Windows läuft (cf. z.B. Zimmermann 1995). Es erlaubt die Realisierung von Backpropagation-Netzwerken, von Kohonen- und von Fuzzy Kohonen-Netzen.

Die besondere Stärke von DataEngine liegt in der Kombination von Fuzzy Logik und neuronalen Netzen in einem Programm. DataEngine erlaubt beim Backpropagation Netz wesentlich weniger Varianten und Möglichkeiten zur Modifikation als SNNS. Insbesondere kann nicht auf einzelne Verbindungen zugegriffen werden, um z.B. manuell Gewichte zu verändern oder Verbindungen zu löschen.

Durch das Bilden von Blöcken können Verarbeitungsprozesse realisiert werden, an denen beide Techniken beteiligt sind. Eine weitere Stärke liegt in den Bearbeitungsmöglichkeiten für Daten, was meist den größten Teil der Arbeit mit neuronalen Netzen ausmacht. Ein mächtiger Spreadsheet-Editor erlaubt typische Vorverarbeitungsschritte und die Anwendung statistischer Funktionen.

---

[1] http://www-ra.informatik.uni-tuebingen.de/SNNS/

[2] http://www.mitgmbh.de/mit/sp/index.htm

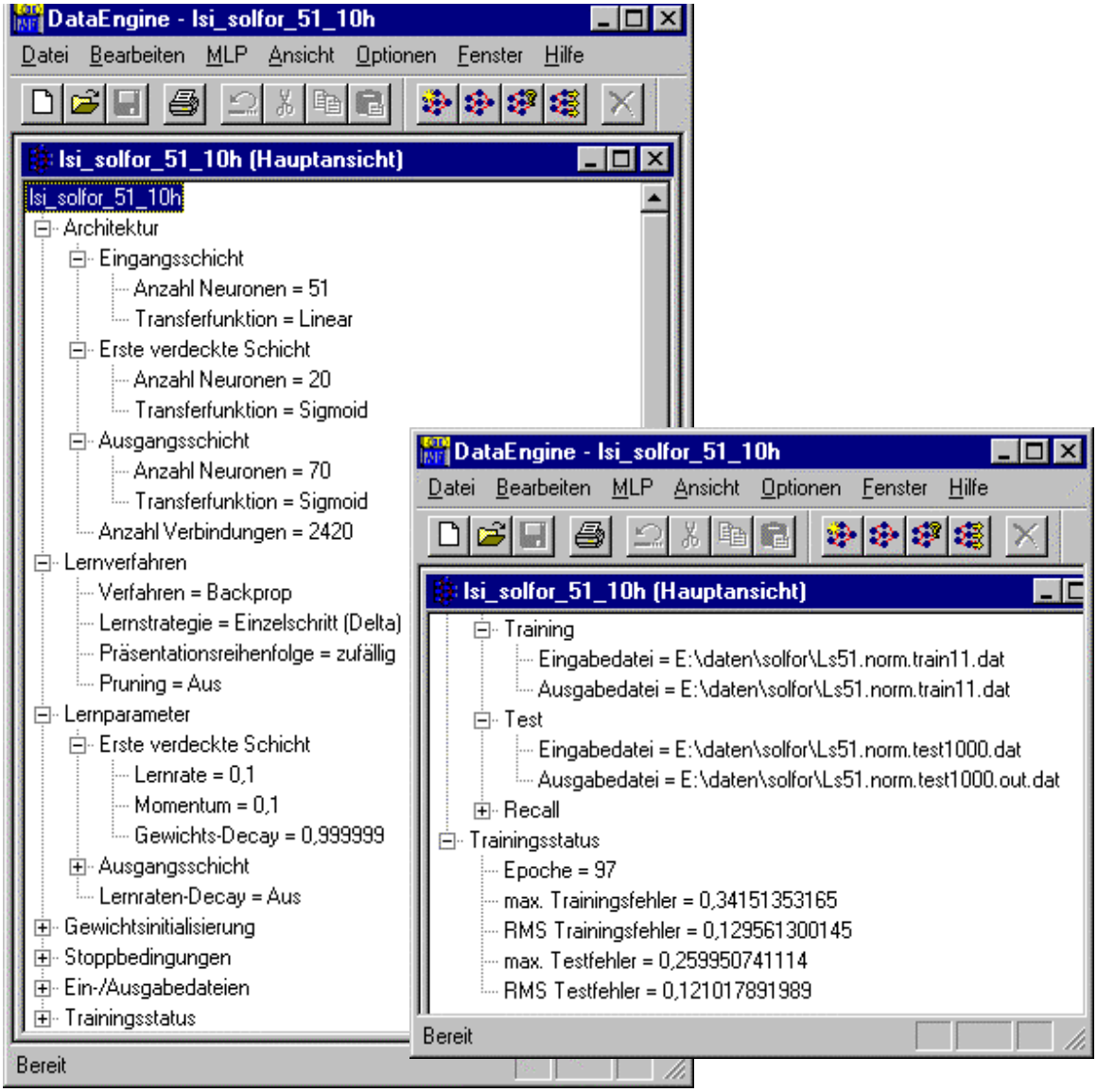


Abbildung 3-9: Die Benutzungsoberfläche von DataEngine

3.7 Fazit: Grundlagen neuronaler Netze

Neuronale Netze sind eine relativ neue, aber bereits erfolgreich eingesetzte Technik zur Verarbeitung von Informationen, die zum Paradigma des Soft-Computing gehört. Neuronale Netze ermöglichen vage und tolerante Informationsverarbeitung und sind lernfähig. Verschiedene Modelle stehen für unterschiedliche Anwendungen zur Auswahl.

Die in Abschnitt 2.3 diskutierten Schwächen bestehender Information Retrieval Systeme erfordern u.a. diese Eigenschaften. Lernen erhöht die Adaptivität von Retrieval Systemen und die Heterogenität erfordert tolerante Verarbei-

tungsverfahren. Beim neuronalen Backpropagation-Algorithmus steht der Benutzer im Mittelpunkt, wenn er die Trainingsbeispiele liefert. Diese Benutzerorientierung wird im Information Retrieval gefordert. Deshalb liegt es nahe, neuronale Netze im Information Retrieval einzusetzen und zu testen.

4 Neuronale Netze im Information Retrieval

Das Potenzial neuronaler Netze als Modell für Information Retrieval schätzen viele Forscher als sehr hoch ein, was die folgenden Zitate verdeutlichen:

> „... connectionist models provide a generic computational methodology of potential applicability to most aspects in information retrieval. To the extent that a problem can be formulated as a connectionist model, enormous computational power can be brought to bear in solving that problem through massively parallel processing." (Doszkocs et al. 1990:228)

> „[It]... makes one wonder why the research toward neural nets in information retrieval still is restricted to such a small school" (Scholtes 1992:638)

> „The learning property of backpropagation networks and the parallel search property of the Hopfield network provide effective means for identifying relevant information items in databases." (Chen 1995:201)

Welche Rolle spielen neuronale Netze aber tatsächlich im Information Retrieval? In kommerziellen Systemen sind sie kaum zu finden. Der folgende state-of-the-art Bericht gibt einen ausführlichen Überblick über die vorhandenen Systeme und ihre Leistungsfähigkeit.

4.1 Historischer Überblick

Neuronale Netze erlebten in den 80er Jahren eine Renaissance, wobei das Erscheinen des Sammelbands Rumelhart/McClelland 1986 als markantes Ereignis gilt. Dieser Einschnitt etablierte vor allem den Backpropagation-Ansatz und führte zu zahlreichen Anwendungen.

Zu dieser Zeit wurden bereits vernetzte Systeme im Information Retrieval diskutiert, jedoch handelte es sich dabei fast durchwegs um semantische Netze, die Verbindungen symbolisch definieren, während die Verbindungen

zwischen künstlichen Neuronen rein numerisch sind. Die semantischen Spreading-Activation-Netzwerke werden in Abschnitt 4.3.2.7 besprochen.

Bereits wenige Jahre später erschienen bei der ACM SIGIR-Konferenz (Association of Computing Machinery, Special Interest Group Information Retrieval) zwei theoretisch fundierte Ansätze, in denen die Potenziale der Spreading-Activation-Netzwerke insbesondere im Bereich Lernen bereits deutlich wurden und die sich im Laufe der Zeit weiterentwickelten (cf. Kwok 1989, Belew 1989). Einen guten Überblick über die frühe Entwicklung dieser Spreading-Activation-Netzwerke für Information Retrieval bieten Doszkocs et al. 1991. Die beiden Ansätze entwickelten sich in verschiedene Richtungen weiter. Belew bemühte sich bei der weiteren Arbeit um eine Integration semantischer und neuronaler Netze (cf. Abschnitt 4.3.2.6). Der Ansatz von Kwok 1989 ging in dem System PIRCS auf (cf. Abschnitt 4.3.2.1) und bewährte sich im Rahmen der TREC Initiative an Massendaten (cf. Abschnitt 4.8). In den 90er Jahren folgten weitere Systeme, wobei neben PIRCS noch Mercure (cf. Abschnitt 4.3.2.4) den Schritt zur Anwendung in TREC schaffte. Diese Erfolge der Spreading-Activation-Netzwerke deuten auf eine gewisse Konsolidierung hin, die sich jedoch nicht in der kommerziellen Entwicklung niedergeschlagen hat. Nach wie vor bekennt sich keines der bekannteren und auf dem Markt erhältlichen IR-Systeme zu neuronalen Netzen. Lediglich die Firma Infoseek berichtet, in ihrer Suchmaschine Technologie der Firma HNC eingebunden zu haben[1]. HNC hat mit MatchPlus auch an TREC teilgenommen (cf. Abschnitt 4.8).

4.2 Retrieval mit Assoziativspeichern

Der Begriff Retrieval ist in der Neuroinformatik bereits belegt, und wird für das Retrieval von Mustern aus Assoziativspeichern benutzt. Vor allem Hopfield-Netzwerke (cf. Abschnitt 3.5.3) wirken als assoziative Speicher (content-addressable memory). Sie speichern Muster durch das Einstellen ihrer Verbindungen. Ein zentraler Begriff bei Assoziativspeichern ist die Energie im Netzwerk, die ein Maß für die Summe aller Aktivierungen darstellt. Ein gespeichertes Muster stellt dann ein Energie-Minimum im Netz dar. Nach einem Input konvergiert das Netz zu einem solchen Energie-Minimum. Hopfield-Netze eignen sich somit vor allem für die Zuordnung von unvollständigen oder beschädigten Mustern zu einem der gespeicherten Muster. Retrieval bedeutet hier also Finden eines möglichst ähnlichen Musters. Dies

[1] http://info.infoseek.com/doc/PressReleases/aptex.html

gilt für die am weitesten verbreitete Form der Assoziativspeicher, die *auto-associative memories*, bei denen die Neuronen zugleich Input und Output bilden. Die hetero-assoziativen Systeme dagegen nutzen für Input und Output verschiedene Neuronen und realisieren so eine Abbildung zwischen beliebigen Räumen verschiedener Dimensionalität. Damit ähneln sie den als Spreading-Activation-Netzen im IR sehr populären Systemen, bei denen Input und Output in eigenen Schichten angeordnet sind und die in Abschnitt 4.3 diskutiert werden. Die Übergänge zwischen diesen Netzwerktypen sind fließend.

In ihrer auto-assoziativen Form eignen sich Hopfield-Netzwerke kaum für den IR-Prozess, auch wenn die Analogie zwischen Retrieval aus Assoziativspeichern und Retrieval im Information Retrieval zunächst plausibel erscheint. Als Muster speichert das Netz die Dokumente und die Anfrage bildet das unvollständige Muster, das eingegeben wird. Jedoch konvergiert das Netz dann nur zu einem einzelnen Muster und liefert somit nur ein Dokument. In einem Information Retrieval Prozess will der Benutzer aber in der Regel eine größere Menge von Dokumenten. Im Prinzip ist ein Hopfield-Netzwerk ein System zu Klassifizierung und eignet sich zumindest in Reinform nicht für Information Retrieval.

4.2.1 Hopfield-Netzwerke

Personnaz et al. 1986 schlagen ein Hopfield-Netz für Information Retrieval vor. Sie zeigen, dass darin keine Zyklen auftreten und betonen, dass das System orthogonale Muster besonders leicht findet. In einem kleinen Beispiel speichern Personnaz et al. 1986 die Titel wissenschaftlicher Zeitschriften als Folgen von Buchstaben. Auch bei Input von unvollständigen Folgen von Buchstaben konvergiert das Netz schnell zu den gespeicherten Titeln.

Chen 1995 beschreibt ein Hopfield-Netz, das nicht in Schichten aufgeteilt ist, und in dem jedes Neuron einen Thesaurusterm repräsentiert. Die Terme sind untereinander verbunden. Ihre Gewichte werden mittels eines Thesaurus initialisiert, wobei die Verbindungsstärken die Beziehung zwischen den Termen wiedergeben. Dabei kann es sich um einen automatisch, auf der Basis von statistischen Informationen erstellten und/oder um einen intellektuell erstellten Thesaurus handeln. Für einen intellektuell erstellten Thesaurus werden heuristisch numerische Werte festgelegt, die Beziehungen wie Synonym oder *allgemeinerer Begriff* entsprechen. Ein Hopfield-Netzwerk verstärkt häufig die genutzten Verbindungen und implementiert so unüberwachtes Lernen. Dies ist bei dem Netz von Chen 1995 nicht der Fall. Lernen findet hier nicht statt.

Ein Benutzer wählt einen oder mehrere Terme für eine Suche. Die Ausbreitung der Aktivierung erreicht mit den ursprünglichen Begriffen assoziierte Terme, die dem Benutzer dann vorgeschlagen werden. Diese Terme kann der Benutzer bewerten und je nach Relevanz werden sie wieder aktiviert. Dann läuft die Aktivierungsausbreitung weiter. Das Netz bildet also nicht den eigentlichen Retrieval-Prozess ab, sondern leistet eine assoziative Term-Expansion, die vor oder während der Interaktion mit einem IR System abläuft. Das System kann als ein Spezialfall des Spreading-Activation-Ansatzes (cf. Abschnitt 4.3) betrachtet werden, bei dem nur eine Term-Schicht vorliegt, die durch inner-layer Verbindungen verknüpft ist.

Interessant an dem Ansatz von Chen 1995 ist die Integration von mehreren verschiedenen Thesauri. Neben einem aus 3000 Dokumenten extrahierten Thesaurus wurden jeweils Teile des *ACM Computing Review Classification System* und der *Library of Congress Subject Headings* benutzt. Insgesamt besitzt das Netz 14.000 Terme und 80.000 Verbindungen. Zwischen den einzelnen Thesauri bestehen zunächst keine Beziehungen. Um diese zu definieren, wurden intellektuell Cluster von zusammengehörenden Termen erstellt. Benutzer stellten dazu für sie interessante Begriffe zusammen. Diese Cluster wurden dann als eigene Neuronen implementiert. Cluster-Neuronen, die Terme aus mehreren Thesauri beinhalten, dienen als indirekte Verbindung zwischen den Thesauri. Dabei erscheint fragwürdig, ob die von einzelnen Benutzern subjektiv erstellten Listen zu Verbindungen führen, die für alle Anwender effizient sind. Weiterhin ist unklar, wie viele solche Beziehungen nötig sind und wie viele im vorliegenden Netz integriert waren. Dies wurde dadurch umgangen, dass in jeder Anfrage Terme aus allen drei Thesauri vorhanden waren. Solche künstlichen Anfragen kommen in einer realen Umgebung natürlich nicht vor. Interessant an dem Ansatz ist, dass er widersprüchliche Aussagen über Beziehungen integrieren kann.

Die Bewertung des Ansatzes prüfte nur die Qualität der Term-Erweiterung. Der Effekt auf den Retrieval-Prozess lässt sich so nicht messen. Dazu müssten z.B. Anfragen mit und ohne Term-Erweiterung evaluiert werden.

Chen et al. 1995 erweitern den Hopfield-Ansatz auf eine multilinguale Umgebung (cf. Abschnitt 5.3.3).

Bordogna et al. 1996 erweitern ein assoziatives Hopfield-Netzwerk auf reelle Aktivierungswerte. Zusätzlich lernt das Netz aus Relevanz-Feedback.

Chung et al. 1998 realisieren ein Hopfield-Netzwerk, das den Indexierungsprozess in vernetzten Umgebungen unterstützt. Sie gehen davon aus, dass viele Autoren von Dokumenten v.a. im Internet nicht für intellektuelle Indexierung geschult sind. Ein System, das den Volltext eines Dokuments analysiert, soll ihnen Terme für die Indexierung vorschlagen. Auch bei der

Auswahl von vorgeschlagenen Termen ist grundsätzlich fragwürdig, ob Laien eine Qualität der Indexierung erreichen, die sich positiv auf den Retrievalprozess auswirkt. In der Forschung ist ohnehin umstritten, welche Art der Indexierung bessere Ergebnisse bringt (cf. Krause 1996a:81f.). Fragwürdig ist auch, ob in einer betont nicht regulierten Umgebung hinreichend genügend Autoren von solchen Werkzeugen Gebrauch machen. Sinnvoller ist ein Mechanismus wie das Schalenmodell, das verschiedene Schalen mit unterschiedlicher Qualität von Dokumenten und Indexierung zulässt und sie verwaltet (cf. Abschnitt 5.1.2).

Das System von Chung et al. 1998 indexiert eine Kollektion und analysiert die Kookkurrenzen für alle Termpaare. Daraus wird unabhängig von der interaktiven Indexierungskomponente mit einer Kohonen-Karte eine zwei- und dreidimensionale Visualisierung erstellt. Vergleichbare Systeme diskutiert Abschnitt 4.4. Die Kookkurrenz-Werte bilden die Gewichtungen der Verbindungen im Hopfield-Netzwerk. Chung et al. 1998 verwenden somit nicht das Hopfield-Lernverfahren (cf. Abschnitt 3.5.3), sondern setzen die Werte der Verbindungen direkt aus der Indexierung ein, um Muster im Netz zu speichern. Das Retrieval oder die Suche nach einem ähnlichen Muster zum Input verläuft dann wieder wie in einem Hopfield-Netzwerk. Das Muster wird als Aktivierungsmuster auf die betroffenen Neuronen übertragen. Die Spreading-Activation verläuft nach einer der üblichen Aktivierungsfunktionen (cf. Abschnitt 3.4.1). Nachdem ein stabiler Zustand erreicht ist, bei dem mehrere Schritte wenig an den Aktivierungzuständen der einzelnen Neuronen ändern, endet der Prozess. Die am stärksten aktivierten Terme bilden den Output und werden dem Benutzer, in diesem Fall dem Autor des Dokuments als Index-Terme vorgeschlagen.

Chung et al. 1998 berichten von Experimenten mit einer Menge von 290 Dokumenten. In einem Test mit Studenten als Benutzer evaluieren sie die Übereinstimmung der ausgewählten Terme mit den von professionellen Indexierern vergebenen Schlagwörtern. Die Übereinstimmung messen sie mit den Maßen Precision und Recall, die im Information Retrieval üblich sind. Hier liefern sie aber keine Aussage über das Retrieval. Die Übereinstimmung von Index-Termen zwischen verschiedenen Indexierungsverfahren sagt noch nichts über die zu erwartende Qualität des Retrievals. Ähnliche Verfahren, die als Vorschlagsmodus Terme aus kontrolliertem Vokabular automatisch bestimmen, stellt Abschnitt 5.3.2.1 im Kontext der Heterogenitätsbehandlung vor.

Die Problematik der Bewertung greift auch Abschnitt 7.2.2 auf. Für die Übertragung des Verfahrens auf große Kollektionen erwarten die Autoren Performanzprobleme, so dass sie an einer Beschleunigung durch Parallelisie-

rung arbeiten. Gegenüber der algorithmischen Optimierung sollte die Integration des Verfahrens in den Information Retrieval Prozess allerdings höhere Priorität besitzen.

4.2.2 Boltzmann-Maschine

Die Boltzmann-Maschine (cf. Abschnitt 3.5.3) ist ein Hopfield-Netzwerk mit spezifischer Aktivierungs- und Lernregel. Die Aktivierung verläuft als *simulated annealing* mit einer probabilistischen Komponente. Dahinter steht die Metapher eines auskühlenden Metalls, das seine Energie schrittweise verringert.

Brachman/McGuiness 1988 stellen ein IR-System auf der Basis einer Boltzmann-Maschine vor. Ihr Ausgangspunkt ist Wissensrepräsentation in der Künstlichen Intelligenz in der Form von Frames. Information Retrieval wird als logische Deduktion interpretiert, in der die Anfrage Ausgangspunkt einer Reihe von Folgerungen darstellt. Diese Sichtweise findet sich auch bei anderen Autoren (cf. z.B. van Rijsbergen 1986, Fuhr 1995, Müller/Thiel 1994). Der Inferenz-Prozess bei Brachman/McGuiness 1988 wird insbesondere von Generalisierungs- und Spezialisierungsrelationen aus einer strukturierten Frames-Repräsentation des Gegenstandsbereichs geleitet. Brachman/McGuiness 1988 übertragen die Repräsentation von Frames auf eine Boltzmann-Maschine um auch nicht exakte Übereinstimmungen (*partial match*) besser ableiten zu können. Ihr System CRUCS (Conceptual Retrieval using Connectionist Style) bildet die Eigenschaften aller beteiligten Klassen und Objekte auf Neuronen ab. Diese *Micro-Features* dürfen jedoch nicht wie Brachman/McGuiness 1988 mit sub-symbolischen Repräsentationen in einem Backpropagation-Netzwerk verwechselt werden, da sie eine symbolische Entsprechung besitzen. Brachman/McGuiness 1988 beschreiben, wie ein Objekt, das in einer komplexen hierarchischen Struktur steht, auf ein neuronales Aktivierungsmuster übertragen wird. In diesem Prozess werden alle Neuronen, die in dem Pfad von dem zu aktivierenden Muster bis zum höchsten Konzept stehen, auf höchste Aktivierung gesetzt. Wie in Abbildung 4-1 deutlich wird, erhöht diese Art der Repräsentation die Verteiltheit der Muster. Ebenso erzeugt sie eine Ähnlichkeit zwischen benachbarten Neuronen. So ist der Aktivierungsvektor des Musters *Innendienst*, der in Abbildung 4-1 aktiviert ist, sehr ähnlich zu dem Aktivierungsvektor von *Angestellte*. In vielen Anwendungsfällen ist eine Erhaltung der Ähnlichkeit über mehrere Stufen hinweg nicht erwünscht, wie etwa bei den Produktnomenklaturen in ELVIRA (cf. Abschnitt 2.2.3.2, cf. Mandl 1998b). Um diese Anforderung mit den

Vorteilen der obigen Repräsentation zu verbinden, könnten höhere Ebenen schwäche Aktivierung erhalten.

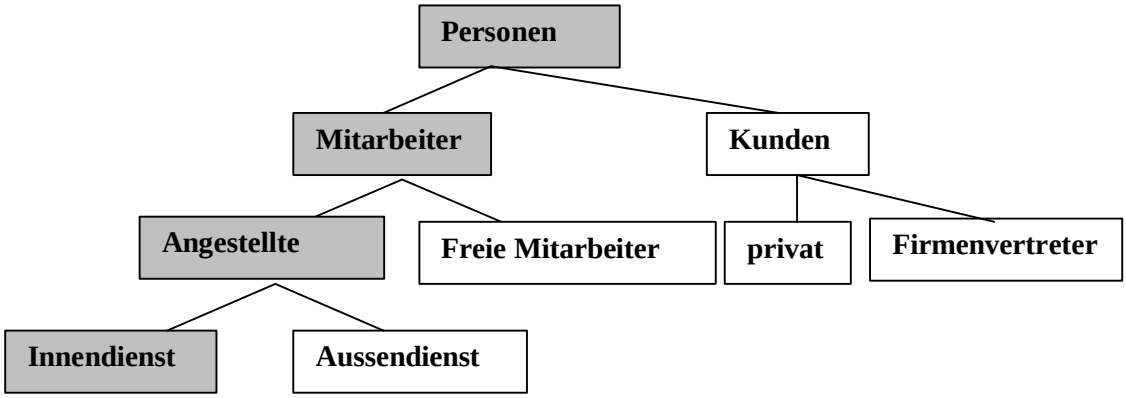


Abbildung 4-1: Beispielhafte Hierarchie. Grau hinterlegte Muster sind aktiviert.

Das Netzwerk von Brachman/McGuiness 1988 ist in Schichten von Neuronen aufgeteilt. Eine Schicht vereint die als *individuals* bezeichneten Dokumente. Die Eigenschaften der Objekte bzw. Dokumente bilden eine weitere Schicht, die von Brachman/McGuiness 1988 als *Micro-Features* eingeführt wird. Dahinter stehen Konzepte wie *objekt-orientiert* oder *LISP-ähnlich*, während die Objekte konkrete Programmiersprachen wie LOOPS repräsentieren.

Die ursprünglichen Frames werden intellektuell erstellt. Brachman/McGuiness 1988 beschreiben eine Anwendung für die Beziehungen zwischen verschiedenen Programmiersprachen. Für größere Anwendungen wäre die Formalisierung umfassenden Wissens erforderlich, was sehr aufwendig ist und u.a. große Konsistenzprobleme mit sich bringt. Für ein IR-System, das nicht auf ein kleines Anwendungsgebiet restringiert ist, kann dieses Verfahren kaum angewendet werden. Der Ansatz von Brachman/McGuiness 1988 bedient sich sowohl neuronaler Netze als auch symbolischer Verarbeitungsmechanismen und gehört so zu den hybriden Systemen. Auch einige Spreading-Activation-Netzwerke integrieren Elemente symbolischer Wissensverarbeitung (cf. Abschnitt 4.3.2.7).

Vor dem Retrieval-Prozess speichern Brachman/McGuiness 1988 die gewünschten Muster in der Boltzmann-Maschine. Dann werden abhängig von Anfrage einige Neuronen aktiviert und konstant auf dieser Aktivierung gehalten (*clamped*). Dabei ist unklar, ob alle *Micro-Features* als Input dienen oder nur die positiv aktivierten. Zählen auch die nicht aktivierten und damit auf Null gesetzten Neuronen zum Input-Vektor, müssen auch sie konstant auf

diesem Wert gehalten werden. Damit bliebe in der gesamten Schicht wenig Spielraum für die Aktivierungsausbreitung und im gesamten Netz sind nur relativ wenig Neuronen frei modifizierbar.

Beim Anwendungsbeispiel Programmiersprachen und ihren Eigenschaften bestehen die Anfragen aus sehr konkreten Faktenabfragen, die einen logischen Ableitungsprozess erfordern, wie ihn z.B. objekt-orientierten Datenbanken gut lösen. Der Vorteil von CRUCS soll insbesondere bei nicht exakten Abgleichen liegen. Der *simulated annealing* Prozess läuft dabei mehrfach ab und ein Neuron bildet das Ergebnis. Dieses Neuron wird im nächsten Durchlauf konstant auf Aktivierung Null festgehalten und damit unterdrückt. Das Netz muss nun einen anderen stabilen Zustand finden und liefert ein anderes Neuron als zusätzliches Ergebnis. Brachman/McGuiness 1988 zeigen an einem Beispiel, dass es so zu *partial match* Ergebnissen kommt.

Der Ansatz von Brachman/McGuiness 1988 bietet einen guten Ausgangspunkt für die Integration von vorhandenem symbolischen Wissen in ein neuronales Netz. Er steht jedoch dem symbolischen Ansatz näher als dem konnektionistischen Paradigma. Aufgrund der aufwendigen, intellektuellen Vorarbeit kommt dieses Vorgehen für ein generelles IR-System kaum in Frage. Deshalb baut das System CRUCS auch nur auf einer sehr kleinen Datenbasis auf. Ein weiterer großer Nachteil besteht darin, dass das System nicht aufgrund von Nutzerurteilen lernt.

Die Verteilung von Eigenschaften und Objekten auf verschiedenen Schichten ähnelt der Funktionsweise von Spreading-Activation-Netzwerken, die Abschnitt 4.3 diskutiert.

4.2.3 Hetero-assoziative Systeme

Die bisher vorgestellten Systeme sind auto-assoziative Speicher, bei denen Input- und Output-Vektor im gleichen Merkmals-Raum liegen und die gleichen Objekte oder Eigenschaften repräsentieren. Das folgende System ist hetero-assoziativ und schlägt bereits die Brücke zum nächsten Abschnitt. Wie die Spreading-Activation-Modelle implementiert es eine Abbildung zwischen unterschiedlichen Räumen.

Den Ansatz von Bentz et al. 1989 entwickeln u.a. Heitland 1994 und Hagström 1996 zu SpaCAM (Sparsely Coded Associative Memory) weiter. Ausgangspunkt ist die Palm-Matrix, ein hetero-assoziativer Speicher, der Paare von Input- und Output-Mustern lernt.

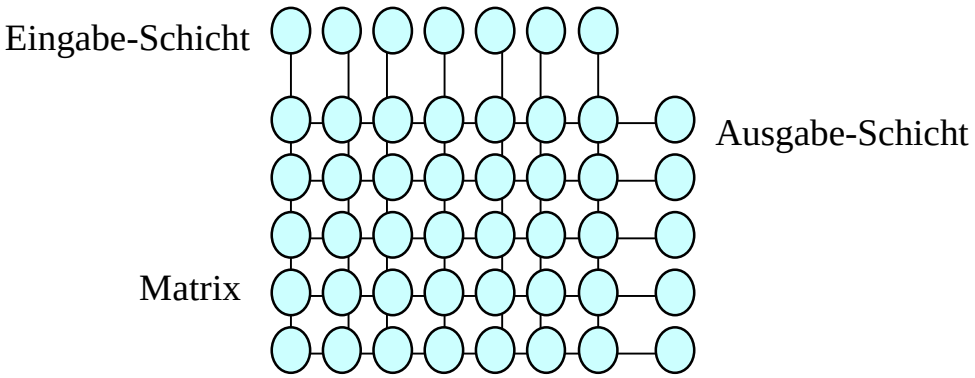


Abbildung 4-2: Schematische Darstellung einer Palm-Matrix (Hagström 1996:18)

Hagström 1996 zeigt, dass sich die Palm-Matrix als neuronales Netz mit einer Input- und einer Output-Schicht interpretieren lässt. In der Initialsierungsphase berechnet das Netz aus den zu speichernden Mustern die Verbindungsstärken. Durch Anlegen eines Input-Vektors ordnet das Verfahren alle Muster nach Ähnlichkeit zum Input.

Bentz et al. 1989 und Hagström 1996 untersuchen und entwickeln Verfahren zur effizienten Repräsentation der Matrix sowie zur effektiven Durchführung von Speicher- und Retrievalaktionen. Da die Matrizen nur sehr spärlich besetzt ist, berücksichtigt die Implementierung nur die mit Einsern gefüllten Zellen und verweist zwischen diesen mit einer Zeiger-Struktur. Dadurch wird SpaCAM sehr flexibel und kann dynamisch auch zur Laufzeit neue Muster aufnehmen. Bei Hopfield-Netzwerken ist dies problematisch, da die Architektur mit der Anzahl der Neuronen eine Obergrenze für die speicherbaren Muster setzt (cf. Abschnitt 3.5.3). SpaCAM ist gegenüber den Hopfield-Netzwerken nicht nur flexibel, sondern auch sehr effizient. Bentz et al. 1989 stellen fest, dass die Zugriffzeit von zwei Parametern abhängt, der typischen Anzahl von Termen pro Dokument und der Anzahl der Dokumente. Die Untersuchungen ergaben, dass die typische Anzahl von Termen oder Merkmalen der dominierende Parameter ist, und dass die Anzahl der Dokumente oder Muster nur eine sehr untergeordnete Rolle spielt. Variiert die Anzahl der Terme pro Dokument sehr stark, dann lohnt sich zu Effizienzsteigerung nach Bentz et al. 1989 sogar die Aufteilung der Kollektion in homogene Cluster.

Die Repräsentation der Muster bei Hagström 1996 analysiert ein Wort als Sequenz von Zeichen mit ASCII-Werten. Bei Wörtern der Länge n und m darzustellenden ASCII-Zeichen enthält der Repräsentationvektor n mal m Stellen, wovon nur n besetzt sind. Dies erzeugt äußerst spärlich besetzte

Vektoren, die sich auf die Effizienz der Verfahrens positiv auswirken (cf. Schwenker et al. 1996, Hagström 1996).

Hagström 1996 setzt SpaCAM für das Retrieval von Wörtern aus sehr großen Textmengen ein. Die hohe Fehlertoleranz liefert auch bei Tippfehlern im Input sinnvolle Ergebnisse und liefert in der Datenbank enthaltene falsche Schreibweisen. Dabei ist eine Obergrenze für die Wortlänge vorgesehen. Komplexere Anfragen aus mehreren Wörtern werden einzeln gestellt und die Ergebnislisten mit Fuzzy-Operatoren kombiniert. Diese Anwendung hat Eingang in mehrere kommerzielle Information Retrieval Anwendungen gefunden.

Ein weiteres assoziatives Speichermodell ist das Random Neural Network, das ähnlich wie das Hopfield-Netzwerk funktioniert. Ein besonderes Merkmal ist, dass der Zufall die Aktivierungsausbreitung steuert. Eine probabilistische Funktion bestimmt, welches Neuron als nächstes feuert. Stafylopatis/Likas 1992 setzen das Random Neural Network zum Speichern und Retrieval von Bildern ein, die von Merkmals-Vektoren repräsentiert werden. Ihr Netz ist hierarchisch organisiert. Zunächst versucht ein globales assoziatives Netz, den Input einem der gespeicherten Muster zuzuordnen. Treten dabei Fehler auf, versuchen weitere Netze aus Teilen der Repräsentation, für die sie das Training spezialisierte, das korrekte Muster zu finden.

4.3 Spreading-Activation-Modelle

In diesem Abschnitt wird das am weitesten verbreitete Information Retrieval Modell auf der Basis neuronaler Netze vorgestellt, das die IR-Literatur meist als Spreading-Activation-Modell bezeichnet. Dieser Name ist allerdings nicht sehr aussagekräftig, da alle künstlichen neuronalen Netze auf dem Prinzip der sich ausbreitenden Aktivierung beruhen. Der Begriff Spreading-Activation-Netzwerk identifiziert also keine besondere Klasse von Modellen.

Obwohl die Spreading-Activation-Netzwerke eindeutig neuronale Netze mit allen entscheidenden Merkmalen sind, lassen sie sich keinem der gängigen Typen zuordnen. Spreading-Activation-Netzwerke haben in der Regel bidirektionale Verbindungen, sind jedoch keine klassischen Hopfield-Netzwerke, da sie über keine Energie-Funktion verfügen. Manche der Netze sind lernfähig. Ihre Lernverfahren sind von der Perzeptron-Lernregel abgeleitet, jedoch haben sie im Gegensatz zum Perzeptron nicht nur vorwärtsgerichte Verbindungen. Am nächsten stehen die Spreading-Activation-Netzwerke dem Pattern Associator Modell, das in McClelland/Rumelhart 1988 vorgestellt wird. Der Pattern Associator ist ein einfaches lineares Modell, das aus einer Schicht von Input-Units und einer Schicht von Output-Units besteht. Darin ist

Lernen nach der Perzeptron-Lernregel vorgesehen. Allerdings verfügen Pattern Associator Systeme nur über Verbindungen in eine Richtung. Das etwas allgemeinere Modell in McClelland/Rumelhart 1988 ist das Interactive Activation and Competition Modell, das hemmende Verbindungen innerhalb der Schichten realisiert. Verbindungen innerhalb von Schichten kommen auch in manchen Spreading-Activation-Netzwerken vor, doch sind sie meistens verstärkend und implementieren so assoziative Beziehungen etwa zwischen Termen oder Dokumenten. Das Interactive Activation and Competition Modell ist jedoch nicht lernfähig.

Die Spreading-Activation-Netzwerke können also nicht eindeutig einer Klasse von Netzwerken zugeordnet werden. Da somit in der Neuroinformatik kein passender Begriff vorliegt, wird in der weiteren Darstellung der in der IR-Literatur übliche Begriff Spreading-Activation-Netzwerke beibehalten.

Eine Beschreibung der Spreading-Activation-Netzwerke erfolgt am besten über die Eigenschaften. Dazu eignet sich das in Abschnitt 3.3 vorgestellte Schema von Rumelhart et al. 1986. Demnach sind die Spreading Activation Netzwerke in Schichten aufgeteilte Netze mit bidirektionalen Verbindungen zwischen den Schichten. Bei lernenden Modellen kommen adaptierte Perzeptron-Regeln und damit einfache Delta-Regeln zum Einsatz.

In diesem Rahmen bewegen sich die Spreading-Activation-Netzwerke, wobei zwischen den einzelnen Ansätzen Unterschiede bestehen. So gibt es teilweise Verknüpfungen innerhalb der Schichten und unterschiedliche Aktivierungsfunktionen kommen zum Einsatz. Diese Variationen werden bei den einzelnen Systemen vorgestellt. Da alle diese Systeme eine sehr einfache Struktur besitzen, die im Wesentlichen nur das Grundprinzip der Aktivierungsausbreitung entlang gewichteter Verbindungen festlegt, ist der Name Spreading-Activation-Netzwerke sozusagen als kleinster gemeinsamer Nenner durchaus gerechtfertigt.

Da sich die Ansätze untereinander stark ähneln, beginnt dieses Kapitel mit einer Darstellung ihrer Funktionsweise. Im Anschluss (Abschnitt 4.3.2.1 bis 4.3.2.6) werden einige wichtige Systeme ausführlich diskutiert. Weitere Systeme werden nur kurz vorgestellt.

4.3.1 Funktionsweise eines Spreading-Activation-Netzwerks

Die Prinzipien des Grundmodells des Spreading-Activation-Netzwerks tauchen bei allen Systemen auf. Darin finden sich die typischen IR-Objekte Dokumente, Indexterme und Anfragen. Das Standard-Modell modelliert die Indexterme und die Dokumente als künstliche Neuronen.

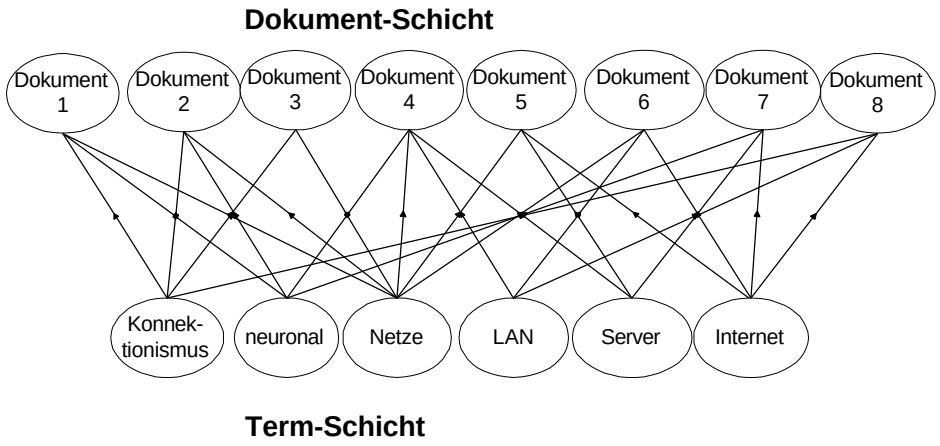


Abbildung 4-3: Zweischichtiges Spreading-Activation-Netzwerk mit beispielhaften Termen

Die Initialisierung der Gewichte der Verbindungen erfolgt anhand der Dokument-Term-Matrix aus der Indexierung. Der Benutzer formuliert die Anfrage als Liste von Termen. Das System aktiviert dann die gewählten Terme und die Aktivierung breitet sich im Netz aus. Zunächst werden die Dokument-Neuronen aktiviert, mit denen die Anfrage-Terme indexiert sind. Alle aktivierten Dokumente senden im zweiten Schritt Aktivierung an alle Terme, mit denen sie verknüpft sind. Nach einer bestimmten Anzahl von Schritten oder nachdem ein bestimmter Aktivierungswert erreicht ist, endet die Aktivierungs-Ausbreitung und das System präsentiert dem Benutzer die am stärksten aktivierten Dokumente als Ergebnis.

Spreading-Activation-Modelle besitzen im Rahmen der IR-Forschung eine hohe Plausibilität. Die Aktivierung eines Dokuments durch seine Indexterme ist eine einleuchtende Metapher. Der assoziative Charakter des IR-Prozesses tritt dabei deutlich hervor. Term-Expansion und Relevanz-Feedback ergeben sich in diesem Rahmen sehr natürlich als inhärente Eigenschaften des Modells. Im Folgenden werden die einzelnen Phasen des Prozesses genauer analysiert.

4.3.1.1 Initialisierung

Die Spreading-Activation-Modelle bestehen aus zwei Schichten von Neuronen, die untereinander verbunden sind. Die Verbindungen im neuronalen Netz verlaufen also zwischen Dokumenten und Termen. Das Gewicht der Verbindungen muss demnach ein Maß für die Beziehung zwischen einem Dokument und einem Term sein. Diese Beziehung und ihr Wert spielt auch in den wich-

tigsten IR-Modellen die entscheidende Rolle für das Ranking. Das Gewicht stammt vom Gewichtungsalgorithmus der Indexierung (cf. Abschnitt 2.1.1). Die Dokument-Term-Matrix entspricht im Spreading-Activation-Modell der Verbindungsmatrix. Ein detaillierter Vergleich zwischen den beiden Matrizen, der weitere Verbindungsmöglichkeiten berücksichtigt, folgt in Abschnitt 4.3.3.

Die Netzwerkmodelle implementieren also kein neuartiges Inhaltsanalyse- und Repräsentationsverfahren, sondern übernehmen eine Dokument-Term-Matrix. Die Verbindungen übernehmen die Gewichte der entsprechenden Zelle der Dokument-Term-Matrix. Für das Netzwerk-Modell wie für das Vektorraum-Modell ist es unerheblich, ob die Dokument-Term-Matrix durch intellektuelle oder automatische Indexierung gewonnen wurde. Bei der Besprechung der einzelnen Systeme werden die jeweiligen Gewichtungsformeln vorgestellt.

Tabelle 4-1 : Dokument-Term-Matrix für das Beispiel aus Abbildung 4-1

Dokument / Term	Dokument 1	Dokument 2	Dokument 3	Dokument 4	Dokument 5	Dokument 6	Dokument 7	Dokument 8
Konnektionismus	0,8	0	0,8	0	0	0	0	0,6
Neuronal	0,6	0,8	0	0	0	0	0	0
Netze	0,6	0,8	0	0,8	0,8	0,8	0	0
LAN	0	0	0	0,6	0	0,6	0	0
Server	0	0	0	0,4	0,4	0	0,6	0
Internet	0	0	0	0	0,6	0,4	0,4	0,6

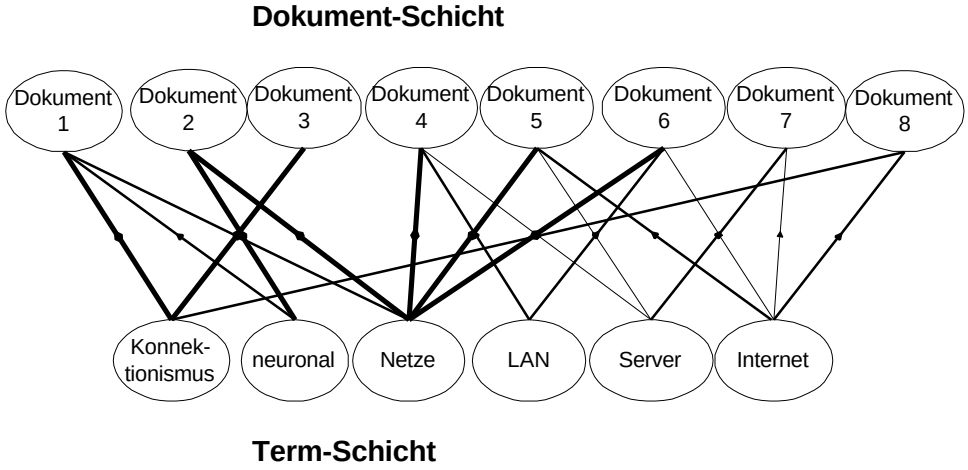


Abbildung 4-4: Initialisierung durch Setzen der Gewichte: Die Verbindungen des Netzes haben die Werte aus der Dokument-Term-Matrix in Tabelle 4-1 übernommen. Die drei unterschiedlichen Linienstärken repräsentieren die Werte in den Zellen (dickste Linie entspricht 0,8). Verbindungen mit dem Gewicht Null sind nicht eingezeichnet.

Die Verbindungen in den Spreading-Activation-Netzwerken sind bidirektional. Die Aktivierung läuft also sowohl von den Termen zu den Dokumenten als auch zurück. Die Gewichte der Verbindungen sind in fast allen Modellen symmetrisch. Sie besitzen also in beide Richtungen das gleiche Gewicht. Damit hat die Beziehung zwischen Term und Dokument den gleichen Wert wie die zwischen Dokument und Term. Die meisten IR-Modelle arbeiten mit dieser Annahme, die zwar sehr plausibel, aber nicht selbstverständlich ist. Die Term-Gewichte berechnen sich in der Regel aus Sicht der Terme, wie etwa die inverse Dokument-Frequenz zeigt (idf, cf. Abschnitt 2.1.1). Betrachtet man die Gewichte des neuronalen Netzes aus Richtung der Dokumente, ist die Berechnung einer analog definierten inversen Term-Frequenz denkbar, welche die Anzahl der Terme eines Dokuments berücksichtigt und zu völlig anderen Werten führt.

Die Frage, ob Null-Werte in der Matrix Verbindungen mit Null-Werten entsprechen, oder ob die diese Verbindungen nicht existieren, spielt prinzipiell eine untergeordnete Rolle. Nachträgliche Lernprozesse weisen diesen Links eventuell noch Werte ungleich Null zu, deshalb sollten sie entweder vorhanden sein oder kreiert werden. Sind die Verbindungen mit dem Wert Null nicht vorhanden, stört dies die Äquivalenz der Dokument-Term-Matrix mit der Verbindungsmatrix nicht. Auch in den Implementierungen von Dokument-Term-Matrizen im Information Retrieval und bei der maschinellen Verarbeitung von spärlich besetzten Matrizen allgemein sind diese Werte nicht

vorhanden. So gibt etwa das Harwell-Boeing-Format für spärlich besetzte Matrizen die Zellen grundsätzlich zeilenweise an. Dabei werden nur Zellen mit Werten ungleich Null und Zeilenwechsel notiert, so dass ein sehr kompaktes Format entsteht (cf. Berry et al. 1996).

Bei der Initialisierung handelt es sich nicht um Lernen. Zwar werden die Parameter des Modells bei diesem Prozess verändert, jedoch fehlen grundlegende Eigenschaften eines Lernprozesses. Lernen in neuronalen Netzen geht in der Regel von einer zufälligen Initialisierung der Gewichte der Verbindungen aus. Von diesem zufällig gewählten Punkt im n-dimensionalen Raum ausgehend verändert ein Lernalgorithmus aufgrund einzelner externer Einflüsse der präsentierten Muster die Gewichte. Dadurch verändert sich das Verhalten des Modells zu den Mustern. Die reine Übernahme von Werten, die ein eindeutig nicht-lernendes Verfahren ermittelt, ist somit kein Lernen im Sinne der Neuroinformatik. Allerdings kann auf dem so erreichten Stand ein Lernprozess einsetzen, der auf äußere Einflüsse reagiert und die Verbindungsgewichte verändert. Allgemein gesprochen ist Lernen ein Prozess, bei dem sich ein System durch Verändern von Parametern an Einflüsse aus der Umwelt anpasst und damit ein iterativer und interaktiver Vorgang. Auch aus dieser Perspektive wird deutlich, dass das einmalige Berechnen von Parametern keinesfalls Lernen darstellt.

4.3.1.2 Anfrage-Formulierung

Die Anfrage besteht im Information Retrieval für den Benutzer aus einer Liste von Begriffen. Diese wählt er entweder aus einer Liste der vorhandenen Terme aus oder gibt sie selbst ein. Die Neuronen, die diese Terme repräsentieren, werden aktiviert. In der Regel erhalten sie die maximal mögliche Aktivierung, die meist Eins beträgt. Aber auch eine Gewichtung der Terme der Anfrage ist denkbar.

Im Information Retrieval bilden grundsätzlich die Anfragen den Input und die Ergebnis-Dokumente den Output, was auch für die Spreading-Activation-Netzwerke zutrifft. Ebenso wie das Vektorraum-Modell sind auch die Spreading-Activation-Netzwerke in dieser Hinsicht flexibel. Das heißt, neben Termen können auch Dokumente als Input dienen. So kann ein System zu bereits als relevant bekannten Dokumenten weitere ähnliche Dokumente suchen. Dazu erhält im ersten Schritt nicht ein Term sondern ein Dokument Aktivierung. Der weitere Prozess verläuft dann völlig analog.

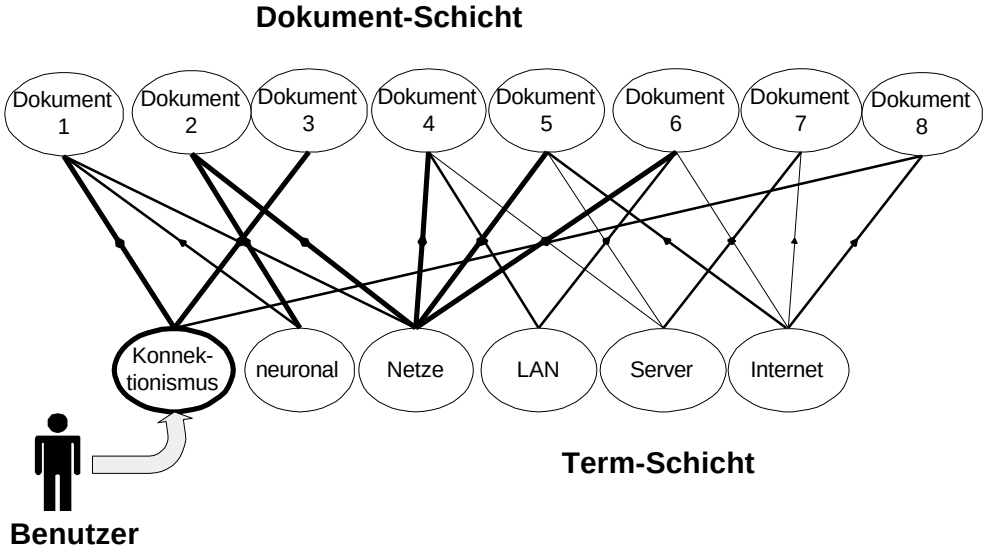


Abbildung 4-5: Anfrage als Setzen von Aktivierung: Die Neuronen der in der Anfrage vorkommenden Terme sind aktiviert. In diesem Beispiel besteht die Anfrage aus dem Begriff *Konnektionismus*. Die Aktivierung des Neurons ist hoch, wie die Stärke der Linien andeutet.

Die Spreading-Activation-Netzwerke sind noch flexibler und erlauben im Prinzip auch eine gemischte Eingabe, die sowohl aus Anfragen als auch aus Dokumenten besteht. Diese Möglichkeit bleibt zudem im laufenden Prozess erhalten. Sie taucht als Relevanz-Feedback in Abschnitt 4.3.1.4 noch einmal auf.

4.3.1.3 Retrieval

Der nächste Schritt im IR-Prozess besteht aus der Ausbreitung von Aktivierung im Netzwerk, also aus Spreading-Activation. Die Aktivierung verläuft entlang der Verbindungen und abhängig von deren Gewichten. Die Ausbreitung wird von der Ausbreitungsfunktion kontrolliert, die in den meisten Modellen sehr einfach ist:

$$\textit{Ausbreitungsfunktion: } Input_j = Output_i \, w_{ij}$$

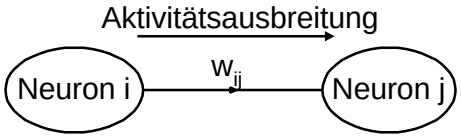


Abbildung 4-6: Aktivierungsfluss zwischen zwei Neuronen

Im ersten Schritt aktivieren die vom Benutzer gewählten Terme die Neuronen in der Dokument-Schicht. Je nach Stärke der Verbindung zwischen Dokument und Term und der Stärke des Gewichts des Anfrage-Terms aktiviert die Aktivierungsfunktion die Dokumente verschieden stark. Im nächsten Schritt läuft die Aktivierung auch von den Dokumenten zu den Termen. Ausgehend von allen aktivierten Dokumenten werden nun deren Terme aktiviert. Dabei können auch Terme aktiviert werden, welche die ursprüngliche Anfrage nicht enthielt. Damit ist im Modell die implizite Expansion der Anfrage um weitere Begriffe angelegt. Die neuen Terme kommen mit den vom Benutzer aktivierten häufig gemeinsam in Dokumenten vor, so dass die Term-Expansion auf Kookkurrenzen beruht (cf. Abbildung 4-7).

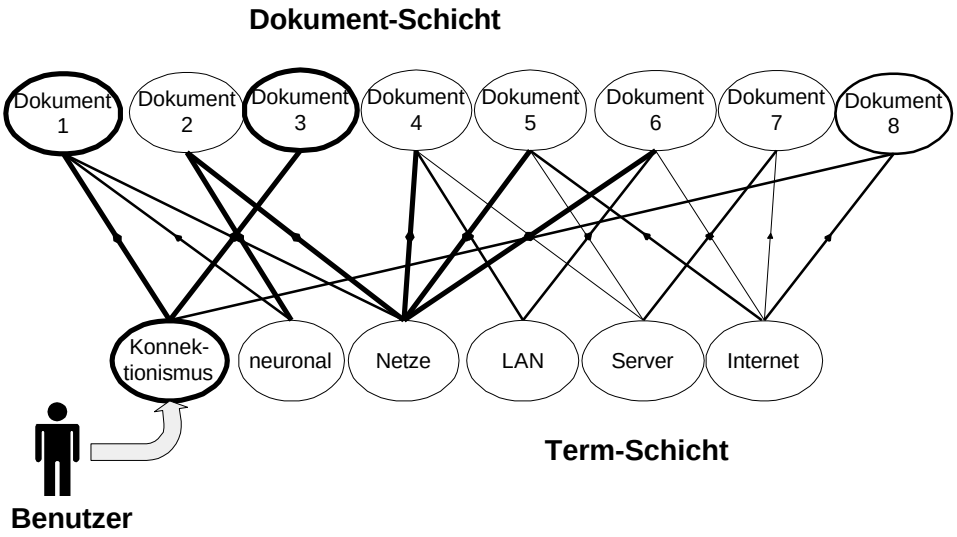


Abbildung 4-7: Aktivierung nach der ersten Phase: Die Aktivierung ist von den Term-Neuronen zur Dokument-Schicht geflossen. Dadurch wurden alle Dokumente aktiviert, die mit dem Term *Konnektionismus* verbunden sind. Je nach Stärke ihrer Verbindung mit dem Term erhielten sie mehr oder weniger Aktivierung, was die unterschiedlichen Stärken der Linien andeuten. Bei den aktivierten Dokumenten handelt es sich um diejenigen, die mit dem Term *Konnektionismus* indexiert sind, die also diesen Term enthalten.

Die Aktivierung der Dokument-Knoten ist eine Funktion von Input-Vektor und Verbindungsmatrix.

$$Aktivierung_I = f\,(Input_A, W)$$

Die Aktivierung eines einzelnen Neurons lässt sich unter Berücksichtigung der Output-Funktion der Term-Neuronen, der Propagierungsregel, der Input-Funktion als Summenfunktion und der Aktivierungsfunktion der Dokument-Neuronen berechnen:

$$Aktivierung_i = f\_akt(\sum_j f\_out(Aktivierung_j)\, w_{ij})$$

Sind sowohl Output- als auch Aktivierungsfunktion die Identitätsfunktion, so ergibt sich nach dem ersten Schritt folgende Aktivierung:

$$Aktivierung_i = \sum_j Aktivierung_j\, w_{ij}$$

Je nach Modell setzt sich die Aktivierungsausbreitung in weiteren Schritten fort. Durch die fortschreitende indirekte Aktivierung immer neuer Knoten werden möglicherweise auch Terme aktiviert, die semantisch sehr weit von den ursprünglichen Anfrage-Termen entfernt sind. Je nach Verknüpfungsgrad des Netzes kann eventuell sehr schnell ein großer Teil des Netzes aktiviert werden. Das kann sich negativ auf die Retrievalqualität auswirken, da das System bei vielen hoch aktivierten Dokumenten nicht mehr gut diskriminiert. Zudem entspricht hohe Aktivierung in vielen Dokument-Neuronen nicht der Realität eines Information Retrieval Prozesses, bei dem ein System in der Regel nur relativ wenige Dokumente aus der Grundmenge als relevant einstuft.

Der Entwickler eines Netzes muss eine Balance zwischen zu geringer und zu starker Ausbreitung finden. Zu geringe Ausbreitung der Aktivierung nutzt die Vorteile des Spreading-Activation-Ansatzes nicht aus. Diese bestehen gerade darin, dass auch Terme aktiviert werden, die nicht in der Anfrage enthalten waren. Im Idealfall findet das System so semantisch ähnliche Begriffe.

Die starke Expansion der Anfrage durch zahlreiche Ausbreitungsschritte oder einen hohen Verknüpfungsgrad führt in der Regel zu mehr gefundenen Dokumenten. Damit erhöht es den Recall und vermindert die Precision (cf. Abschnitt 2.1.4.1), während sich eine geringe Ausbreitung umgekehrt auswirkt. Je nach Benutzersicht kann in einer Retrievalsituation der Recall oder

die Precision wichtiger sein. Den Expansionsgrad könnte ein Benutzer über die Zahl der Ausbreitungsschritte steuern.

In jedem Fall ist es sinnvoll, die Aktivierungsausbreitung auch zu hemmen. Folgende Strategien sind dabei denkbar:

- Die Anzahl der erlaubten Aktivierungsschritte wird beschränkt.
- Die gesamte Aktivierung des Netzes wird beschränkt:
  - Die Summe der Aktivierung im gesamten Netz muss immer konstant sein.
  - Die Summe der Aktivierung, die jedes Neuron aussendet, besitzt ein Maximum.
- Durch einen Verfallsfaktor (decay) verlieren alle Neuronen im Netz bei jedem Schritt bei bestimmtes Maß an Aktivierung. Dadurch ist sichergestellt, dass nur Neuronen, die bei vielen Schritten angesprochen werden, aktiv bleiben, während gleichzeitig das Rauschen von schwach und einmalig aktivierten Neuronen ohne Folgen bleibt.
- Die Neuronen einer Schicht sind untereinander vollständig mit hemmenden Verbindungen verknüpft. Dadurch wird ein Wettbewerb zwischen den Knoten ausgelöst, der in sogenannten Winner-Take-All-Netzwerken so weit ausgeprägt ist, dass immer nur ein Neuron in einer Schicht *feuert* (cf. Zell 1994:191). In Spreading-Activation-Netzwerken ist eine solche extreme Ausprägung nicht sinnvoll, da immer mehrere Dokumente und Terme aktiviert sein sollen. Ein einziger aktiver Term in der Term-Schicht führt kaum zur gewünschten zusätzlichen Aktivierung von semantisch ähnlichen Termen. Und wird nur ein Dokument-Neuron aktiviert, erhält der Benutzer eine Antwortmenge mit nur einem Dokument.

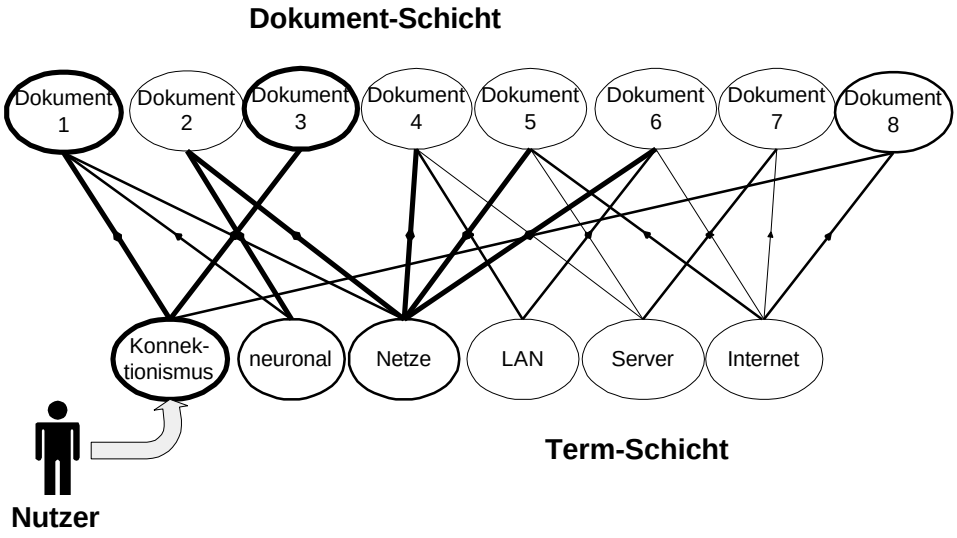


Abbildung 4-8: Automatische Termerweiterung nach mehreren Schritten: Beim zweiten Schritt senden die im ersten Schritt aktivierten Dokumente ihrerseits Aktivierung aus. Dadurch erhalten alle mit ihnen verbundenen Terme Impulse. So führt v.a. die hohe Aktivierung von Dokument 1 entlang der starken Verbindungen zu *neuronal* und *Netze* zu einer positiven Aktivierung in diesen beiden Knoten.

Die durch das Netz fließende Aktivierung kann semantisch als Relevanz oder Interesse interpretiert werden. Das Interessenspektrum, das hinter einer Anfrage steht, läuft durch das Netz und aktiviert relevante Dokumente und Terme. Ergebnis am Ende der Aktivierungsausbreitung sind die am stärksten aktivierten und damit interessantesten Dokumente. Die Aktivierung jedes Dokuments stellt dessen Retrieval Status Value (RVS) dar.

$$RSV_i = Aktivierung_i$$

Die Grundidee der Spreading-Activation-Netzwerke ist intuitiv einleuchtend. Jedoch sind wie bei der Arbeit mit neuronalen Netzen fast immer zahlreiche heuristische Entscheidungen notwendig.

- Der genaue Ablauf der Aktivierungsausbreitung
  Durch den Aufbau des Netzes in Schichten ergibt sich die Möglichkeit, die Ausbreitung ebenfalls nur schichtweise zu erlauben. Dann sendet pro Phase eine Schicht und die andere empfängt. Das andere Szenario besteht darin, alle Neuronen in einer Phase gleichzeitig senden zu lassen, für alle

das neue Input-Signal und anschließend die neuen Aktivierungslevel zu berechnen.

- Behandlung der Anfrage-Terme
  Die Anfrage-Terme können nach der initialen Aktivierung den Aktivierungsregeln überlassen werden und somit auch abgeschwächt werden oder sie werden konstant auf dem Anfangswert gehalten (*clamped*).
- Wahl der Aktivierungs-, Input- und Outputfunktion und ihrer Parameter

4.3.1.4 Relevanz-Feedback

Bei Relevanz-Feedback bewertet ein Benutzer die Dokumente eines Zwischenergebnisses und das System optimiert davon ausgehend die Anfrage (cf. Abschnitt 2.1.2.2 und Abschnitt 2.1.2.3). Wie die Term-Erweiterung ergibt sich auch Relevanz-Feedback als inhärente Eigenschaft des Spreading-Activation-Modells, für das keine neuen Konzepte erforderlich sind. Die Veränderung von Aktivierung einzelner Neuronen, wie sie bei der Anfrage nötig ist, und die darauf folgende Ausbreitung reichen aus. Lediglich der Zeitpunkt des Eingriffs des Benutzers ändert sich. Während die Anfrage-Terme zu Beginn aktiviert werden, greift der Benutzer bei Relevanz-Feedback bei einem Zwischenstand ein und modifiziert die Aktivierung einzelner Neuronen. In der Regel betrachtet der Benutzer Dokumente und bewertet, ob sie für sein Informationsbedürfnis relevant sind oder nicht. Entsprechend wird die Aktivierung verändert. Relevante Knoten erhalten eine höhere oder maximale Aktivierung und nicht relevante Knoten verringern ihre Aktivierung oder setzen sie auf Null.

Relevanz-Feedback kann in Spreading-Activation-Netzen sehr flexibel ablaufen. Grundsätzlich ist jederzeit ein Eingriff möglich und neben Dokumenten kann ein Benutzer auch die bisher aktivierten Terme bewerten. Auch dieses Verfahren kann negative Auswirkungen von zu starker Aktivierungsausbreitung mindern. Prinzipiell könnte ein Benutzer nach jedem Schritt die aktivierten Terme und Dokumente betrachten und ihre Aktivierungswerte verändern. Das Funktionsprinzip des Systems bliebe dabei gleich.

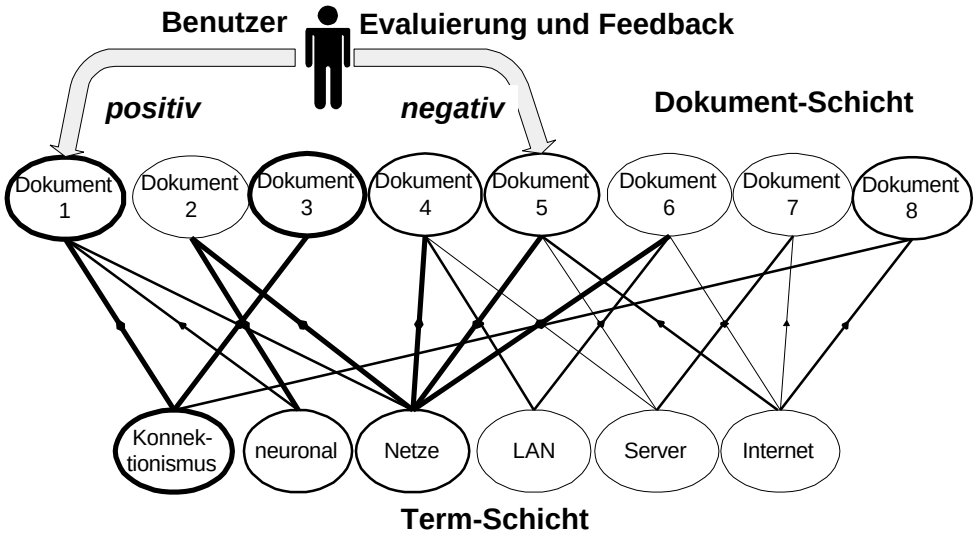


Abbildung 4-9: Relevanz-Feedback als Modifizierung der Aktivierungswerte von Neuronen: Zu einem bestimmten Zeitpunkt betrachtet der Benutzer die momentan am stärksten aktivierten Dokumente, bewertet sie, und das System nutzt diese Information. Im vorliegenden Beispiel erhält der Benutzer die Dokumente 1, 3, 4, 5 und 8 und bewertet 1 als positiv und 5 als negativ. Vor den nächsten Ausbreitungsschritt erhöht das System die Aktivierung von Dokument 1 und verringert die von Dokument 5.

Im Detail werden die Informationen über Relevanz und Nicht-Relevanz des Benutzers verschieden eingesetzt. Vergibt der Benutzer eine positive Bewertung für ein Dokument, so kann seine Aktivierung einmalig auf einen entsprechenden Wert gesetzt werden oder um einen bestimmten Wert erhöht werden. Die Höhe der Aktivierung in den nächsten Schritten kann dem Aktivierungsverlauf und der Aktivierungsregel überlassen werden. Alternativ dazu kann die Aktivierung auch auf dem gewünschten Level konstant gehalten werden. Dann behält das Neuron diese Aktivierung für den Rest des gesamten Prozesses. Das entsprechende positiv bewertete Dokument *belohnt* dann im weiteren Verlauf immer wieder die darin enthaltenen Terme. Daneben verwenden einige Systeme Relevanz Information der Benutzer zum Lernen und verändern damit das Modell durch Einstellen der Verbindungen.

4.3.2 Beispiele für Spreading-Activation-Netzwerke

Die oben beschriebene Funktionsweise der Spreading-Activation-Netzwerke ist in den folgenden Systemen realisiert. Die wichtigsten Systeme werden in

eigenen Abschnitten diskutiert und die Abschnitte 4.3.2.5 bis 4.3.2.6 fassen weitere Systeme zusammen.

4.3.2.1 Probabilistic Indexing and Retrieval-Component-System

Ein typisches Beispiel für ein Spreading-Activation-Netz bietet Kwok 1989, der eines der ersten Modelle dieser Art vorstellte. Lernfähigkeit ist vorgesehen und wird von Kwok 1991 zum Lernen aus Relevanz-Feedback ausgebaut. Das System Probabilistic Indexing and Retrieval-Component-System (PIRCS) an mehreren Runden der TREC Evaluierungsstudie teil (cf. Kwok et al. 1993, Kwok/Grunfeld 1994, cf. Abschnitt 4.8).

Kwok versteht seinen Ansatz als „attempt to employ the NN [neural networks] paradigm to reformulate the probabilistic model of IR“ (Kwok 1989: 21). Darin zeigt sich bereits die Nähe der Spreading-Activation-Netzwerke zu den Standard-Verfahren im Information Retrieval.

Gegenüber dem oben dargestellten Standard-Ansatz zeichnet sich PIRCS durch einige Besonderheiten aus. Es besteht aus drei Schichten, wobei die mittlere Schicht Terme repräsentiert, die zur einen Seite mit den Anfragen und zur anderen mit den Dokumenten verbunden sind.

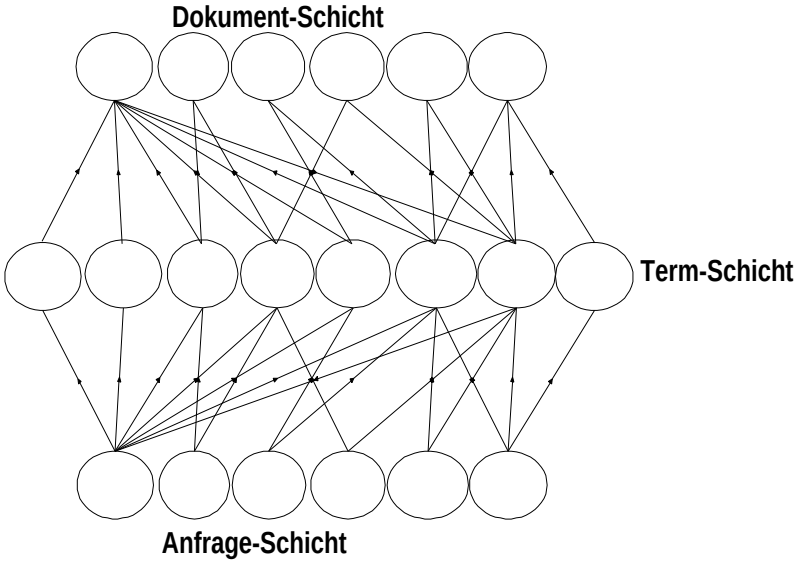


Abbildung 4-10: Dreischichtiges Netz nach Kwok 1989

Sowohl Anfragen als auch Dokumente können als Input dienen, d.h. das Netz kann auch die relevanten Anfragen für ein Dokument zurückliefern. Dies ist der Fall bei Filter- oder Routing-Aufgaben. Dabei bleibt die Anfrage als Interessensprofil über einen längeren Zeitraum konstant und Dokumente, die

neu zur Kollektion hinzukommen, werden diesen Profile zugeordnet. So identifizieren etwa Filtersysteme in einem Strom von aktuellen Nachrichten die, welche für ein Benutzerprofil besonders interessant sind. Routing-Aufgaben spielen auch in TREC eine Rolle (cf. Womser-Hacker 1997). Dies ist durchaus sinnvoll, da das Korpus hauptsächlich aus Zeitungstexten und Nachrichtenagentur-Meldungen besteht. PIRCS bearbeitete in TREC mehrfach sowohl Routing- als auch Ad-Hoc-Aufgaben (cf. Abschnitt 4.8). Auch für Routing-Aufgaben funktioniert das System völlig analog zu dem oben diskutierten Grundmodell für Spreading-Activation-Netzwerke. Lediglich die initiale Aktivierung erfolgt je nach Aufgabentyp in der Anfrage- oder Dokument-Schicht.

Die Übereinstimmung von PIRCS zum geschilderten Standard-Modell geht jedoch noch weiter. Betrachtet man das Ad-Hoc-Retrieval, also den Normalfall mit einer Anfrage als Eingabe und Dokumenten als Ausgabe, führt die Anfrage-Schicht zu keiner konzeptuellen Änderung und könnte auch weggelassen werden. Nach Kwok reduziert sich das dreischichtige Netz je nach Art des Retrievals zu einem zweischichtigen System. Folgende Gründe sprechen gegen die Einführung einer Anfrage-Schicht:

- Die Anfrage-Schicht leitet lediglich die Aktivierung zu den Anfrage-Termen in der Term-Schicht. Im zweischichtigen Modell erfolgt diese Aktivierung durch den Benutzer direkt in der Term-Schicht (cf. Abschnitt 4.3.1.1).
- Die Anfragen sind beim Ad-Hoc-Retrieval keine konstanten Objekte. Für jede Anfrage müsste vielmehr ein neues Neuron hinzugefügt werden. Zu Beginn des Systemeinsatzes wäre die Schicht damit leer und würde mit der Zeit anwachsen. Die Wiederverwendung einer Anfrage kommt selten vor. Damit dient die Anfragen-Schicht lediglich der Speicherung alter Anfragen. Im Falle von Routing verhält es sich genau umgekehrt, dann sind die Anfragen die konstanten Objekte und die Dokument-Schicht ist nicht notwendig.
- Dokumente und Anfragen können im IR als Objekte gleichen Typs betrachtet werden, die jeweils durch Terme beschrieben werden. Dies gilt nach Kwok auch für sein Modell (cf. Kwok 1989:25). Betrachtet man die Eigenschaften eines Anfragen- und eines Dokument-Neurons, so sind beide durch Verbindungen zur Term-Schicht gekennzeichnet. Damit könnten sie auch in einer Schicht angeordnet sein und ihre Funktionalität würde sich nicht ändern.

Die Einführung von Anfrage-Knoten und ihre Zusammenfassung in einer eigenen Schicht dient also nur der Übersichtlichkeit und ist keine konzeptuelle

Neuerung. Auch die Anpassung eines Netzwerks an Routing-Aufgaben erfordert nicht zwingend die Einführung einer dritten Schicht. Beim Routing ersetzen entweder die Anfragen die Dokumente oder beide Dokumente stehen gleichzeitig in einer Schicht. Alle diese Fälle erfasst man, wenn man bei dem zweischichtigen Standard-Modell von einer Objekt-Schicht und einer Eigenschafts-Schicht spricht.

Die Details der Initialisierung und Veränderung der Gewichte in PIRCS wird im Folgenden am Beispiel von Kwok et al. 1993 erläutert, wobei die Unterschiede zu den anderen Aufsätzen minimal sind. Verbindungen innerhalb der Schichten existieren bei PIRCS nicht. Die Gewichte der Verbindungen sind asymmetrisch, haben also in beiden Richtung verschiedene Werte. Die Anfangswerte der Verbindungen werden aus den Häufigkeiten der Terme in den Dokumenten und Anfragen abgeleitet, wobei folgende Formeln gelten:

$$w_{ki} = \ln\frac{p}{1-p} + \ln\frac{1-s_{ik}}{s_{ik}}, \qquad \text{wobei } s_{ik} = \frac{F_k - f_{ik}}{\text{Anzahl Terme} - L_i}$$

w_{ki} *Verbindung von Term k zu Dokument i*

p *Konst. p=1/50 L_i Länge Dokument i*

F_k, f_{ik} *Frequenz von Term k in der Kollektion bzw. Dokument i*

Kwok et al. 1993:155

$$w_{ik} = \frac{f_{ik}}{L_i} w_{ki} \quad \text{Verbindung von Dokument } i \text{ zu Term } k$$

L_i *Länge Dokument i*

f_{ik} *Frequenz von Term k in Dokument i*

Kwok et al. 1993:158

Die Formeln für die Verbindungsgewichte von den Termen zu den Dokumenten entsprechen im Wesentlichen Gewichtungsmaßen, die Standard IR-Systemen einsetzen (cf. z.B. Womser-Hacker 1997:105ff.). Darin wird sowohl die Term-Frequenz in der Kollektion als auch im Dokument berücksichtigt. In der Formel von Kwok et al. 1993:155 wirkt sich die Häufigkeit des Vorkommens eines Terms im Dokument nur minimal aus. Die Gewichte von den Dokumenten zu den Termen dagegen bestehen lediglich aus der längennormalisierten Dokument-Frequenz. Das Dokument aktiviert also die in ihm vorkommenden Terme ohne Rücksicht auf deren Vorkommen in der gesamten Kollektion. Vor dem Hintergrund der Betonung lokaler In-

formationsverarbeitung in neuronalen Netzen erscheint dies durchaus plausibel. Das Dokument initialisiert die Verbindung in seiner Richtung nur unter Verwendung von Wissen, das im Dokument selbst steckt. Aus der anderen Richtung ist dieses Prinzip aber nicht realisiert. Die Terme initialisieren die Verbindungen zu den Dokumenten unter Zuhilfenahme globalen Wissens. Kwok et al. 1993 begründen diese Entscheidung nicht.

Lernen aus Relevanz-Feedback-Informationen verändert nur die Gewichte zu den Dokumenten, während die Verbindungen zu den Termen konstant bleiben. Auch diese Entscheidung wirkt willkürlich. Die Lernregel erinnert zum einen an das probabilistische IR-Modell und zum anderen an die Delta-Regel. Vor dem eigentlichen Lernen aus Relevanz-Feedback wird mit den gleichen Formeln sogenanntes *Self-Learning* durchgeführt, bei dem das Dokument als relevant zu sich selbst markiert wird. Dadurch gelangt keine neue Information in das Netz, so dass *Self-Learning* nur einen weiteren Schritt bei der Initialisierung darstellt. Dabei werden die Verbindungen von den Termen zu den Dokumenten durchlaufen, das entsprechende Dokument erhält die Aktivierung Eins zugewiesen und die Spreading Activation läuft einmal in die Term-Schicht. Die Aktivierung der Terme ist dann der entscheidende Faktor der Lernregel. Für diese Aktivierung spielen die Verbindungen von den Dokumenten zu den Termen eine große Rolle, bei deren Initialisierung die Kollektionsfrequenz keine Rolle spielt. Damit stellt das *Self-Learning* eine Stärkung des Einflusses der Termfrequenz in den Term-Dokument-Verbindungen dar. Die folgende Formel bestimmt den Lernprozess im Ad-Hoc-Betrieb:

$$\Delta w_{ki} = \frac{h_D(x_k - p_{ki}^{old})}{p_{ki}^{old}(1 - p_{ki}^{old})}$$

Δw_{ki} *Änderung am Gewicht der Verbindung von Term k zu Dokument i*

x_k *Aktivierung Termk* h_D *Lernrate für Ad−Hoc*

p_{ki}^{old} *aktuelle Wahrscheinlichkeit, dass Dokument i relevant*

Kwok et al. 1993:158

Im Gegensatz zur allgemeinen Delta-Regel (cf. Abschnitt 3.4.3) kommt der Teaching-Input in der Formel nicht vor. Beim Retrieval in PIRCS läuft die Aktivierung im ersten Schritt von den Termen zu den Dokumenten. Relevanz-Feedback setzt die Aktivierung der relevanten Dokumente auf 1 und die Aktivierung breitet sich wieder in Richtung Term-Schicht aus. Damit wirkt sich der Teaching-Input indirekt auf die Aktivierung der Term-Knoten aus, die

sich dann als Faktor in der obigen Formel niederschlägt. Die Wahrscheinlichkeit p_{ki}, dass Dokument i relevant ist, ergibt sich im probabilistischen Modell aus dem Gewichtungsfaktor von Term k für Dokument i. Dieser Faktor ist in PIRCS bereits in das Verbindungsgewicht von Term zu Dokument geflossen. Damit ist die Lernregel eine Funktion der Lernrate, der aktuellen Aktivierung in Term k und der aktuellen Verbindungsstärke zwischen Term und Dokument. In den Experimenten in Kwok 1991a zeigte sich, dass Lernen von relevanten Dokumenten Verbesserungen bringt, nicht aber das Lernen von nicht-relevanten Dokumenten. Grundlage waren die kleinen Kollektionen CACM und CISI, die in der IR-Literatur zwar häufig zu Tests herangezogen werden, aufgrund ihres geringen Umfangs aber nur zu Ergebnissen mit bedingter Aussagekraft führen (für eine Beschreibung cf. Baeza-Yates/Ribeiro-Neto 1999:91ff.).

Neben der Veränderung von Verbindungsgewichten kreiert das Lernen auch neue Verbindungen. Dies kann als Veränderung der Verbindungsstärke von Null auf einen Wert ungleich Null betrachtet werden. Die meisten neuronalen Netzwerkmodelle sehen vollständig verknüpfte Schichten vor, so dass die Schaffung der Verbindungen nicht gesondert behandelt wird.

Lernen durch die Schaffung von Verbindungen zwischen Dokumenten und Termen, die bisher nicht verbunden waren, stellen Kwok 1991a und 1991b vor. Später wurde diese Strategie auch in den TREC-Experimenten eingesetzt (cf. Abschnitt 4.8). Die gleichzeitige Aktivierung im Laufe einer Anfrage führt zu einer neuen Verbindung. Dabei werden nach dem Relevanz-Feedback die 15 bis 30 am stärksten aktivierten Terme, die nicht in der originalen Anfrage enthalten waren, auch mit den relevanten Dokumenten verbunden. Die Gewichte ergeben sich aus folgenden Formeln:

$$w_{ik} = a\, x_k$$

w_{ik} *Verbindung von Dokument i zu Term k*

a *Lernrate*

x_k *Aktivierung Term k*

$$w_{ki} = \ln \frac{p_{ki}}{1 - p_{ki}} + \ln \frac{Anzahl\,Terme}{F_K}$$

$$p_{ki} = b h_D x_k$$

w_{ki} *Verbindung von Term k zu Dokument i*

b, h_D *Lernraten*

F_k *Frequenz von Term k in der Kollektion*

Kwok et al. 1993:159

Diese Implementierung der Term-Expansion entspricht nicht der im Standard-Modell, bei dem sich die Expansion im Laufe der Aktivierungsausbreitung automatisch durch die Aktivierung weiterer Terme ergibt. PIRCS nutzt diese Fähigkeit des Spreading-Activation-Ansatzes nicht voll aus. Beim Retrieval fließt die Aktivierung nur einmal von den Anfrage-Termen zur Dokument-Schicht. Deshalb muss die Term-Expansion sich in konkreten Verbindungen niederschlagen. Nur beim Lernen erfolgt wie gezeigt ein weiterer Schritt und Aktivierung fließt wieder in die Term-Schicht zurück.

PIRCS orientiert sich stark am Vektorraum-Modell. Dies zeigen auch die Weiterentwicklungen. Kwok 1996 und Kwok/Chan 1998 z.B. benutzen kaum die Terminologie neuronaler Netze. Kwok 1996 befasst sich mit den Problemen kurzer Anfragen. Die Verbesserungen in Kwok/Chan 1998, die auf Pseudo-Relevanz-Feedback beruhen, das die ersten n Treffer als relevant betrachtet, sind auf jedes Vektorraum-Modell übertragbar. Dabei ließe sich gerade diese Technik mit einem Spreading-Activation-Ansatz mit mehreren Aktivierungsschritten elegant realisieren.

Unklar ist bei PIRCS wie bei vielen Modellen, in welchen Umfang Lernen das Modell verändert. Wie stark ändern sich die Verbindungen und wie hoch ist der Anteil der Verbindungen, die betroffen sind? Hat das Relevanz-Feedback entscheidenden Einfluss auf die Performanz des Netzes oder hängt die Qualität lediglich von der Initialisierung ab? Um dies zu messen, müsste die Retrieval-Qualität anhand von Test-Anfragen mit einem nur initialisierten Netz und mit einem durch Lernen modifizierten Netz gemessen werden.

Im Rahmen der TREC Konferenzen erprobten Kwok et al. 1993 und Kwok/Grunfeld 1994 das Modell an großen Datenmengen, wobei sie auch das Lernen durch Relevanz-Feedback einsetzten. In TREC 5 wurde PIRCS für Retrieval in Chinesisch adaptiert (Kwok/Grunfeld 1995, cf. auch Kwok 1997). Insgesamt erreichte PIRCS bei TREC sehr gute Ergebnisse und zählte mehrfach zu den acht besten Systemen für einzelne Kategorien, die der Überblicksartikel erwähnt (cf. Abschnitt 4.8). So gehörte PIRCS bei den TREC-Konferenzen 5 bis 7 zu den acht besten Systemen beim Standard Retrieval.

Zusätzlich sind in PIRCS *soft boolean* Anfragen implementiert. Dabei wird eine boolesche Formel in Gewichte umgesetzt, die das Netz durch zusätzliche Neuronen auf der Seite der Anfragen realisiert. Dies ist die einzige bekannte Umsetzung von Booleschen Retrieval in ein neuronales Netz.

4.3.2.2 Adaptive Information Retrieval (AIR)

Belew 1986 und 1989 stellen das System Adaptive Information Retrieval (AIR) vor, das in drei Schichten Terme, Dokumente und Autoren repräsentiert. Wie im Grundmodell der Spreading-Activation-Systeme für Information Retrieval (cf. Abschnitt 4.3.1) vorgesehen, breitet sich die Aktivierung in AIR in mehreren Phasen aus.

Für die Initialisierung der Gewichte zwischen Dokumenten und Termen werden typische IR-Indexierungstechniken verwendet. Belew 1989 verwendet ein Gewichtungsschema auf Basis der inversen Dokument-Frequenz der Terme. Die Verbindungen sind bidirektional und symmetrisch. Zusätzlich sind Autoren mit ihren Dokumenten verbunden. Die Summe aller Gewichte im System bleibt konstant. Beim Hinzufügen neuer Dokumente zur Laufzeit berücksichtigt dies der Gewichtungsalgorithmus.

Die Anfrage kann aus Elementen verschiedenen Typs bestehen und potenziell Terme, Dokumente und Autoren enthalten. Ebenso liefert das Ergebnis verschiedene Typen von Elementen. Auch Relevanz-Feedback kann sich auf alle Knoten beziehen. Damit nutzt AIR die Flexibilität des Spreading-Activation-Ansatzes in dieser Hinsicht voll aus.

Lernen durch Relevanz-Feedback verändert die Gewichte der Verbindungen und beeinflusst dadurch das Modell. Das Feedback kann vier Werte annehmen, darunter auch negative: *very relevant, relevant, irrelevant, very irrelevant*. Die vom Benutzer vorgegebene Feedback-Information läuft im Netz entlang der Verbindungen. Die Verbindungsgewichte werden nach einer der Delta-Regel ähnlichen Lernregel modifiziert (cf. Abschnitt 3.4.3). Die Aktivierungsstärke des Ausgangsneurons und das Relevanz-Feedback-Signal sind die entscheidenden Faktoren.

AIR enthält auch Verbindungen innerhalb von Schichten. Diese entstehen durch Lernen. Belew 1986 berichtet vom Entstehen bemerkenswerter Assoziationen innerhalb der Schichten. So wurden in einem Fall Wörter mit gleichem Stamm sehr stark assoziiert und ein irrtümlich falsch geschriebener Autor hatte einen starken Bezug zu dem Neuron mit der richtigen Schreibweise (Belew 1986:186).

Die Tests mit AIR beruhen auf einem Korpus von 1500 bibliografischen Angaben. Für die Indexierung stehen nur die Titel zur Verfügung, so dass eine

Generalisierung der gewonnenen positiven Ergebnisse nicht möglich scheint. Das Netz für 1500 bibliographische Angaben besitzt insgesamt 5000 Neuronen, so dass 10 Millionen Verbindungen möglich sind. Durch die Initialisierung werden nur 0,2% davon besetzt (cf. Belew 1986). Die Entscheidung, ob die Aktivierung eines Anfrage-Terms konstant bleibt, oder vom Netz festgelegt wird, löst AIR mit einem Kompromiss. Die Anfrage-Terme werden zwischen ein und zehn Epochen konstant gehalten. Dies führte zu den besten Ergebnissen. Bei der Anfrage waren auch negative Terme erlaubt, die dann als negative Aktivierung ins Netz eingingen.

Auch das Lernen durch Relevanz-Feedback führte bei Belew 1986 zu Verbesserungen. Allerdings ergaben sich negative Folgen, die wohl auf die geringe Anzahl von Dokumenten zurückzuführen ist. Innerhalb des Netzes entstand ein untereinander stark assoziiertes Cluster von Dokumenten, die häufig eine negative Beurteilung erhielten. Sobald eines davon im Verlauf der Spreading-Activation aktiv wurde, aktivierte es das gesamte Cluster. Trotz einiger weiterer kleiner Schwächen der Lernregel glaubt Belew 1986, dass das System insgesamt überzeugt und sich noch optimieren lässt.

AIR verfügt über eine Benutzungsoberfläche, die den Fluss von Aktivierung und die beteiligten Verbindungen und Neuronen transparent macht. Die Oberfläche von Belew benutzt sehr kleine Knoten und die Beschriftung verläuft teilweise schräg. Damit wirkt die Benutzungsoberfläche eher verwirrend und wenig benutzungsfreundlich. Allerdings ist die Idee, den Spreading-Activation-Prozess zu visualisieren, durchaus interessant und verfolgenswert. Dies gilt insbesondere, da Information Retrieval Systeme in der Regel kaum transparent sind und für den Benutzer eine *black box* darstellen.

Rose/Belew 1991 bauen AIR durch die Integration semantischen Wissens zu dem hybriden System SCALIR aus, das in Abschnitt 4.3.2.7 besprochen wird.

4.3.2.3 SYRENE

Mothe 1994 präsentiert mit SYRENE ein weiteres zweischichtiges Netz. Die Schichten repräsentieren Terme und Dokumente. Die Verbindungen zwischen den Termen und Dokumenten werden durch die Indexierung initialisiert und mit der inversen Dokument Frequenz (cf. Abschnitt 2.1) gewichtet. Daneben entstehen bei der Initialisierung bereits Verbindungen innerhalb der Term-Schicht. Diese initialisiert SYRENE aufgrund von Kookkurrenzen oder semantischem Wissen. Mothe 1994 nennt vier Arten von Verbindungen:

- Aufgrund von Thesuarusbeziehungen manuell vergebene Verbindungen:
  - Synonyme
  - Spezifische Begriffe
  - Allgemeine Begriffe
- Automatisch erstellte Verbindungen:
  - Kookkurrenz-Analyse

Die Werte der Kookkurrenz-Verbindungen ergeben sich wie im Standard-Modell für Spreading-Activation-Netzwerke aus den Gewichtungen der Terme für die berechnet. Die semantischen Beziehungen werden ebenfalls in numerische Werte umgesetzt. Jede Klasse von Beziehungen erhält heuristisch einen Wert, z.B. w_{sy} für Verbindungen zwischen Synonymen. Alle Verbindungen von dieser Klasse besitzen diesen gesetzt. Die semantischen Beziehungen führen so zu zusätzlichen Verbindungen mit numerischen Gewichten, die das Netzwerk wie normale Verbindungen behandelt. Das Netz verarbeitet somit Gewichtungsinformationen und zusätzliche semantische Informationen einheitlich.

Bei der Aktivierungsausbreitung ist denkbar, dass der Benutzer nur bestimmte semantische Verbindungsarten auswählt und die übrigen ausblendet.

Mothe 1994 glaubt, dass die entstehenden assoziativen Beziehungen in verschiedenen Kollektionen und für verschiedene Benutzer unterschiedlich sind, dass jedoch ein Kern der semantischen und statistischen Links zwischen den Termen über mehrere Text-Kollektionen hinweg konstant bleibt. Diese könnten einen weithin gültigen Bestand assoziativer Beziehungen bilden, der ohne weitere intellektuelle Überprüfung auf weitere Text-Bestände übertragen wird. Die Beschreibung der Realisierung zeigt aber nicht deutlich, wie dieser Kern identifiziert werden soll. Sollte dieses Verfahren in Tests die Qualität des Retrievals verbessern, dann ist die Identifizierung eines Kerns zur Übertragung auf neue Datenbestände nur für eine Anfangsphase sinnvoll. Das Ausnutzen der spezifischen Eigenschaften eines Korpus lässt eine höhere Qualität erwarten. So erreichen verschiedene Information Retrieval Systemen bei verschiedenen Korpora oft unterschiedliche Qualität.

Die Aktivierungsausbreitung erfolgt zunächst innerhalb der Term-Schicht und setzt sich nach eventuell mehreren Phasen in Richtung der Dokument-Schicht fort. Von den Dokumenten fließt nur beim Relevanz-Feedback Aktivierung in die Term-Schicht. In diesem Fall bewertet der Benutzer die stark aktivierten Dokumente. Deren Aktivierung wird dann je nach Grad des Benutzerurteils verändert. Die Aktivierung fließt wiederum in die Term-Schicht und die nun aktivierten Terme senden ihre Signale erneut in die Dokument-Schicht. Auch bei Relevanz-Feedback sind damit maximal drei Schritte oder Phasen der Ak-

tivierungsausbreitung erlaubt. Lernen in Form von Gewichtungsänderungen findet durch die Relevanz-Feedback-Informationen nicht statt. Die Informationen des Benutzers über Relevanz der vorgelegten Dokumente besteht nur flüchtig und geht für spätere Anfragen verloren.

Aufgrund der restriktiven Ausbreitung von Aktivierung ist es sinnvoll, die Kookkurrenzen von Termen zu berechnen und im Netz zu verankern, da eine indirekte Aktivierung wie im Standard-Modell für Spreading-Activation-Netzwerke (cf. Abschnitt 4.3.1.3) nicht möglich ist. Die Stärken des Spreading-Activation-Ansatzes wie die inhärente Term-Expansion und die Lernfähigkeit neuronaler Netze schöpft SYRENE nicht aus. Mothe 1994 testet SYRENE mit einer kleinen Kollektion von 67 Dokumenten und zwölf Anfragen. Damit optimiert sie einige Parameter des Systems. Aufgrund des geringen Umfangs dieser Testmenge lassen sich die Ergebnisse aber nicht auf andere Daten übertragen.

Interessant ist der Ansatz von Mothe 1994 vor allem deshalb, weil die Autorin die Beziehung zwischen Spreading-Activation-Netzwerken und Standard IR-Modellen formal untersucht und ihre Ergebnisse empirisch belegt. Zunächst beweist die Autorin, dass Spreading-Activation nach Initialisierung und einem einzigen Aktivierungsschritt bei entsprechender Wahl der Aktivierungsfunktion zu den gleichen Ergebnissen führt wie ein Vektorraum-Modell mit äquivalenter Ähnlichkeitsfunktion. Mothe 1994 stellt drei Ähnlichkeitsfunktionen des Vektorraum-Modells den entsprechenden Aktivierungsfunktionen der Spreading-Activation-Modelle gegenüber:

	Ähnlichkeitsfunktion im Vektorraum-Modell	Aktivierungsfunktion im Spreading-Activation-Netzwerk
Inneres Produkt	$\vec{D}_j\vec{Q} = \sum_i d_{ij} q_i$	$Akt_j^D(t=1) = \sum_i Akt_i(t=0)\, TW(i)\, w_{ij}$
Kosinus-Maß	$Cos(\vec{D}_j\vec{Q}) = \frac{\sum_i d_{ij} q_i}{\sqrt{\sum_i q_i^2}\sqrt{\sum_i d_{ij}^2}}$	$Akt_j^D(t=1) = \frac{\sum_i Akt_i(t=0)\, TW(i)\, w_{ij}}{\sqrt{Akt_i(t=0)^2}\sqrt{(TW(i)\, w_{ij})^2}}$
Jaccard-Maß	$Jac(\vec{D}_j\vec{Q}) = \frac{\sum_i d_{ji} q_i}{\sum_i q_i^2 + \sum_i d_{ji}^2 - \sum_i q_i d_{ji}}$	$Akt_j^D(t=1) = \frac{S}{\sum_i (Akt_i(t=0))^2 + \sum_i (TW(i)\, w_{ij})^2 - S}$ $S = \sum_i Akt_i(t=0)\, TW(i)\, w_{ij}$

Mothe 1994:283

Mothe 1994 demonstriert die enge Beziehung zwischen den etablierten Information Retrieval Modellen und dem Spreading-Activation-Ansatz sehr eingängig. Nach einem Aktivierungsschritt sind die beiden Modelle äquivalent und führen so auch zu identischer Retrieval-Qualität. Damit demonstriert Mothe 1994 auch, dass im Spreading-Activation-Ansatz kein völlig neuartiges Modell darstellt.

4.3.2.4 Mercure

Mercure (cf. Boughanem et al. 1999, Boughanem/Soule-Dupuy 1994/1997) ist ein weiteres Spreading-Activation-Netzwerk im Rahmen der TREC Evaluierungsstudie getestet wurde (cf. Boughanem/Soule-Dupuy 1997/1998). Abschnitt 4.8 vergleicht die Ergebnisse mit anderen Systemen.

Mercure verfügt über eine Input- und Output-Schicht, die jedoch nur als Interface dienen und keine weitere entscheidende Bedeutung haben. Von der Anfrage-Schicht aus werden die Term-Neuronen nach einem Gewichtungs-Schema aktiviert. Im Kern enthält Mercure Verbindungen zwischen einer Term-Schicht und einer Dokument-Schicht und innerhalb der Term-Schicht. Die Gewichte werden aufgrund von Vorkommenshäufigkeiten und Kookkurrenzen initialisiert. Nachdem Boughanem/Soule-Dupuy 1994 noch ein einfacheres Gewichtungsschema einsetzen, berechnen sich die Gewichte in Boughanem/Soule-Dupuy 1997 aus folgender Formel:

$$w_{ij} = \frac{freq_{ij}}{\sqrt{\sum_{k=1}^{T}(freq_{ik}^2 \log(\frac{M}{m_k})^2)}} \log\frac{M}{m_i}$$

w_{ki} *Gewicht der Verbindung von Term i zu Dokument j*

$freq_{ij}$ *Frequenz von Term i in Dokument j*

M ,T Anzahl Doumente und Anzahl Terme in der Kollektion

m_i *Anzahl Doumente mit Term i in der Kollektion*

Boughanem/Soule-Dupuy 1997:2f.

Unter Einbeziehung der Dokumentlänge entwickeln Boughanem/Soule-Dupuy 1998 ein elaboriertes Gewichtungsschema, das an das Okapi-Verfahren angelehnt ist (cf. Robertson et al. 1997). Okapi ist ein Information Retrieval System, das mehrfach mit Erfolg an TREC teilgenommen hat und auf dem Vektorraum-Modell beruht.

$$w_{ij} = \frac{\frac{(1+\log(freq_{ij}))}{1+\log(average_j(freq_{ij}))}(h_1 + h_2 \log(\frac{N}{n_i}))}{h_3 + h_4 \frac{doc\_length}{average\_doc\_length}}$$

w_{ij} *Gewicht der Verbindung von Term i zu Dokument j*

$freq_{ij}$ *Frequenz von Term i in Dokument j*

N *Anzahl Doumente in der Kollektion*

n_i *Anzahl Doumente mit Term i in der Kollektion*

h_x *verschiedene Parameter*

Boughanem/Soule-Dupuy 1998:3

Mit dieser Formel erzielte Mercure bessere Ergebnisse. Analog veränderten Boughanem/Soule-Dupuy 1998 auch die Gewichtung der Anfrage-Terme.

Die Verbindungen innerhalb der Term-Schicht ergeben sich aus folgender Gleichung:

$$c_{ij} = a \frac{\sum_{k=1}^{T}(w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^{T} w_{ik}^2 + \sum_{k=1}^{T} w_{jk}^2 - \sum_{k=1}^{T}(w_{ik} w_{jk})}}$$

w_{ij} *Gewicht der Verbindung von Term i zu Dokument j*

T *Anzahl der Terme in der Kollektion*

a *Parameter*

Boughanem/Soule-Dupuy 1997:2f.

In Boughanem/Soule-Dupuy 1994 lernt Mercure aus Relevanz-Feedback und verändert die Verbindungen innerhalb der Term-Schicht. Die Lernregel ist an die Delta-Regel angelehnt und berücksichtigt die Anzahl der als relevant und nicht relevant eingestuften Dokumente:

$$\Delta w_{ij} = a\ d\ Aktivierung_i\ Aktivierung_j$$

a *Lernparameter*

d *Parameter, abh. von der Anzahl der relevanten Dokumente*

Boughanem/Soule-Dupuy 1994:525

Das Lernverfahren wurde mit einer kleinen Menge von 350 bibliographischen Angaben getestet. Die Ergebnisse dürfen bei dieser kleinen Grundlage nur als erster Hinweis gewertet werden. Die bei TREC eingesetzten Varianten von Mercure beinhalten kein Lernen. Dort dient Relevanz-Feedback lediglich für eine Term-Expansion. Die Aktivierungsausbreitung ist dabei stark begrenzt und umfasst pro Anfrage nur drei Schritte, so dass die Aktivierung nur einmal aus der Dokument-Schicht in die Term-Schicht zurück fließt. Dies wird als Backpropagation bezeichnet, was missverständlich ist, da sich dieses Verfahren stark vom Backpropagation-Algorithmus unterscheidet (cf. Abschnitt 3.5.4). Mercure erreichte in TREC 6 in zwei Kategorien des Standard Retrieval einen Platz unter den besten acht Systemen (cf. Abschnitt 4.8).

4.3.2.5 Weitere nicht lernende Spreading-Activation-Netze

Ein frühes Beispiel für ein Spreading-Activation-Netz bieten Salton/Buckley 1988. Ihr System orientiert sich stark an semantischen Netzen. Semantische Netze sind ein Wissensrepräsentations- und -verarbeitungsmechanismus, in dem eine Netzstruktur die Beziehungen zwischen Objekten wiederspiegelt. Die Verbindungen haben semantische Labels und erlauben Ableitungsprozesse. Gewichte wie in neuronalen Netzen sind für die Verbindungen nicht

vorgesehen (cf. Rich/Knight 1991:251 ff.). Ein Überblick über Spreading-Activation-Netzwerke mit Elementen semantischer Netze bietet Abschnitt 4.3.2.7.

Da in einem IR-System normalerweise wenig semantische Beziehungen aus dem Anwendungsfall bekannt sind und maschinenlesbar zur Verfügung stehen, greifen auch Salton/Buckley 1988 nach einer Diskussion der Chancen semantischer Netze im IR auf ein neuronales Spreading-Activation-Netz zurück, das im Wesentlichen dem Standard-Modell ohne Lernen entspricht (cf. Abschnitt 4.3). Die semantischen Beziehungen werden dabei durch die Gewichte aus der Indexierung ersetzt.

Eine Möglichkeit für semantische Beziehungen zwischen Dokumenten, die Salton/Buckley 1988 anführen, sind Zitate. Dies wären Verbindungen innerhalb von Schichten, die in diesem Modell auch vorgesehen sind. Die Stärke der Spreading-Activation-Modelle besteht jedoch gerade darin, dass solche direkten Verbindungen aufgrund der indirekten, assoziativen Beziehungen nicht nötig sind. Ähnliche Dokumente sollten auch ohne die explizite Angabe von Synonymen gefunden werden, da ein Term sein Synonym indirekt über die Dokument-Schicht aktiviert. Ein Term aktiviert Dokumente, die dann wiederum Terme aktivieren. Da die Terme in den gleichen Dokumenten vorkommen, sind sie ähnlich und aktivieren wiederum ähnliche Dokumente. Die Netzwerkfunktionen sind so gewählt, dass die Aktivierungsausbreitung folgende Ähnlichkeitsfunktion implementiert:

$$similarity(D,Q) = A\sum_{t=1}^{m} \frac{w_{qt} w_{dt}}{(\sum_{k=1}^{m} w_{qk})(\sum_{j=1}^{n} w_{jt})}$$

Salton/Buckley 1988:151

Diese Formel weist darauf hin, dass nur ein Aktivierungsschritt vorgesehen ist und bestätigt die Ergebnisse von Mothe 1994, die auf die Äquivalenz von Vektorraum-Modell und Spreading-Activation-Netzwerken hinweist. Damit unterliegt das Modell den gleichen Schwächen wie die meisten der vorgestellten Systeme, es nutzt die Möglichkeit der Spreading-Activation-Netzwerke nicht voll aus.

Salton/Buckley 1988 vergleichen für sechs kleine IR Testkollektionen die Performanz ihres einfachen Spreading-Activation-Ansatz mit der eines Vektorraum-Modells. Dabei evaluieren sie verschiedene Verfahren für die Termgewichtung. Längennormalisierung verbessert die Resultate beider Verfahren. Das Vektorraum-Modell erreichte mit Berücksichtigung der inversen

Dokument-Frequenz und Normaliserung der Dokumentlänge die besten Resultate. Dieses Vektorraum-Modell setzte ein ausgefeilteres Gewichtungsschema ein als die Spreading-Activation-Netzwerke, die in keinem Fall die inverse Dokument-Frequenz berücksichtigen. Das Vektorraum-Modell ohne inverse Dokument-Frequenz schnitt dagegen sehr viel schlechter ab als das einfachste Spreading-Activation-Netz. Salton/Buckley 1988 zeigen somit, dass Spreading-Activation durchaus vergleichbare Ergebnisse erbringt.

Aus dem gleichen Jahr datiert der Ansatz von Cochet/Paget 1988, der eine Kollektion von Bildern, die durch intellektuell vergebene Terme repräsentiert wird, in einem Spreading-Activation-Ansatz modelliert. Cochet/Paget 1988 führen einige Besonderheiten ein, die bei anderen Netzen nicht zu finden sind. Die Aktivierungsfunktion berücksichtigt die Verbindungsstärken des gesamten Netzes und realisiert damit keine streng lokale Berechnung.

$$\Delta w_{ij} = \frac{\sum_{Akt(v_j)\neq 0} w_{ij}}{\sum_{J=1} w_{ij}} netinput$$

Cochet/Paget 1988:666

Die Summe aller bestehenden Gewichte wirkt sich negativ auf den Betrag der Gewichtsveränderung aus, während die Summe der aktiven Gewichte positiv wirkt. Letztere besitzen damit doppelten Einfluss, da sie im Netinput, der Summe der gewichteten Eingänge, bereits enthalten sind.

Weiterhin führen Cochet/Paget 1988 einen Mechanismus zur Veränderung der Topologie der Verbindungen ein, den sie Absorption nennen. Dieser dient dazu, die Zahl der Verbindungen zu begrenzen. Die Absorption analysiert die gemeinsamen Bilder von Termen. Bildet die Menge der Bild-Dokumente eines Terms eine Untermenge zu einem anderen Term, so entsteht eine neue Verbindung zwischen den Termen, welche die Verbindungen des ersten Terms zu seinen Bilder ersetzt. Eine starke Beziehung oder eine hohe Kookkurrenz führt zu einer Verbindung innerhalb der Schicht, wie dies auch bei manchen anderen Modellen der Fall ist. Zusätzlich werden aber einige der originalen Verbindungen getilgt.

Lernfähigkeit durch Relevanz-Feedback ist lediglich geplant, wobei Cochet/Paget 1988 zu Recht feststellen, dass keine Grenze zwischen Retrieval und Lernphase besteht, sondern beide gehen ineinander über und verlaufen parallel.

Boyd et al. 1994 setzen ein zweischichtiges Netz im TREC Kontext ein (cf. Abschnitt 4.8). Sie wollen es als Maßstab für ihre eigentlichen semantischen Experimente nutzen. Eine Schicht repräsentiert Terme und die zweite Topics. Da die Autoren damit ein Routing-Experiment durchführen, handelt es sich dabei um die Dokumenten-Schicht. Die eigentlichen Dokumente werden als Aktivierung an die Term-Schicht angelegt und aktivieren so die für sie relevanten Topics.

Wilkinson/Hingston 1991 und 1992 stellen ebenfalls ein Netz vor, das eine Term- und eine Dokument-Schicht benutzt. Die von den Autoren beschriebene Anfrage-Term-Schicht kann vernachlässigt werden, da sie nur die Aufgabe hat, Anfrage-Terme über Verbindungen mit einem festem Gewicht mit dem Wert Eins in die eigentliche Term-Schicht, die hier Dokument-Term-Schicht genannt wird, zu propagieren. Durch geschickte Wahl der Aktivierungsfunktion und Propagierungsregel implementiert das netz von Wilkinson/Hingston 1991 die Kosinus-Ähnlichkeitsfunktion. Nach einem Aktivierungsschritt ohne implizite Expansion erhalten sie auch empirisch die gleichen Ergebnisse wie in einem Test mit dem Vektorraum-Modell und der Kosinus-Ähnlichkeitsfunktion. Wilkinson/Hingston 1991 integrieren Relevanz-Feedback in ihr System. Die im ersten Ergebnis vom Benutzer als relevant erkannten Dokumente werden stark aktiviert und behalten diese Aktivierung für den weiteren Verlauf. Sie aktivieren wiederum bestimmte Terme und diese neue Dokumente. Relevanz-Feedback verändert nur Aktivierung, die Gewichte des Netzes bleiben konstant.

Wilkinson/Hingston 1991 und 1992 experimentieren mit traditionellen IR-Kollektionen, wie CACM (cf. Baeza-Yates/Ribeiro-Neto 1999:92f.) und Cranfield-Kollektion (cf. Abschnitt 7.1.1). Sie zeigen, dass das Netz nach dem ersten Aktivierungsschritt die gleiche Qualität erreicht wie der Kosinus. Messungen nach einzelnen Schritten zeigen, dass sich die durchschnittliche Precision schnell erhöht und dann langsam sinkt. Das beste Ergebnis erzielte das System nach nur zwei Aktivierungsschritten. Nach zwanzig Schritten war der durchschnittliche Recall nur leicht gesunken und lag immer noch über dem Vektorraum-Modell mit Kosinus-Ähnlichkeitsmaß. Auch die Ergebnisse mit Relevanz-Feedback verbesserten sich bei einer geringen Anzahl von Aktivierungsschritten. Wilkinson/Hingston 1991 zeigen für alle Test-Kollektionen, dass zumindest eine geringe Anzahl von Aktivierungsschritten positiv auf das Retrievalergebnis wirkt.

Das Modell von Mothe/Soule-Dupuy 1992 besteht ebenfalls aus zwei Schichten, eine Schicht repräsentiert die Dokumente und eine die Terme. Die Anfrage besteht aus einer Untermenge von Termen, die der Benutzer auswählt. Mothe/Soule-Dupuy 1992 lassen auch Verbindungen innerhalb der

Schichten zu. Relevanz-Feedback verändert auch in diesem Modell lediglich die Aktivierung des Netzes.

Crestani 1997a schlägt ein Spreading-Activation-Modell in einem Hypertext vor. Das Spreading-Activation-Netz besteht dabei aus den semantischen Beziehungen innerhalb des Hypertexts. Damit präsentiert Crestani 1997a ein hybrides Modell, gibt jedoch keinen Algorithmus für die Kombination beider Wissensarten an. Crestani 1997a betont, dass Spreading-Activation eingeschränkt werden muss, da ansonsten schnell das gesamte Netz hoch aktiviert ist. Die semantischen Beziehungen sieht er als eine Möglichkeit, die Aktivierung zu hemmen. Da semantisch motivierte Verbindungen aber zusätzliche Links sind, erhöhen sie eher die Gesamtaktivierung. Weitere Netze mit semantischen Elementen diskutiert Abschnitt 4.3.2.7.

Einen sehr interessanten Ansatz und fundierte experimentelle Ergebnisse bieten Syu et al. 1996. Sie kombinieren die Dimensionalitätsreduktion von Latent Semantic Indexing (LSI, cf. Abschnitt 2.1.2.4.3) mit dem Spreading-Activation-Netzwerk. LSI reduziert die Dokument-Term-Matrix auf zwischen zwanzig und 200 LSI-Terme. Dadurch ist die Term-Schicht bei Syu et al. 1996 wesentlich kleiner als in anderen Systemen. Vor der Komprimierung erstellen sie die Dokument-Term-Matrix mit folgender Gewichtungsfunktion:

$$w_{ij} = \frac{tf_{ij} idf_j}{\max\_tf_i \log(Anzahl\_Dokumente)}$$

Syu et al. 1996:148

Im Gegensatz zum Standard-Modell breitet sich die Aktivierung zunächst in der Term-Schicht aus, wo ein winners-take-all-Verfahren wirkt. Dadurch begrenzen die Autoren die Anzahl der aktivierten Terme. In der nächsten Phase läuft die Aktivierung mehrfach zur Dokument-Schicht und zurück. In der Dokument-Schicht wirkt ebenfalls ein winners-take-all-Verfahren. Zusätzlich hält eine Funktion die Aktivierung der Knoten im Bereich [0; 1]. Stopp-Kriterium ist wie in einem Hopfield-Netzwerk (cf. Abschnitt 3.5.3 und 4.2.1) ein stabiler Zustand (Equilibrium), bei dem weitere Aktivierungsschritte keine wesentliche Veränderung des Netzwerkzustands bewirken. Die aktivierten Dokumente haben dann Werte nahe Eins erreicht.

Im Gegensatz zu vielen anderen Systemen durchläuft die Aktivierung mehrfach die Verbindungen zwischen Termen und Dokumenten und nutzt so die Vorteile des Spreading-Activation-Modells besser aus. Die potenziellen Nachteile werden durch mehrere hemmende Faktoren kontrolliert (Constrained Spreading-Activation).

Für die Experimente greifen Syu et al. 1996 auf vier klassische Test-Kollektionen zurück. Sie vergleichen die Performanz ihres Systems sowohl zu einem Netzwerk ohne Komprimierung durch LSI als auch zu einem Vektorraum-Modell mit Kosinus-Ähnlichkeitsfunktion auf Basis der reduzierten Repräsentation. Das Spreading-Activation-Netzwerk mit LSI-Vektoren brachte die besten Resultate. Zusätzlich experimentieren sie für jede Kollektion mit verschiedenen Niveaus der Komprimierung zwischen zehn und 400 LSI-Dimensionen und finden so für jede Kollektion die optimale Komprimierung.

Die Ergebnisse deuten insgesamt darauf hin, dass Systeme mit Aktivierungsausbreitung in mehreren Schritten durchaus positive Ergebnisse erzielen und die Beschränkung vieler Modelle auf ein oder zwei Schritte nicht unbedingt erforderlich ist.

4.3.2.6 Weitere lernende Spreading-Activation-Netzwerke

Layaida/Caron 1994 benutzen in ihrem Modell ebenfalls eine Schicht für Autoren, die mit der Dokument-Schicht verbunden ist. Autoren werden mit ihren Dokumenten verknüpft und erhalten zudem Verbindungen für gemeinsame Publikationen. Auch in diesem Netz steht vor der Term-Schicht eine Anfrage-Schicht, die nur die Aufgabe hat, die Terme in der Term-Schicht zu aktivieren. Wie in Abschnitt 4.3.2.1 diskutiert, sind solche Schichten nicht erforderlich. Die Initialisierung der Term-Dokument-Verbindungen erfolgt mit einer Form der inversen Dokument Frequenz. Die Terme sind untereinander verknüpft und die Gewichte der Verbindungen werden mit einem Assoziationsfaktor initialisiert.

Suchbedingung in Anfragen können sowohl Terme als auch Autoren sein. Die Autoren-Schicht wird auch bei Anfragen benutzt, die ausschließlich aus Termen bestehen. Das System lernt langfristig aus Relevanz-Feedback-Information und verändert die Verbindungsgewichte innerhalb der Term-Schicht und zwischen Term- und Dokument-Schicht. Die Lernregel ähnelt der Delta-Regel des Backpropagation-Algorithmus, führt aber nicht zu subsymbolischen Repräsentationen, da keine versteckte Schicht beteiligt ist. Innerhalb der Autoren-Schicht modifiziert eine Form Hebb'schen Lernens (cf. Abschnitt 3.4.3) die Verbindungsstärken. Layaida/Caron 1994 berichten von Experimenten mit 250 Dokumenten und 25 Anfragen. Das Netz ähnelt dem System AIR (cf. Belew 1989, cf. Abschnitt 4.3.2.2).

Wong et al. 1993 konstruieren ein Netz, dessen Architektur sich vom Standard-Modell unterscheidet. Darin gibt es zwei Term-Schichten, eine für Dokument-Terme und eine für Anfrage-Terme. Die Verbindungsmatrix zwischen diesen beiden Schichten ist somit eine Assoziationsmatrix, die

Termerweiterungen realisiert. Grundsätzlich können in diesem Modell das Anfrage- und Dokument-Vokabular unterschiedlich sein, so dass auch heterogene Datenbestände erschließbar sind. Zusätzlich fasst eine Output-Schicht mit nur einem Neuron die Aktivierung der Anfrage-Term-Schicht zusammen, wie Abbildung 4-11 zeigt.

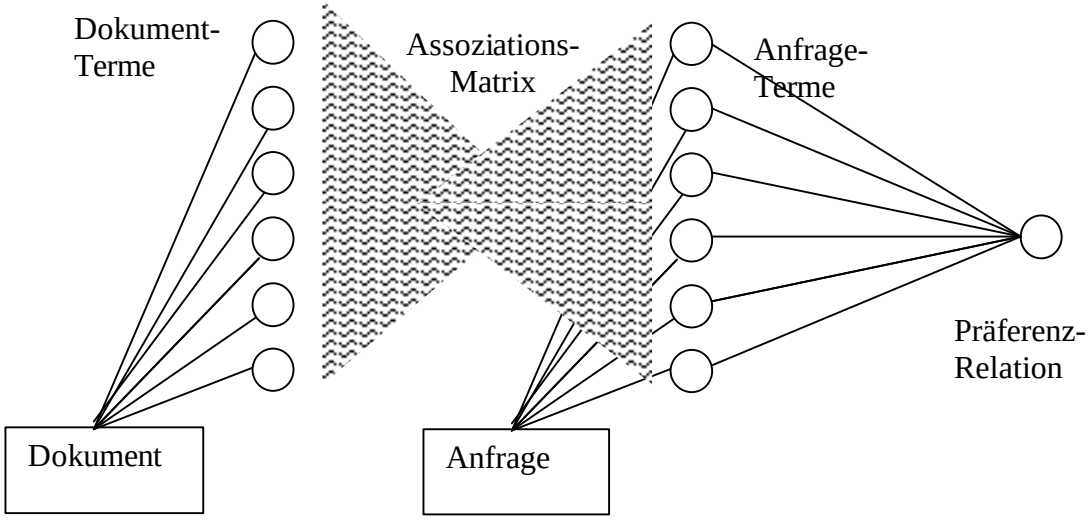


Abbildung 4-11: Struktur des Modells von Wong et al. 1993

An beiden Schichten werden die in Dokument und Anfrage vorkommenden Terme aktiviert. Dies geschieht durch eigene Schichten für Dokumente und Anfragen, die aber weiter keine Rolle spielen. An der Dokument-Schicht wird der Differenz-Vektor zweier Dokumente angelegt, d.h. das System vergleicht bei jedem Schritt zwei Dokumente in Bezug auf eine Anfrage. Die Aktivierung läuft durch das Netz und eine Output-Schicht mit nur einer Unit, die mit der Anfrage-Term-Schicht voll vernetzt ist, sammelt ein Fehlersignal. Es misst, ob die Matrix die Präferenz des richtigen Dokuments bei dieser Anfrage erkennt. Bei einem Output größer Null hat das System ein korrektes Ranking ermittelt, bei negativem Output wird die Matrix adaptiert. Bei linearer Aktivierungsfunktion und dem Verzicht auf Gewichtungsparameter bei der Spreading-Activation ergibt sich folgender Output:

$$g(d,q) = \sum_j \sum_i d_i a_{ij} q_j = dAq$$

A *Assoziationsmatrix*

d *Dokumentvektor* (*Differenzvektor*)

q *Anfragevektor*

Wong et al. 1993:109

Damit ist der Output ein einfaches Produkt, das sich auch ohne neuronales Netz implementieren ließe.

Die Assoziationsmatrix wird durch gradient-descent-Lernen also durch Minimierung eines Fehlers mit Relevanz-Feedback optimiert. Ausgangspunkt ist die vereinfachende Annahme, dass die Rankingfunktion sich zu den Dokumenteigenschaften linear verhält. Die Komplexität dieser Abbildung ist jedoch nicht bekannt, so dass diese Annahme problematisch ist. Mit dieser Vereinfachung beweisen Wong et al. 1993, dass ihr Lernverfahren eine Lösung in endlich vielen Schritten erreicht, falls eine Lösung existiert.

Wong et al. 1993 experimentieren mit einer Testkollektion, die aus 82 Dokumenten und 35 Anfragen besteht, von denen 31 für das Training und nur vier für den Test benutzt wurden. Die Kollektion enthält 1217 Terme, wobei nur 200 in allen Anfragen vorkommen. Gegenüber einer Ähnlichkeitsberechnung mit dem Kosinus-Maß verbessert sich der durchschnittliche Recall um über 60%.

Interessant an diesem Modell ist der Einsatz zweier Term-Schichten. Dadurch wird das Indexierungsvokabular transparent auf das Anfragevokabular abgebildet. Des weiteren besitzt das System zwei Eigenschaften, in denen es sich vom Standard-Spreading-Activation-Modellen unterscheidet und die auch das COSIMIR-Modell auszeichnen (cf. Kapitel 6).

- Es gibt ein einziges Output-Neuron, dessen Aktivierung das Ranking steuert.
- Der Retrievalprozess wird in Dokument-Anfrage-Paare zerlegt, die das Netz einzeln verarbeitet.

4.3.2.7 Semantische Spreading-Activation-Netzwerke

Hybride Systeme verwenden mindestens zwei verschiedene Arten von Wissensmodellierung. Typischerweise ist eine davon sub-symbolisch und eine symbolisch. Das Ziel hybrider Systeme ist es, die Nachteile einer Technik durch die Vorteile einer anderen auszugleichen. Damit bilden sie eines der

interessantesten Themen sowohl in der Kognitionswissenschaft als auch in der praktischen Anwendung, wobei gerade bei hybriden Systemen heuristische Faktoren eine große Rolle spielen.

Ein Beispiel für hybride Systeme sind konnektionistische Expertensysteme. Dies sind Systeme in denen sowohl neuronale Netze als auch konventionelle Expertensysteme aus der Künstlichen Intelligenz, die aufgrund von Regel Inferenzen bilden, eine Rolle spielen. Einen Überblick über die Koppelung von neuronalen Netzen mit Fuzzy Logik bieten Nauck et al. 1994. Auch die semantischen Spreading-Activation-Netzwerke können als hybride Systeme betrachtet werden.

Semantische Netze sind keine neuronalen Spreading-Activation-Netze im Sinne der in Kapitel 3 vorgestellten Modelle. Semantische Ansätze benutzen inhaltliche Verbindungen wie *Synonym* oder *Is-a*, während Verbindungen in neuronalen Netzen nicht inhaltlich belegt sind und ausschließlich numerische Gewichtung besitzen. Gemeinsam ist den Systemen das Prinzip der sich ausbreitenden Aktivierung. Automatische semantische Verfahren haben sich im Information Retrieval bisher nicht durchgesetzt, da sie für Massendaten nicht geeignet sind. Manuelles Setzen der Verbindungen ist für einen realistischen großes Korpus kaum möglich. Trotzdem greifen einige wenige Systeme auf sicheres semantisches Wissen zurück oder kombinieren Elemente semantischer Netze mit neuronalen Spreading-Activation-Netzen. Einen Überblick über semantische Spreading-Activation-Netze im Information Retrieval bietet Crestani 1997.

Ein erster und durchaus häufig vorkommender Schritt in die Richtung semantischer Netze ist die Integration semantischer Beziehungen in neuronale Verbindungen. Semantische Verbindungen werden nach Heuristiken in numerische Beziehungen umgewandelt wie etwa in SYRENE (cf. Abschnitt 4.3.2.3). Die Integration einer Autoren-Schicht ermöglicht die Realisierung einiger sicherer semantischer Beziehungen. Die Beziehung zwischen einem Autor und seinem Dokument ist eindeutig, die Beziehung zwischen Term und Dokument dagegen nur vage. Eine Autoren-Schicht enthalten die Systeme von Belew 1989 (AIR, cf. Abschnitt 4.3.2.2) und Layaida/Caron 1994 (cf. Abschnitt 4.3.2.6).

Neben Autorenschaft ergeben sich weitere Ansatzpunkte für die Integration semantischen Wissens, die auch in vielen kommerziellen Literaturdatenbanken erfasst sind, wie etwa Zitationsbeziehungen oder Ko-Autorenschaft. Die Arten von Verbindungen veranschaulicht ein Überblick im Abschnitt 4.3.3.

Das System von Layaida/Caron 1994 (cf. Abschnitt 4.3.2.6) ist ein Beispiel für die Integration von Ko-Autorenschaft. Neben den üblichen Schichten für Dokumente, Terme und Anfragen beinhaltet ihr Spreading-Activation-Modell

eine Schicht für Autoren. Autoren, die gemeinsam publizieren, erhalten Verbindungen, die das Lernen nicht beeinflusst. Zwischen Autoren und ihren Dokumenten fügen Layaida/Caron 1994 Verbindungen mit der Stärke eins (= maximal) ein. Diese Links haben eine klare Semantik und bilden den ersten Schritt in Richtung hybride Systeme.

Mothe 1994 (cf. auch Abschnitt 4.3.2.3) verfolgt einen ähnlichen Ansatz. In dem zweischichtigen Spreading-Activation-Netzwerk gibt es in der Term-Schicht Verbindungen zwischen den Termen. Diese werden entweder durch Kookkurrenzen oder durch semantisches Wissen initialisiert. Als mögliche Instanzen von solchen intellektuell vergebenen Verbindungen nennt Mothe 1994 Synonyme, spezifischere und generellere Begriffe. Mothe 1994 glaubt aber, dass viele der semantischen und statistischen Links über mehrere Datenbanken hinweg konstant bleiben.

In bestimmten Anwendungsfällen liegen semantische Beziehungen zwischen den beteiligten Klassen von Objekten vor und sollten dann ausgenutzt werden. Ein System, das semantische und numerische Verbindungen kombiniert, ist SCALIR von Rose/Belew 1991. SCALIR (Symbolic and Connectionist Approach to Legal Information Retrieval) integriert konnektionistische und symbolische Elemente und ist ein vielversprechender hybrider Ansatz.

Rose/Belew 1991 gehen von der Beobachtung aus, dass Experten in juristischen Fragen zwar die üblichen Probleme haben, ihr intuitives Wissen in Regeln zu formulieren. Daneben existieren eindeutige und sichere Sachverhalte wie etwa die Beziehungen zwischen verschiedenen Instanzen eines Rechtsstreits oder der Bezug einer Urteilsbegründung zu einem bestimmten Gesetz. SCALIR basiert auf früheren Arbeiten von Belew im Rahmen des Spreading-Activation-Ansatzes und der Implementierung seines Systems AIR (cf. Abschnitt 4.3.2.2). Das Netzwerk besteht aus Schichten für Gesetzestexte, Urteile und Terme. Zwischen den einzelnen Neuronen können verschiedene Verbindungen bestehen.

S-Links (symbolic) bilden die symbolische Komponente. Sie verfügen über ein Label wie z.B. *widerrufen* oder *Verweis*, sind fest zugewiesen nur innerhalb von Schichten zugelassen. Die C-Links (connectionist) werden wie bei Kwok 1989 mit Termhäufigkeiten initialisiert und können durch Relevanz-Feedback verändert werden. Sie bestehen von Anfang an zwischen Termen und Urteilen und zwischen Termen und Gesetzen. Während des Lernens können sie an allen Stellen entstehen. Durch sie soll das Netz im laufenden Betrieb die komplexen und subtilen Zusammenhänge lernen, welche die intuitive Komponente des informationellen Prozesses bilden.

SCALIR verfügt weiterhin über sogenannte H-Links (hybrid), die nur zwischen Neuronen für Urteile vorkommen. Diese H-Links verändern ihr

Gewicht durch Lernen, besitzen aber eine feste semantische Bedeutung. Die Bedeutungen stammen aus einer Liste, die die Autoren zwei Achsen zuordnen. Die Position auf einer Achse spiegelt die Ähnlichkeit der Fälle wider. Sie reicht von *follows* (höchste Ähnlichkeit) bis zu *overruled* (niedrigste Ähnlichkeit).

Die Neuronen in SCALIR besitzen einen zweistelligen Aktivierungsvektor mit Komponenten für C- und S-Aktivierung, die entlang den entsprechenden Verbindungen läuft. Während die S-Links nur die Aktivierungskomponente der mit ihnen verknüpften Beziehung weiterleiten, leiten C-Links das gesamte Spektrum der Aktivierung weiter. Da die C-Links also auch semantische Aktivierung weiterleiten, besteht SCALIR nicht nur aus zwei parallelen Netzen, die unabhängig voneinander Information verarbeiten und kann zu Recht als hybrides System bezeichnet werden.

SCALIR erhält das semantische Wissen aus bestehenden Wissensbasen wie etwa einem Verzeichnis von Gerichtsurteilen und ihren Beziehungen. Dann lernt es anhand von Relevanz-Feedback nach einem heuristischen Algorithmus, den Rose/Belew 1991 als *localized reinforcment* bezeichnen. Verbindungen, die viel zu einem positiven Resultat beigetragen haben, werden verstärkt und umgekehrt. Ähnliche Verfahren setzen manche der lernenden Spreading-Activation-Ansätze ein (cf. Abschnitt 4.3.2). Sub-symbolische Informationsverarbeitung ist auch in SCALIR nicht realisiert.

SCALIR verfügt über eine grafische Benutzungsoberfläche, in der Einschränkungen des Dokumenttyps auf z.B. Urteile möglich sind. Die nach der Anfrage am stärksten aktivierten Knoten werden in ihrem Beziehungsgeflecht visualisiert. SCALIR ist ein interessanter Ansatz, der das Spreading-Activation-Modell erweitert und heuristische, aus dem Anwendungsfall gewonnene Strukturen integriert. Das System ist für den Anwendungsfall Gerichtsurteile optimiert. Damit ist die Generalisierbarkeit allerdings sehr eingeschränkt. Insbesondere liegt in vielen anderen Domänen nicht so gut strukturiertes und erschlossenes Wissen vor.

Die beiden im Folgenden vorgestellten Systeme basieren auf Wissen, das in noch stäkerem Maße strukturiert ist.

Cohen/Kjeldsen 1987 präsentieren das System GRANT, das Forscher bei der Suche nach geeigneten Förderprogrammen unterstützen soll. GRANT besteht aus einem Netzwerk der Forschungslandschaft, mit semantischen Verbindungen wie *component of*, *causes* oder *effected by*. Der Benutzer aktiviert für ihn interessante Forschungsthemen und die Aktivierung breitet sich entlang der Verbindungen aus. Die Ausbreitung wird von drei Mechanismen beschränkt:

- *Distance-constraint*: Die Aktivierung darf maximal über vier Knoten laufen.
- *Fan-out-constraint*: Knoten für sehr generelle Konzepte wie *Person* leiten die Aktivierung nicht weiter.
- Sicherheitswerte für die Regeln: Die Aktivierungsausbreitung in einem semantischen Netz gleicht der Anwendung von Regeln in einem Produktionssystem. Die Inferenz, ob eine Fördereinrichtung ein Thema fördert, basiert auf zwei Prämissen. Die Fördereinrichtung fördert ein weiteres Thema, das zu dem gesuchten in Beziehung steht. Je nach Beziehung und Art der Themen erhält jede Regel einen Sicherheitswert, so dass sich bei mehreren Ergebnissen, aufgrund der Sicherheitswerte ein Ranking entsteht. Die Regeln entsprechen den Verbindungen des Netzes, der Sicherheitswert erinnert somit an die Verbindungsgewichte.

Cohen/Kjeldsen 1987 bewerten GRANT mit den im Information Retrieval üblichen Qualitätsmaßen. Wie fast alle semantischen Ansätze hat auch GRANT den Nachteil, dass das Wissen intellektuell erfasst werden muss und das Gesamtsystem nicht auf andere Domänen übertragbar ist.

Das bereits in Abschnitt 4.2 diskutierte System CRUCS (Brachman/McGuiness 1988) benutzt semantische Verbindungen innerhalb einer Boltzmann-Maschine. Das semantische Netz repräsentiert die Beziehungen zwischen logikbasierten Programmiersprachen, führt aber aufgrund der Implementierung als Boltzmann-Maschine vages Retrieval in Form eines *partial match* durch.

Die Kombination von zwei verschiedenen Wissenstypen führt auch zu Problemen. Bei den Spreading-Activation-Netzen kann das Nebeneinander unterschiedlicher Verbindungen zu Widersprüchen führen. Es sind Fälle denkbar, in denen objektives semantisches Wissen dem momentanen Nutzeranforderungen widerspricht. So kann beispielsweise die Aktivierung von weiteren Dokumenten eines Autors unerwünscht sein. Lernende Systeme müssen dann das semantische Wissen mit anderen Arten von Verbindungen ausgleichen. Damit wird der Lernprozess natürlich erschwert.

Ein generell einsetzbares Information Retrieval System sollte aus diesen Gründen und aufgrund des hohen intellektuellen Aufwands für die Definition semantischer Beziehungen ohne diese auskommen. Eine zusätzliche Berücksichtigung erscheint jedoch in gut strukturierten Anwendungsfällen sinnvoll.

Die bisherigen Überlegungen beziehen sich auf die Ausnutzung semantischer Beziehungen zwischen Dokumenten, Termen und Anfragen. Dieses Wissen kann im Retrieval-Prozess eine Rolle spielen und etwa zur Erhöhung der In-

teraktivität eingesetzt werden. Das Verfolgen von semantisch klar interpretierbaren Verbindungen zum Beispiel zu anderen Dokumenten eines Autors oder zu zitierten Dokumenten stellt einen Mehrwert für den Anwender eines Information Retrieval Systems dar, der sich auch leicht vermitteln lässt. Allerdings ergeben sich auch daraus einige Probleme. Zum einen verweisen Zitationen, falls sie überhaupt maschinell verarbeitbar vorliegen, nicht immer auf Dokumente innerhalb des Korpus. Eine etwas andere Situation ergibt sich bei der Betrachtung von Internet-Dokumenten. Hier bestehen Hypertext-Verbindungen, hinter denen semantisch sehr verschiedene Beziehungen stehen. Sie stehen prinzipiell vollständig maschinell lesbar zur Verfügung. Die zur Zeit populären Internet-Suchmaschinen nutzen die Verbindungen zumindest dahingehend, dass häufige Verweise die Relevanz einer Seite erhöhen (cf. Mönnich 1999). Bekavac 1999 stellt ein Modell vor, das die Einbettung einer Internet-Seite in die Struktur des Servers in den IR-Prozess einbezieht. Die Einbeziehung der Links stößt jedoch aufgrund der hohen Anzahl von Seiten an Grenzen.

4.3.3 Vergleich der Spreading-Activation-Netzwerke mit dem Vektorraum-Modell

Die Dokument-Term-Matrix aus dem Vektorraum-Modell für Information Retrieval wird beim Spreading-Activation-Netzwerk als Verbindungsmatrix übernommen. Damit wird sie zur Dokument-Neuron-Term-Neuron-Matrix. Eine vollständige Verbindungsmatrix für ein neuronales Netz sollte sich jedoch immer über alle Neuronen erstrecken, so dass sowohl Zeilen als auch Spalten für alle Neuronen vorhanden sind. Damit werden Verbindungen zwischen allen Paaren von Neuronen möglich. Die folgende schematische Übersicht zeigt diese Matrizen im Überblick:

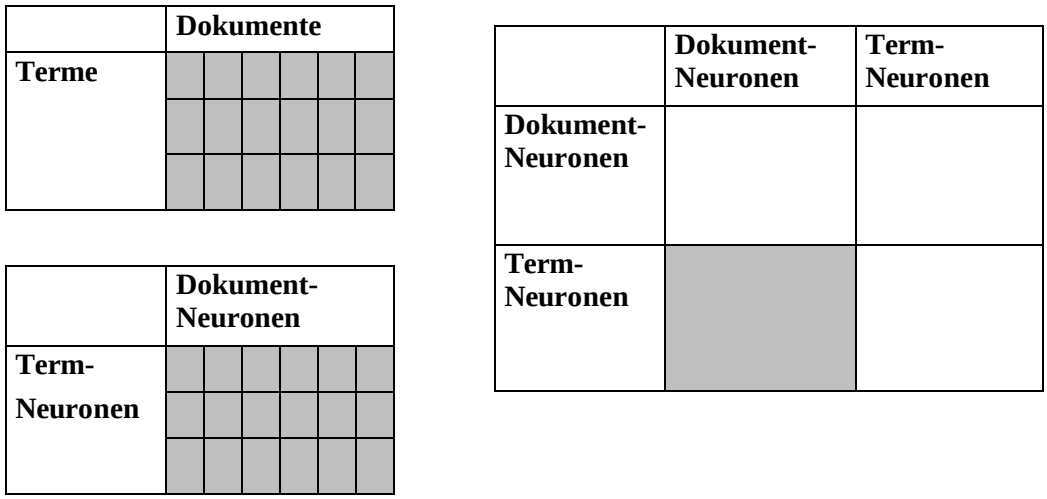


Abbildung 4-12: Schematische Gegenüberstellung von Dokument-Term-Matrix und vollständiger Verbindungsmatrix, bei der die entsprechenden Verbindungen grau unterlegt sind.

Neben den Verbindungen zwischen Termen und Dokumenten können damit auch Verbindungen innerhalb von Schichten definiert werden. Die von einigen Modellen vorgenommene Erweiterung des Standard-Modells für Spreading-Activation-Systeme um Verbindungen innerhalb von Schichten lässt sich demnach als Erweiterung der Matrix interpretieren. Auch die Frage nach der Symmetrie der Verbindungen stellt sich bei Betrachtung der Matrix erneut. Die folgende Abbildung fasst die Möglichkeiten zusammen:

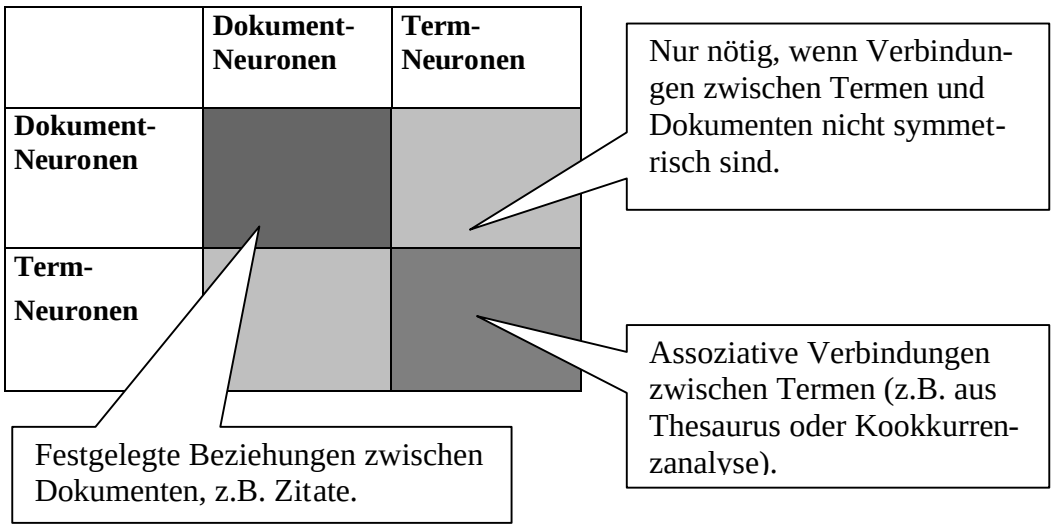


Abbildung 4-13: Die vollständige Verbindungsmatrix ermöglicht zum einen assoziative Verbindungen innerhalb von Schichten. Daneben veranschaulicht diese Darstellung, dass die Symmetrie der

Verbindungen zwischen Dokumenten und Termen nicht selbstverständlich ist.

Die Einführung einer zusätzlichen Schicht für Autoren, wie sie manche Systeme vornehmen, eröffnet weitere Möglichkeiten, wie Abbildung 4-14 zeigt.

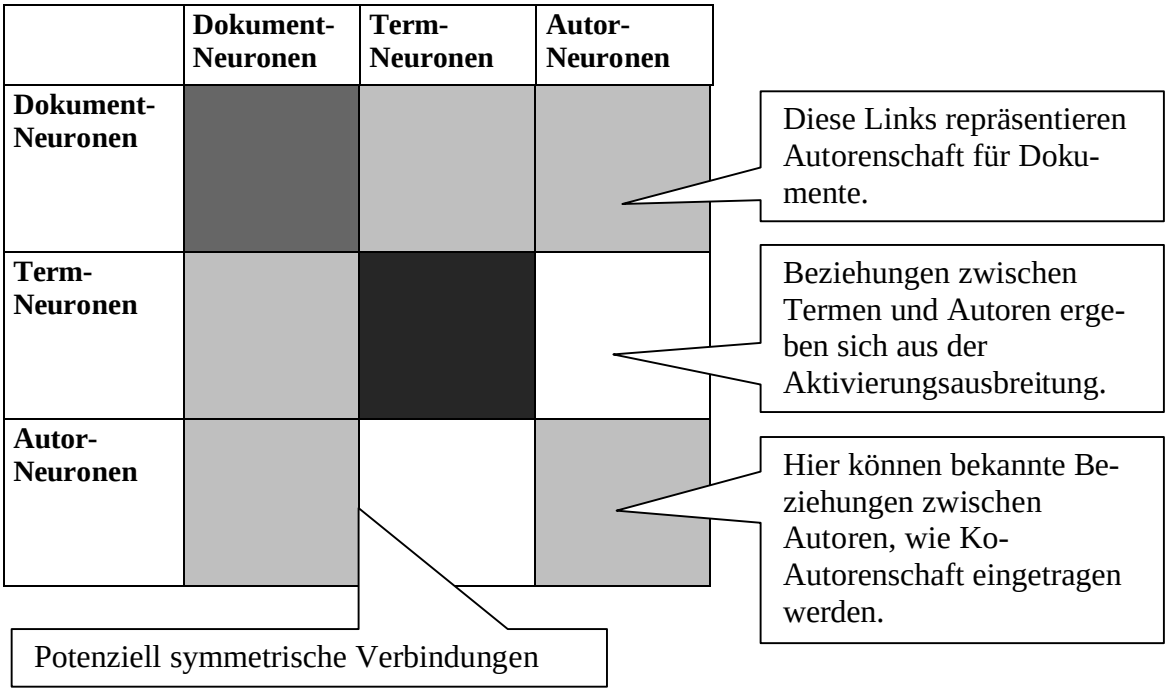


Abbildung 4-14: Die Verbindungsmatrix nach der Einführung einer Autoren-Schicht. Die nicht beschrifteten Verbindungstypen sind bereits in den obigen Abbildungen enthalten.

Dieser Überblick unterstreicht erneut die Nähe zwischen Spreading-Activation-Modellen und dem Vektorraum-Modell.

4.3.4 Fazit: Spreading-Activation-Modelle

Die Spreading-Activation-Modelle haben sich im Information Retrieval als mögliche Alternative zum probabilistischen Modell und dem Vektorraum-Modell etabliert. Die abschließende Bewertung berücksichtigt folgende Vor- und Nachteile:

- Vorteile der Spreading-Activation-Netze:
  - Erfolg einiger implementierter Systeme
  - Tragfähige Metapher des IR-Prozesses

  - Flexibilität innerhalb des Modells
- Nachteile der Spreading-Activation-Netze:
  - Ansätze zur Flexibilität werden allgemein wenig benutzt
  - Erfolgreiche Systeme schöpfen die Flexibilität nicht aus
  - Spreading-Activation-Netze sind formal dem Vektorraum-Modell sehr ähnlich
  - Diese Ähnlichkeit ist in den implementierten Systemen besonders hoch
  - Die Lernfähigkeit ist niedrig ausgeprägt
  - Spreading-Activation-Netze nutzen nicht alle Stärken neuronaler Netze

Die Spreading-Activation-Modelle bieten einige erhebliche Vorteile:

- Die Leistungsfähigkeit der Spreading-Activation-Modelle ist mit anderen Information Retrieval Techniken vergleichbar, wie die guten Ergebnisse von PIRCS und Mercure bei den TREC-Studien gezeigt haben (cf. Abschnitt 4.8, cf. Voorhees/Harman 1998a, 1999a). Für eine endgültige Entscheidung, welches Modell besser ist, reichen diese Ergebnisse allerdings nicht aus. Ein solches Ergebnis ist auch auf lange Sicht nicht zu erwarten, da die Eigenschaften der Anwendungsbereiche sehr unterschiedlich sind und Information Retrieval Systeme bei Vergleichen mit unterschiedlichen Kollektionen sehr unterschiedliche Ergebnisse erreichen.
- Die Ausbreitung von Aktivierung als Metapher für Relevanz und Interesse des Benutzers ist sehr plausibel. Die Aktivierung eines Dokuments durch seine Index-Terme ist anschaulich und Term-Expansion und Relevanz-Feedback ergeben sich in diesem Rahmen sehr natürlich. Wie die Diskussion der Funktionsweise in Abschnitt 4.3.1 unterstreicht, bieten Spreading-Activation-Netze eine gute Metapher des Information Retrieval Prozesses.
- Die Spreading-Activation-Modelle sind äußerst flexibel. Prinzipiell können sowohl Dokumente als auch Index-Terme gleichzeitig als Input und Output dienen. Weiterhin sind nach jedem Aktivierungsschritt Eingriffe des Benutzers möglich. So kann als zusätzlicher Input zu jeder Zeit die Auswahl und die Aktivierung zusätzlicher Terme oder Dokumente erfolgen. Die Möglichkeiten des Benutzers und somit die Interaktivität des Retrievalprozesses können im Rahmen des Modells auf einfache Weise erhöht werden.

Den Vorteil der erhöhten Flexibilität schränken die folgenden Beobachtungen ein:

- Trotz der Plausibilität einer Erhöhung der Interaktivität durch die Möglichkeit zahlreicher Eingriffe durch den Benutzer lässt sich diese Vorgehen wohl kaum in die Praxis übertragen. Zahlreiche Untersuchungen zeigen, dass Relevanz-Feedback als einfache Art der Interaktion nachweislich eine der besten Strategien zur Verbesserung des Retrievalergebnisses ist. Trotzdem wird es von Benutzern relativ selten eingesetzt (cf. Womser-Hacker 1997). Sie scheinen bereits diesen Aufwand zu scheuen, so dass kaum davon auszugehen ist, dass eine verstärkte Interaktion überhaupt genutzt würde. Die große Flexibilität der Spreading-Activation-Netze hinsichtlich Eingabe und Möglichkeiten der Interaktion kann somit nicht als entscheidender Vorteil gewertet werden.
- Die Flexibilität u.a. durch häufiges Feedback ist insbesondere bei den in TREC eingesetzten Systemen nicht realisiert und nicht vorgesehen. Sie setzt die mehrfache Aktivierungsausbreitung zwischen den Schichten voraus, die nur wenige Systeme erlauben.

Die letzte Beobachtung führt zur Frage, was die Spreading-Activation-Modelle als IR-Modell auszeichnet. Die von vielen Autoren betonte Nähe zu den klassischen Modellen deutet bereits an, dass die Spreading-Activation-Modelle keine völlig neuartige Modellklasse darstellen.

- Die Standard Spreading-Activation-Netzwerke für Information Retrieval ähneln stark der Vektorraum-Repräsentation. Doszkocs et al. 1990:231 drücken dies folgendermaßen aus: „A connectionist network representation can be compared with the well-known vector space model of information retrieval“. Auch Chen 1995:199 teilt diese Ansicht: „Neural networks computing, in particular, seems to fit well with conventional retrieval models such as the vector space model and the probabilistic model.“
- Die Ähnlichkeit geht jedoch über eine formale Ähnlichkeit der Repräsentation weit hinaus. So weist Mothe 1994 (cf. Abschnitt 4.3.2.4) theoretisch und empirisch nach, dass ein Spreading-Activation-Modell nach einem Aktivierungsschritt äquivalent zum Vektorraum-Modell ist. Erst wenn weitere Aktivierungsschritte folgen, unterscheidet sich das Spreading-Activation-Netzwerk von etablierten IR-Modellen. Dies ist jedoch nur in wenigen Systemen der Fall und dazu gehören nicht die in TREC überprüften Systeme PIRCS und Mercure. Damit wird auch dieser potenzielle Vorteil der Spreading-Activation Modelle nicht ausgenutzt. Möglichweise

induziert der Aufbau der Standard-TREC Tests diese Einschränkung, da Relevanz-Feedback darin eine untergeordnete Rolle spielt.

- Wie in vielen anderen Information Retrieval Modellen kann auch bei den Spreading-Activation-Modellen die Bedeutung heuristischer Faktoren nicht unterschätzt werden. Dazu zählen die Wahl der Aktivierungsfunktion, die Lernregel, die Anzahl der Aktivierungsschritte und die Wahl zahlreicher anderer Parameter. Besonders wichtig sind in diesem Zusammenhang hemmende Mechanismen, die ein schnelle und vollständige Aktivierung des Netzes nach einigen wenigen Schritten verhindern.

Aus der Sicht der neuronalen Netze bestehen Schwächen der Spreading-Activation-Modelle darin, dass die Lernfähigkeit niedrig ist und keine versteckte Schicht die Mächtigkeit der Netze erhöht. Eine versteckte Schicht mit symbolisch nicht interpretierbaren Neuronen erhöht die Leistungsfähigkeit eines Netzes wesentlich (cf. Abschnitt 3.5.4.1, cf. Zell 1994:100f.).

- Bei den Vorstellungen der Spreading-Activation-Netze wird die Initialisierung des Netzes anhand der Termhäufigkeiten meist als Lernen bezeichnet. Dabei handelt es sich jedoch nicht um Lernen, sondern lediglich um das Einstellen der Verbindungen auf die Ausgangssituation. Die Möglichkeit, ohne die initialen Verbindungsstärken ein Netz allein aufgrund von Benutzerurteilen zu trainieren, wird teilweise pessimistisch eingeschätzt. Dabei gilt vor allem die notwendige Größe der Netze für realistische Anwendungen als wichtiges Argument:

  > „The possibility of developing learned connection strengths based on a large number of queries and the desired document weight is infeasible given the size of the network. The connection weights are determined using techniques developed for information retrieval and are fixed.“ (Wilkinson/Hingston 1992:72)

- Die größte Schwäche der Spreading-Activation-Modelle wird aus der kognitionswissenschaftlichen Warte deutlich. Die große Stärke neuronaler Netze besteht in ihrer sub-symbolischen Leistungsfähigkeit, die besonders der Backpropagation-Ansatz repräsentiert. Dieses Potenzial wird nicht genutzt, keines der Modelle verfügt über versteckte Schichten, die nicht symbolisch interpretiert werden können. Diese Einschätzung beruht jedoch nicht nur auf der kognitionswissenschaftlichen Theorie. Das Backpropagation-Netzwerk ist das mit Abstand am häufigsten eingesetzte Modell im Rahmen der neuronalen Netze. Es kann umfassende Klassen von Funktio-

nen implementieren, von denen viele von zweischichtigen Netzen nicht erreicht werden können. Da nicht bekannt ist, von welcher Art und Komplexität eine optimale Ähnlichkeits- oder Match-Funktion ist, sollte ein möglichst mächtiges Modell gewählt werden.

Trotz der genannten Schwächen bilden die Spreading-Activation-Modelle ein plausibles Modell, das den verbreiteten Information Retrieval Modellen formal sehr ähnlich ist und bei TREC teilweise vergleichbare Ergebnisse erzielen konnte (cf. Abschnitt 4.8). Bereits modellintern ergeben sich Verbesserungsmöglichkeiten. Unter verschiedenen Blickwinkeln fallen Schwächen des Modells ins Auge. Die semantischen Netze bilden für den wenig strukturierten Bereich des Text-Retrieval keine Alternative. Im Folgenden werden daher weitere IR-Modelle mit neuronalen Netzen analysiert.

4.4 Kohonen-Netze im Information Retrieval

Kohonen-Netze oder selbstorganisierende Karten (Self-Organizing-Maps, SOM) erfreuen sich in den letzten Jahren im Information Retrieval steigender Beliebtheit. Ihre Funktionsweise erläutert Abschnitt 3.5.1. Kohonen-Netze dienen als unüberwachtes Cluster-Verfahren, das ähnliche Muster auf topologisch nahe Neuronen in der Kohonen-Schicht abbildet. Die Abbildung von Dokumenten aus dem vieldimensionalen Termraum auf eine zweidimensionale dargestellte Karte nutzen viele Ansätze als Möglichkeit zur Visualisierung und Interaktion. Dahinter steht die Hoffnung, dass semantisch ähnliche Dokumente oder Terme in der Kohonen-Schicht nahe beieinander liegen. Gloor 1997 sieht Ähnlichkeit neben z.B. Verlinkung und Sequentialisierung als eines von sieben Konzepten für die Gestaltung von Navigation in Hypertext-Systemen wie dem Internet.

4.4.1 Grundprinzip

Clustering mit dem Kohonen-Netz stellt eine Form der Dimensionsreduktion dar. Weitere Verfahren zur Komprimierung des Merkmalsraums im IR wie etwa Latent Semantic Indexing (LSI) diskutiert Abschnitt 2.1.2.4.

Bei der Reduktion gehen natürlich immer Aspekte des Ausgangsraums verloren wie Abbildung 4-15 illustriert. Bei LSI werden diese Verluste durch Weglassen kleiner Singular Values gesteuert, die bei der Rekonstruktion zu Verlusten führen. Bei SOM ist die Rekonstruktion nicht möglich und es gibt keine Abschätzung über die Qualität der Komprimierung. Während LSI und

ähnliche Verfahren nach der Reduktion eine Ähnlichkeitsberechnung anschließen, dienen Self Organizing Maps vorwiegend der Visualisierung.

Bei explorativen und assoziativen Suchen sollen Benutzer ausgehend von bekannten Objekten entlang der Cluster interessante Dokumente oder Terme finden. Die Verwendung des Begriffs Dimension für die Kohonen-Schicht in der Literatur ist zweideutig. Ein Kohonen-Netz bildet grundsätzlich einen n-dimensionalen Raum in einen m-dimensionalen ab. Während die Dimensionen der Eingabe-Schicht durch die Daten vorgegeben sind, ist die Anzahl der Klassen in der Kohonen-Schicht beliebig. Die Anzahl wird heuristisch festgelegt und sollte der Anzahl der gewünschten Klassen entsprechen. Die Kohonen-Schicht dient meist der Visualisierung und ordnet die Neuronen zweidimensional an. Die m Dimensionen oder Cluster werden zu Koordinaten von Punkten im zweidimensionalen Darstellungsraum. Grundsätzlich können die Ausgangs- oder Kohonen-Neuronen jedoch beliebig und in mehr als dreidimensionalen Räumen angeordnet werden.

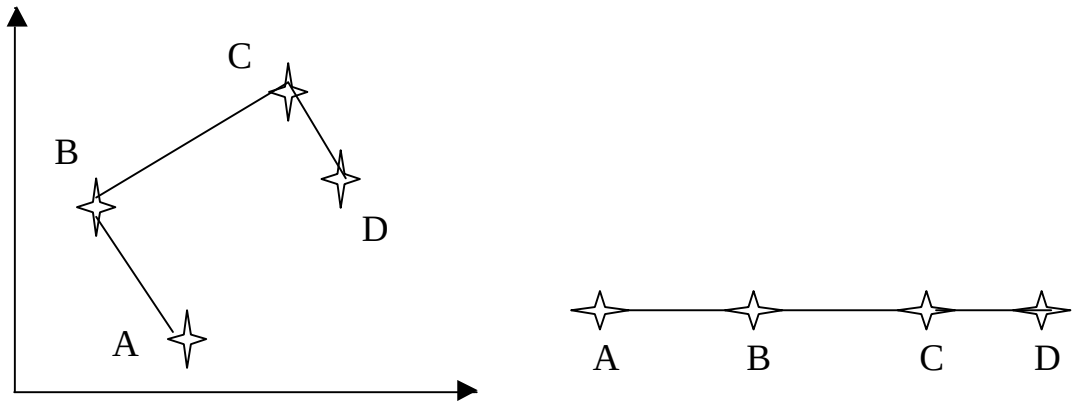


Abbildung 4-15: Reduktion durch Abbildung von einem zweidimensionalen auf einen eindimensionalen Raum. Bereits dieses einfache Beispiel zeigt, dass die Abbildung nicht völlig ähnlichkeitserhaltend ist. In der zweidimensionalen Darstellung sind sich die Objekte A und D ähnlicher als B und C. Auf der Geraden haben A und D dagegen minimale Ähnlichkeit.

Die Mächtigkeit von Kohonen-Netzen bestätigen z.B. Graupe/Kordylewski 1998, die ein System für medizinische Diagnose mit realen Daten vorstellen. Das System klassifiziert Patientendaten besser als statistische Verfahren und andere neuronale Netze. Es implementiert eine Datenbankabfrage mit unvollständigen Mustern. Die im Folgenden beschriebenen Systeme bearbeiten Probleme des Text-Retrieval mit dem Kohonen-Netz.

4.4.2 Systeme

Lin et al. 1991 nutzen ein Kohonen-Netz für das Erstellen einer zweidimensionalen semantischen Karte. Eine Kollektion von 140 Artikeln zur Künstlichen Intelligenz wurde anhand der Titel indexiert und mit einem Kohonen-Netzwerk verarbeitet. Die Dokumentvektoren mit der Länge 25 dienten als Input für das Training. Das heißt, es wurden nur 25 Terme analysiert. Die Punkte in der Karte repräsentieren Dokumente, die Regionen sind aber nach Termen benannt. Dazu erstellten Lin et al. 1991 Term-Vektoren, in diesem Fall Einheitsvektoren, die ein Dokument mit nur einem Term nachbilden. Der Term-Vektor mit der höchsten Ähnlichkeit zu einem Punkt oder Neuron in der Karte bzw. dessen Gewichtsvektor bestimmt die Benennung. Die Region eines Terms erstreckt sich nur über benachbarte Neuronen, da das Lernverfahren der Kohonen-Karte Ähnlichkeit als Nähe im zweidimensionalen Raum darstellt. Lin et al. 1991 beobachten, dass die Dokument-Vektoren neben den Termen, mit denen sie indexiert sind, weitere Terme mit Gewichten ungleich Null enthalten. Diese bilden assoziative Erweiterungen der Indexterme, da sie in der Kollektion häufig gemeinsam mit ihnen vorkommen. Aufgrund der starken Dimensionsreduktion eignet sich die Karte als Grundlage einer grafischen Benutzungsoberfläche für assoziative Navigation im Korpus. Die Autoren schlagen vor, die Karte zur Visualisierung des Ergebnisses eines IR-Systems einzusetzen. Eine Diskussion der Kohonen-Netze als Benutzungsoberfläche folgt am Ende dieses Abschnitts.

Scholtes 1992 präsentiert eine SOM für Information Retrieval, die sich im Repräsentationsmechanismus vom Standard-IR-Modell unterscheidet. Das System für russische Texte nutzt N-Gramme von Buchstaben und nicht Wörter als Eigenschaften der Dokumente. Eine Häufigkeitsverteilung über die 225 vorkommenden Trigramme ersetzt die Terme in Form von Wörtern. Diese Repräsentationen werden zunächst mit der Repräsentation von Texten durch Terme verglichen. Eine interessante Eigenschaft ist, dass auch ohne morphologische Analyse eine hohe Ähnlichkeit zwischen verschiedenen Wortformen eines Begriffs besteht. Insgesamt scheint die Repräsentation durch Häufigkeitsverteilungen von N-Grammen besser für sprachliche Analysen, wie z.B. die Identifizierung von Sprachen oder die Zuordnung von Buchstaben zu Phonemen, geeignet zu sein. Für Probleme der Semantik wie im Information Retrieval scheinen Terme besser geeignet zu sein. Endgültig kann dies jedoch nur eine empirische Überprüfung klären.

Im Zusammenspiel mit dem Kohonen-Netz führt die Repräsentation durch Trigramme zu zahlreichen Detailproblemen. So sind z.B. die häufigsten

Trigramme nicht in der 15x15 Neuronnen umfassenden Kohonen-Karte vertreten. Da das Kohonen-Netz ein unüberwachtes Netz ist, ergeben sich die Klassen, die in der Karte repräsentiert sind, nur aus dem Lernalgorithmus und nicht aus äußeren Vorgaben. Nach der Einschätzung von Scholtes 1992 identifiziert das Netz die für die Aufgabe am besten geeigneten Trigramme. Die Trigramm-Analyse berücksichtigt die Leerzeichen (Blanks) zwischen den Wörtern nicht, so dass das Konzept Wort keinerlei Rolle spielt.

Der Autor räumt ein, dass er für den Kontext Prawda typische Stoppwörter wie *Held* oder *Kommunismus* nicht eliminiert. Die Aussagekraft der durchgeführten Tests ist sehr gering, da die Testkollektion nur 50 Texte umfasst. Die Aufgabe im Test ist unklar und es werden keine Standard-Maße wie Recall und Precision untersucht. Die Tests scheinen die Wiedererkennung von Texten und ihre Zuordnung zu anderen, ähnlichen Texten aus der Kollektion zu bewerten. Ähnlichkeit wird durch Themengleichheit in mindestens einem Abschnitt definiert. Reale Benutzer oder reale Anfragen fließen offensichtlich nicht ein. Es zeigte sich lediglich, dass die Qualität bei längeren N-Grammen besser wird, was auf die höhere Adäquatheit von Wörtern als Repräsentationsmechanismus hinzuweisen scheint.

Rozmus 1995 kritisiert an dem Original-Algorithmus von Kohonen, dass die Muster nicht homogen in der Output-Schicht verteilt sind. In seiner Implementierung, die 50.000 bibliographische Angaben in Cluster einteilt, verwendet er einen veränderten Algorithmus, der die Muster besser im Zielraum verteilt. Dabei stellt sich die Frage, inwieweit die Daten nicht homogene Verteilungen erfordern. Da ein Clustering-Algorithmus aber ohnehin nicht die gesamte, komplexe Struktur wiederspiegelt und eine bessere Verteilung die Benutzerfreundlichkeit erhöht, können derartige Überlegungen durchaus sinnvoll sein.

Lesteven et al. 1996 berichten von der Erstellung einer Kohonen-Karte für Literatur aus dem Bereich der Astronomie. Eine 20x16 Neuronen umfassende Karte bildet 2063 bibliographische Angaben ab. Als Input dienten 463 Terme aus einem astronomischen Thesaurus. Die entstandene Karte und die darin gefundene Cluster bilden den Kern einer Benutzungsoberfläche für die Dokument-Kollektion auf der Basis von HTML. Die Autoren geben keine Einschätzung zur Qualität der Karte.

Zavrel 1996 wendet eine Erweiterung des Kohonen-Netzes u.a. auf die Cranfield-Kollektion (cf. Abschnitt 7.1.1) an. Nach einer Einordnung in die Forschungskontexte Visualisierung, Kohonen-SOM und neuronale Netze im IR präsentiert Zavrel 1996 seine Erweiterung des Kohonen-Algorithmus, bei der während des Trainings neue Neuronen in die Kohonen-Schicht eingefügt werden. Dazu misst jedes Kohonen-Neuron die Größe des von ihm abge-

deckten Eingaberaums. Das Neuron, das den größten Bereich von potenziellen Eingabemustern vertritt, erhält einen Nachbarn, der diesen Raum mit ihm teilt und so eine bessere Differenzierung zulässt. Damit verfolgt Zavrel 1996 ein ähnliches Ziel wie Rozmus 1995, der auch eine homogene Verteilung der Muster anstrebt.

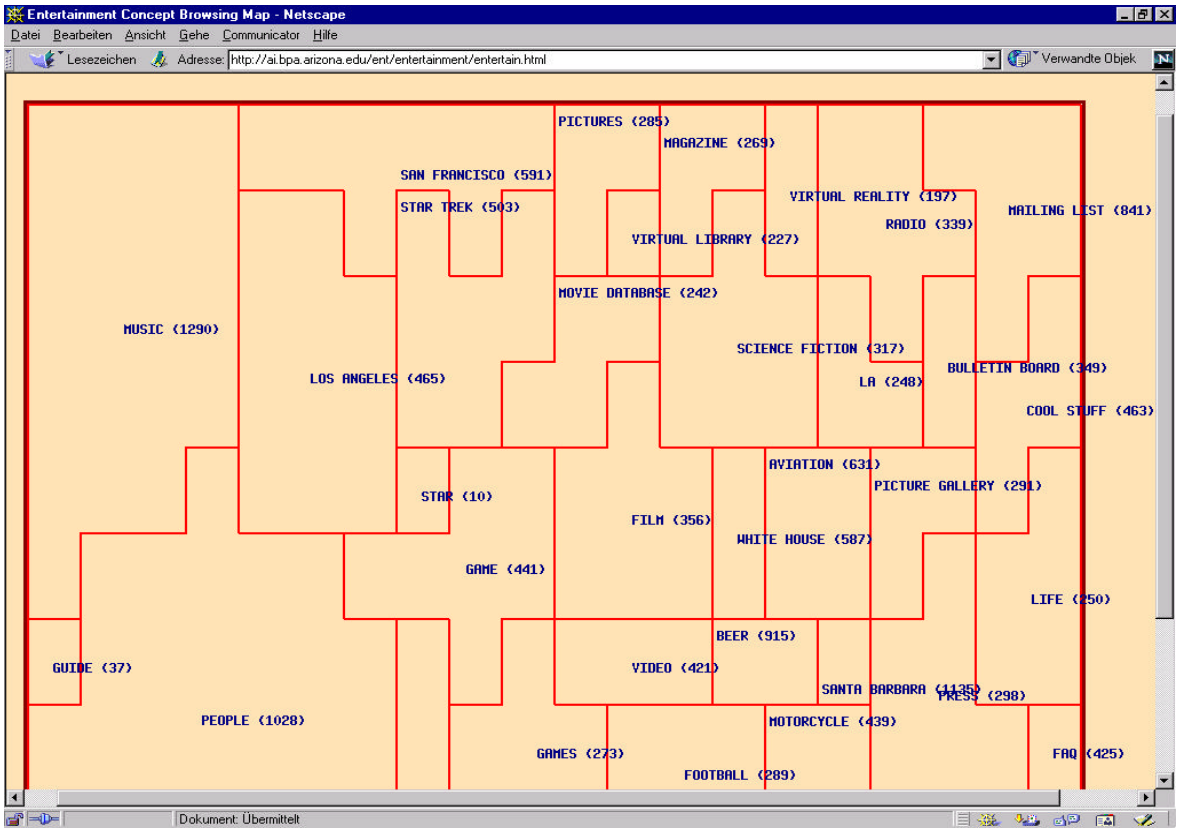


Abbildung 4-16: Ein Ausschnitt aus der Visualisierung der Kohonen-Karte von Chen et al. 1996

Zavrel 1996 evaluiert die Qualität des Kohonen-Netzwerks als IR-System im Vergleich zu anderen Clustering-Methoden und dem Vektorraum-Modell. Dazu wird das Clustering-Verfahren als IR-Modell interpretiert und die Anfrage dient als Eingabe im Kohonen-Netz. Der Cluster, den das aktivierte Ausgabe-Neuron repräsentiert, ist die Ergebnismenge. Die Evaluierung erfolgt also auf der Basis einer booleschen Menge und kommt damit dem Clustering-Verfahren entgegen. Das Ergebnis des Vektorraum-Modells, eine gerankte Liste aller Dokumente (cf. Abschnitt 2.1.2.2), wird ebenfalls als boolesche Menge interpretiert. Das Vektorraum-Modell verwendet in diesem Fall das Kosinus-Ähnlichkeitsmaß. Die Anzahl der Dokumente bestimmt Zavrel 1996 aus der Anzahl der durchschnittlich in den Clustern enthaltenen Dokumenten. Die vom Vektorraum-Modell gelieferte Liste von Dokumenten

wird bei einem Schwellenwert in zwei boolesche Mengen geteilt. Die Clustering-Verfahren geben damit das Retrieval-Verfahren vor und die im Vektorraum-Modell enthaltene Reihenfolge der Dokumente geht für die Bewertung verloren. Zudem wird die Größe der zu berücksichtigenden Menge aus dem Durchschnitt der Clustering-Verfahren berechnet. Trotzdem schneidet das Vektorraum-Modell in dem Vergleich von Zavrel 1996 besser ab als alle Clustering-Verfahren, unter denen das modifizierte Kohonen-Netz die beste Qualität erreicht. Als Maßstab zieht Zavrel 1996 das E-Maß von van Rijsbergen heran, das Recall und Precision kombiniert (cf. Abschnitt 2.1.4.1).

Chen et al. 1996 übertragen den Kohonen-Ansatz auf Internet-Dokumente. Trotz der Beschränkung auf eine Kategorie des Internet-Katalogs Yahoo (entertainment) führen sie eine Vielzahl semantischer Karten ein. Sobald ein Cluster mehr als eine bestimmte Menge von Seiten enthält, wird er auf einer eigenen Karte dargestellt. Die auf Basis einer Trainingsmenge von 10.000 Seiten entstehenden Karten wurden in Benutzertests empirisch getestet. Dabei zeigte sich, dass viele Versuchspersonen im Umgang mit der Karte ein Gefühl der Desorientierung befiel, das auch in assoziativ verknüpften Hypertexten vorkommt und als „lost in hyperspace“-Syndrom bekannt ist (cf. Kuhlen 1991). Die Orientierung fiel oft schwer und die Benutzer fragten nach geordneten Listen der enthaltenen Suchwörter. Die Karte eignet sich nach Einschätzung der Autoren vor allem für assoziative Suchen, bei denen Benutzer einen Überblick über die vorhandenen Suchbegriffe wünschen.

Eine kleinere Implementierung dieses Systems organisiert Äußerungen aus elektronischen Online-Besprechungen (cf. Orwig et al.1997). Die Testmenge umfasst 202 Dokumente mit 190 Termen und damit Eingabedimensionen. Die Dokumente werden auf eine 10x20 Neuronen große Kohonen-Schicht abgebildet. Die Qualität der Cluster wird mit intellektuell erstellten Cluster verglichen, wobei die maschinell erstellten etwas schlechter waren. Die Kosten für die Erstellung sind jedoch erheblich niedriger. Falls in der Domäne solche Cluster für Analysen von elektronischen Treffen regelmäßig erstellt werden, bietet sich die Kohonen-Karte als Alternative an.

Das Problem der Überladung einer Karte tritt in diesem Anwendungsfall kaum auf, da jede Besprechung nur eine begrenzte Anzahl von Redebeiträgen umfasst.

Eine weitere Implementierung einer Kohonen-Karte, die als Interface im Internet zur Verfügung steht, ist WEBSOM[1] (cf. Kohonen 1997/1998, Honelka et al. 1997). WEBSOM stellt Karten mit großen Mengen von Newsgroup-Artikeln zur Verfügung. Abbildung 4-17 zeigt die Eingangsseite. Aufgrund

[1] http://websom.hut.fi/websom

der großen Anzahl von Dokumenten ordnen die Autoren wie im Ansatz von Chen et al. 1996 die Karten in mehreren Schichten an. Die Cluster in der ersten Karte führen also zu weiteren Karten und erst beim dritten oder vierten Schritt gelangt der Benutzer zu einem Cluster von Newsgroup-Artikeln.

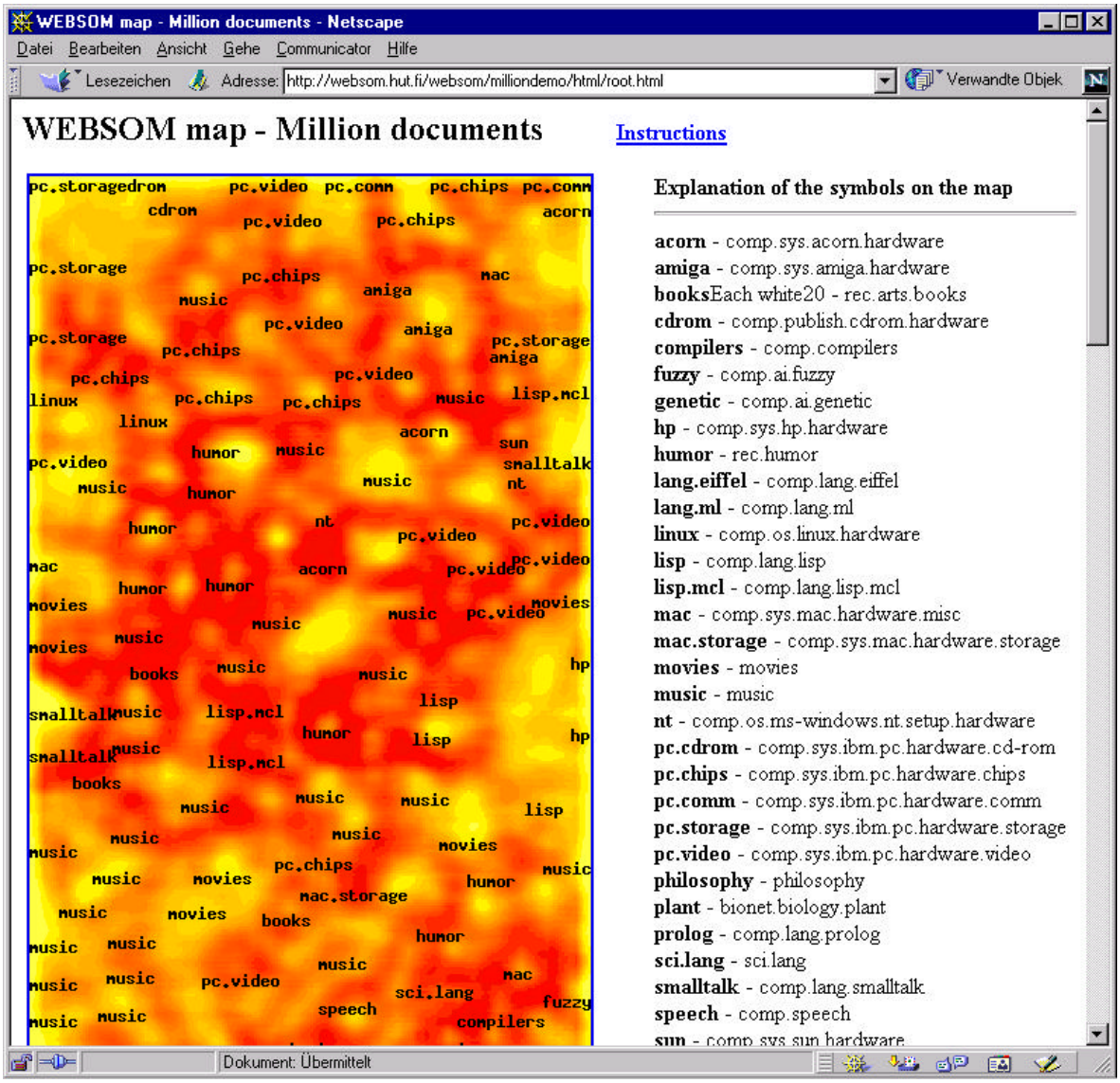


Abbildung 4-17: Einstiegsseite der WEBSOM-Karte

Die Abbildung 4-18 zeigt eine Karte vor diesem letzten Schritt. Die Interaktion beschränkt sich bei mehrschichtigen Systemen nicht auf das Klicken in einer Karte. Zusätzlich ist in der Seite ein Kontrollelement integriert, um zu einer anderen Karte zu wechseln. Die Beschriftung der Cluster ist stark verkürzt, was die Interaktion erschwert. Der Benutzer muss die Langform der Einträge rechts außerhalb der Visualisierung lesen. Dies stört die angestrebte Direktheit der Interaktion.

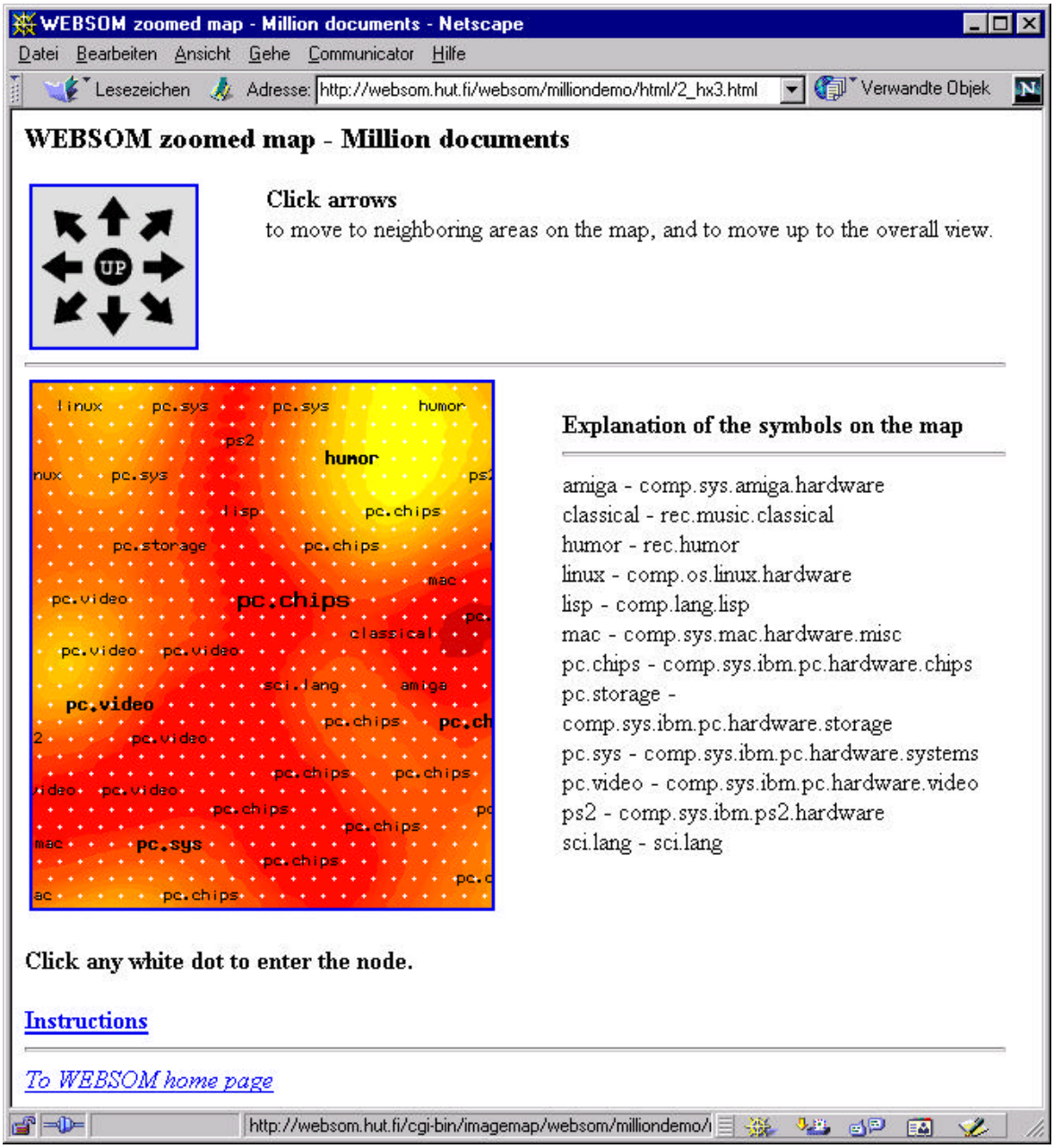


Abbildung 4-18: Benutzungsoberfläche der WEBSOM-Karte mit News-Artikeln

Kaski 1998 schlägt die Methode Random-Mapping für Dimensionalitätsreduktion als Vorverarbeitung für Kohonen-SOM vor. Dabei wird die originale Dokument-Term-Matrix mit einer Matrix zufällig gewählter Werte multipliziert. Die Größe der Zufallsmatrix bestimmt die Reduktion der Dimensionalität in der Ergebnismatrix. Kaski 1998 führt das Random-Mapping ein und experimentiert mit einer Kollektion von 1800 Artikeln aus 20 Newsgroups. Die ursprünglich 5781 Dimensionen werden mit Random-Mapping auf 90 Dimensionen und mit Principal-Component-Analysis (PCA,

eine Art Faktorenanalyse) auf 50 Dimensionen reduziert. Dann klassifiziert eine Kohonen-Karte mit 768 Neuronen die Dokumente. Jede Klasse wird nach der Newsgroup benannt, von der sie die meisten Artikel enthält. Die Qualität misst Kaski 1998 über der Trennfähigkeit, indem er die Anzahl der Artikel bestimmt, die nicht zur in dem Cluster vorherrschenden Newsgroup gehören. Dies erscheint fragwürdig, da eine Zuordnung unabhängig von der Zugehörigkeit zu den Diskussionsforen semantisch sehr sinnvoll sein kann. Häufig senden Benutzer identische Nachrichten an mehrere Newsgroups. Nach Kaski 1998 erzielte Random-Mapping eine etwas bessere Trennfähigkeit als PCA. Beide waren wiederum etwa 1% schlechter als die Originalmatrix.

Kohonen 1998 wendet das Verfahren auf größere Datenmengen an. Eine Million Newsgroup-Artikel führen zu 63.773 Termen oder Dimensionen, die durch Ramdom-Mapping auf 315 Dimensionen reduziert wurden. Das Kohonen-Netz ordnete die Dokumente in über 100.000 Cluster. Mehrfach geschachtelte Karten erlauben den Zugriff. Inwieweit ein Browsing-Ansatz den Zugriff auf eine derart große Menge von Dokumenten erleichtert, ist unklar.

Lagus 1998 überträgt den WEBSOM-Ansatz auf eine sehr kleine Anzahl von Fachtexten und auf finnische Texte.

Schatz 1998 überträgt die Input-Muster auf einen dreidimensionalen Raum, in dem der Benutzer navigiert (*space flight*). Abbildung 4-19 zeigt die dreidimensionale Darstellung, in der jedoch die dritte Dimension nicht homogen besetzt ist. Nur die am Rand der zweidimensionalen Oberfläche liegenden Bereiche verlaufen nennenswert in der dritten Dimension. Dadurch entsteht eine Art Terrain, das sicher leichter zu verstehen ist, als ein homogen mit Dokumenten besetzter dreidimensionaler Raum. Allerdings ist fraglich, ob jedoch ein Vorteil gegenüber den obigen zweidimensionalen Darstellungen besteht. Eine Diskussion zwei- und dreidimensionaler Visualisierungen im Information Retrieval findet sich in Eibl 2000.

Lamirel/Crehange 1994 stellen das komplexe Retrievalsytem NOMAD vor, das den Dokumentbestand in Kohonen-Karten aufteilt und als Ergebnis von Anfragen Sichten auf die Ergebnisdokumente in ihrem Kontext bietet. NOMAD partitioniert manuell die Eigenschaften des Gegenstandsbereichs und erlaubt so die Suche nach bestimmten Kriterien. Die Anwendungsgebiete liegen im Faktenretrieval (Hundetypen, Fotos). Im Textretrieval sind solche Partitionen schwer zu finden. Lamirel/Crehange 1994 versuchen, kurz- und langfristige Strategien des Nutzers zu identifizieren und beim Retrieval auszunutzen.

MacLeod/Robertson 1991 stellen ein Cluster Verfahren auf der Basis eines neuronalen Netzes vor, das zwar unüberwacht lernt, aber wie ART zweifach die Ähnlichkeit überprüft. Dadurch entsteht zusätzlicher Rechenaufwand. MacLeod/Robertson 1991 prüfen in ihrem Ansatz vor allem die Effizienz dieses Verfahrens und stellen fest, dass ihr Algorithmus theoretisch mit hierarchischen Cluster Methoden vergleichbar ist. Sie vergleichen die Retrievalqualität einiger Varianten des Algorithmus für die Cranfield- und Keen-Kollektion mit dem E-Maß. Dabei gehen Sie vor wie Zavrel 1996. Die Dokumente werden vom Lernalgorithmus in Cluster geordnet und eine Anfrage wird auf ein Cluster abgebildet. Alle enthaltenen Dokumente bilden das Ergebnis. Der vorgestellte MacLeod-Cluster-Algorithmus wird nicht mit der Kohonen Self-Organizing Map, anderen Clustering-Verfahren oder anderen Information Retrieval Verfahren verglichen.

Hui/Goh 1996 implementierten ein komplettes IR-System, dessen Kern ein Kohonen-Netz bildet. Ihr System lernt aus Relevanz-Feedback, indem es seine Gewichte verändert, hat jedoch einige Schwächen. In der Trainingsphase extrahiert es Terme, die dann in eine kodierte Repräsentation umgeformt werden. Das Trainingsverfahren präsentiert sie dem Netz einzeln und der Kohonen-Algorithmus ordnet sie in Cluster. Beim Retrieval werden ebenfalls die Terme der Anfrage einzeln an das Netz gelegt und einem Cluster zugeordnet. Die Terme aller Cluster, die die Anfrage-Terme aktivieren, bilden dann die eigentliche Anfrage. Das Kohonen-Netz dient also der Termexpansion. Alle Dokumente, in denen die Terme in den aktivierten Clustern vorkommen, bilden die Ergebnismenge. Wie bei anderen Kohonen-Modellen entsteht dadurch ein unflexibles Retrieval-System.

Zudem schlagen Hui/Goh 1996 ungünstig kodierte Repräsentationen vor. Die Terme werden entweder auf Basis ihrer Schreibweise oder einer phonetischen Transkribierung in hexadezimale Zahlen umgewandelt. Dadurch werden zwar ähnlich geschriebene oder gesprochene Deskriptoren gefunden, eine semantische Ähnlichkeit zwischen ähnlichen Repräsentationen ist jedoch rein zufällig. Bei der Standard-Repräsentation repräsentiert jedes Vektorelement einen Term. Jedes Dokument und jede Anfrage besteht aus einem Vektor, in dem die Werte die Wichtigkeit des jeweiligen Terms für das Dokument ausdrücken. Dadurch haben ähnliche Dokumente ähnliche Repräsentations-Vektoren.

Hui/Goh 1996 integrieren als einzige unter den vorgestellten Kohonen-IR-Systemen Relevanz-Feedback. Für die Benutzerurteile geben sie eine Skala von eins bis zehn vor. Die Skalenwerte sind in einem Fuzzy-Modell den Konzepten *most important*, *relatively important* und *least important* zugeordnet. Ein Fuzzy-Algorithmus verändert ausgehend von den Benutzerurteilen die

Verbindungsgewichte. Als Vergleichsmaßstab für die Performanz des Systems bei der Wiedererkennung von Termen dient ein ART-Netzwerk. Die Performanz der SOM liegt minimal über der von ART, wobei die Auswirkungen von Relevanz-Feedback nicht untersucht werden. Die Tests basieren auf einer Kollektion von nur 100 Dokumenten, so dass sie nicht sehr aussagekräftig sind.

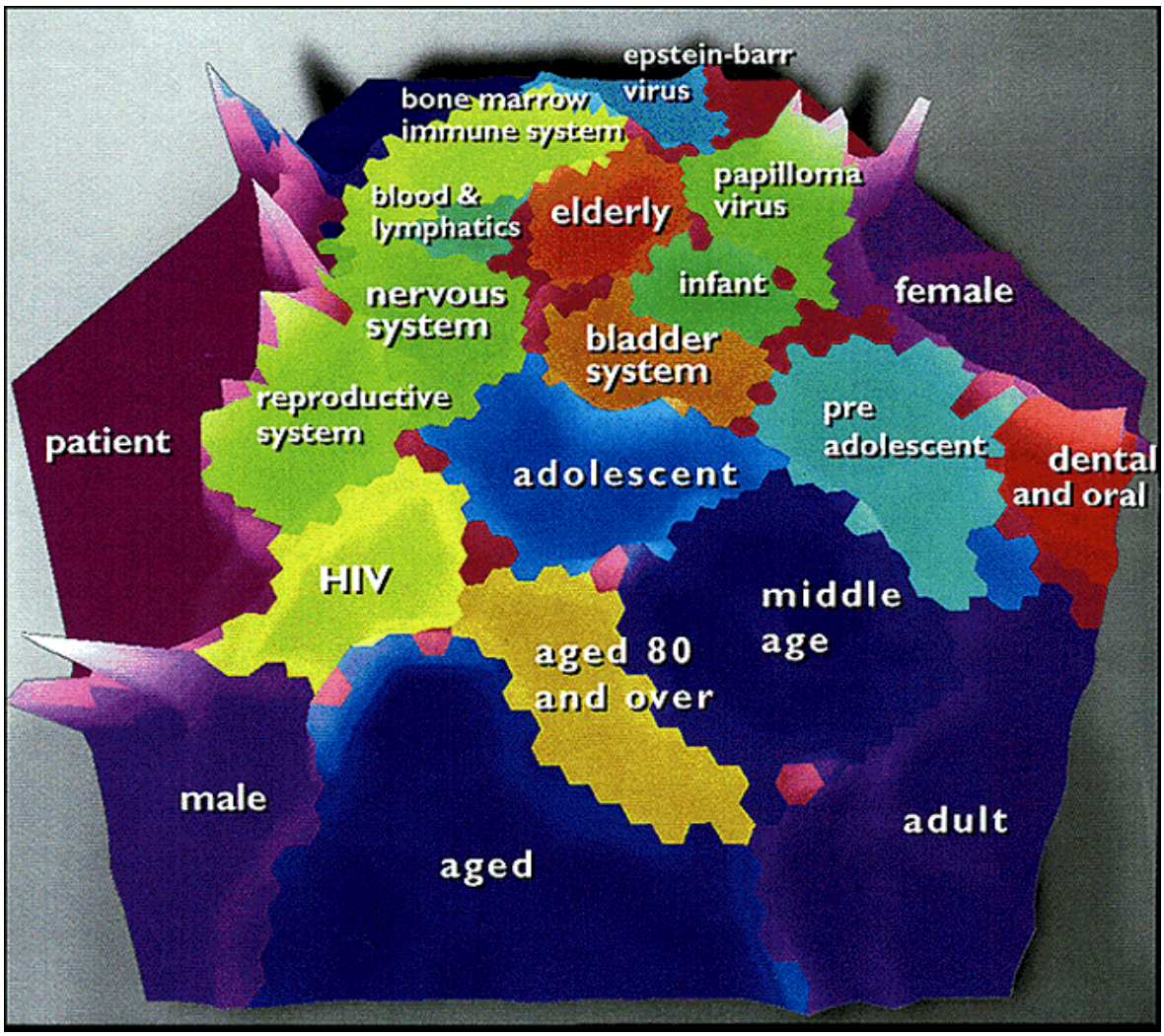


Abbildung 4-19: Visualisierung einer SOM als dreidimensionales Terrain (aus Schatz 1998)

Der hohe Zeitbedarf der Kohonen Self-Organizing Map gilt als einer ihrer größten Nachteile. Dem begegnen z.B. Kaski 1998 und Kohonen 1998 mit einer Komprimierung der Eingangsdaten. Merkl 1995 komprimiert mit einem Backpropagation-Netzwerk (cf. Abschnitt 2.1.2.4.2). Sein experimentelles IR-System sucht nach den Beschreibungstexten einer C++ Klassenbibliothek, aus der 489 Terme extrahiert wurden. In Merkl et al. 1994 werden für einen ähn-

lichen Anwendungsfall 39 Eigenschaften extrahiert. Das System von Merkl 1995 organisiert die Software-Komponenten auf einer zweidimensionalen Kohonen-Karte. Merkl 1995 verringert die langen Trainingszeiten des Kohonen-Netzes, indem er den Merkmalsraum von 489 auf 75 und einmal 30 Merkmale komprimiert. Ein Backpropagation-Netzwerk, bei dem Input und Output identisch sind, leistet die Komprimierung. Das Netz lernt, Inputmuster zu kopieren. Die dazwischenliegende versteckte Schicht enthält wesentlich weniger Neuronen als die Input- und Output-Schicht und damit der Merkmalsraum. Nach erfolgreichem Training enthält die versteckte Schicht eine reduzierte Repräsentation der Muster, die durch Beschreiten der Verbindungen von der versteckten Schicht in die Output-Schicht wieder expandiert werden kann.

Durch die Komprimierung konnte Merkl 1995 die Trainingszeit seines Netzes auf unter 20% der ursprünglichen Zeit drücken. Allerdings vergleicht er nicht die Qualität der komprimierten mit der ursprünglichen Repräsentation. Die Analyse des Netzes beschränkt sich auf eine intuitive Überprüfung durch den Autor. Die Eignung für reale *software-reuse*-Szenarien müsste in empirischen Untersuchungen überprüft werden.

Lelu/Claire 1992 unterstreichen die Wichtigkeit der Navigation neben der Suche als Paradigma im IR und implementieren eine zweidimensionale Karte mit Clustern von Dokumenten. Sie interpretieren den k-means-Clustering-Ansatz (cf. Ludwig/Mandl 1997) als ein neuronales Netz mit einer Schicht von Neuronen, die die Cluster repräsentieren. Der Ansatz ähnelt damit dem Kohonen-Netz.

4.4.3 Fazit: Kohonen-Netze im Information Retrieval

Dieser Überblick über implementierte Information Retrieval Systeme auf der Basis von Kohonen-Netzen zeigt vor allem, dass in realen Umgebungen der Umfang der zu verarbeitenden Daten für eine zweidimensionale Karte zu groß ist. Die meisten Karten erlauben weniger als 1000 Eingangsdimensionen und Muster. Chen et al. 1996 und WEBSOM (cf. Kohonen 1997/1998) testen mit realen Daten und lösen das Problem durch die Einführung mehrerer Schichten von Kohonen-Netzen, was weitere Interaktionsmechanismen zum Wechseln von einer Karte zur anderen erfordert. Das ursprüngliche Grundprinzip einer einfachen und scheinbar natürlichen Visualisierung wird dadurch überlagert. Eine weitere Schwäche besteht darin, dass das Kohonen-Netz nicht den Kern eines IR-Systems implementiert.

Die meisten Systeme liefern ein vollständiges Cluster von Dokumenten als Ergebnismenge. Die statische Zuordnung der Dokumente in die Cluster bildet

damit die Grundlage für das eigentliche Retrieval. In diesem Modell ist der dynamische Aspekt von Retrieval-Prozessen und die Integration von Relevanz-Feedback schwierig zu modellieren. Selbst wenn ein Benutzer einige Dokumente des Clusters als relevant und andere als nicht relevant einordnet, findet eine automatisch neu formulierte Anfrage in der Regel nur wieder das gleiche Cluster. Gelingt es einem Verfahren, dann ein zweites Cluster zu treffen, sind die relevanten Dokumente aus dem ersten gefundenen Cluster nicht mehr in der Treffermenge.

Grundsätzlich erscheint die Kohonen-Karte als Repräsentation einer Dokumenten-Kollektion problematisch. Um eine einfache und benutzerfreundliche Visualisierung zu erhalten, werden die zahlreichen Dimensionen des Term-Raums auf zwei reduziert. Dass eine solche Reduktion die Struktur der Kollektion noch adäquat widerspiegelt, ist eher unwahrscheinlich. Trotz der visuellen Adäquatheit ist die kognitive Repräsentation des Term-Raums wohl komplexer als eine zweidimensionale Karte. Die Clusterbildung ist stark kontextabhängig. Die zweidimensionale Darstellung erlaubt in einer größeren Kollektion insgesamt nur eine begrenzte Anzahl von Assoziationen in Form räumlicher Nachbarschaft. Zwar bietet die Kohonen-Karte für kleinere Mengen von Dokumenten einen möglichen Einstieg für assoziative und explorative Informationsbedürfnisse, als einziger Zugang ist sie in jedem Fall ungeeignet.

Einen interessanten Anwendungsfall aus Lin et al. 1991 greift Lin 1995 in seinen Tests mit verschiedenen kartenorientierten Darstellungen auf. Die Tests von Lin 1995 basieren auf einer sehr kleinen Menge von Daten. Die Aufgabe für die Benutzer bestand darin, aus 133 dargestellten Dokumenten zehn zufällig ausgewählte zu finden, was keine reale Retrievalsituation wiederspiegelt. Die Grundidee der Beschränkung auf kleine Mengen ist jedoch verfolgenswert. Dient die Karte nur der Ergebnisanzeige, ist das Problem der großen Datenmengen gelöst. Durch die Beschränkung auf eine kleine Aufgabe innerhalb des Retrievalprozesses könnten sich die Vorteile der Visualisierung in einem ausgewogenen Gesamtsystem besser entfalten. Optimierte Benutzungsoberflächen für Textretrieval versuchen, einzelne Komponenten so zu integrieren, dass ihre Nachteile durch andere Interaktionsmöglichkeiten aufgehoben werden, gleichzeitig aber ihre Vorteile voll ausgenutzt werden (cf. Krause/Schaefer 1998).

4.5 Adaptive Resonance Theory-Modelle

Information Retrieval Systeme auf Basis der Adaptive Resonance Theory (ART) sind bisher selten. Die Funktionsweise der ART-Netzwerke beschreibt

Abschnitt 3.5.2. ART ist ein unüberwachtes Clusterverfahren, das im Gegensatz zu Kohonen-SOM selbständig neue Cluster erzeugt. Für Anwendungen von ART im IR gelten ebenfalls die meisten der unter den Kohonen-Netzen (cf. Abschnitt 3.5.1) diskutierten Probleme von Cluster-Methoden.

Im letzten Abschnitt erschienen bereits zwei Ansätze, die ART oder ähnliche Verfahren einsetzen. Der Algorithmus von MacLeod/Robertson 1991 berechnet die Ähnlichkeit wie im ART-Algorithmus vorgesehen ebenfalls zweimal. Die Autoren betrachten ihre System als sehr nahe verwandt zu ART. Hui/Goh 1996 implementieren ein ART-Netzwerk als Vergleichsmaßstab zu ihrem Fuzzy-Kohonen-Netz. Die Fuzzy-Variante des Kohonen-Netzwerks erlaubt abgestufte Zugehörigkeitsgrade von Mustern zu Neuronen in der Kohonen-Schicht (zu Fuzzy Logik cf. Abschnitt 2.2.1). Dagegen ordnet das Standard-Kohonen-Verfahren ein Muster immer nur einem Neuron in der Kohonen-Schicht zu (cf. Abschnitte 3.5.1 und 4.4).

Das prominenteste System in diesem Kontext ist NIRS (Neural Information Retrieval System, cf. Escobedo et al. 1993, Caudell 1994, Smith et al. 1997). Die Firma Boeing setzt NIRS zum Retrieval ähnlicher Bauteile bei der Konstruktion von Flugzeugen ein. Damit ist es dem Fakten-Retrival zuzuordnen. Die Vagheit bei den Anfragen und die Unsicherheit bei der Repräsentation der Objekte führen in dem Anwendungsfall zu typischen Information Retrieval Problemen. In der aktuellsten Beschreibung von Smith et al. 1997 umfasst die Datenbasis über 90.000 Teile und mehrere Tausend Benutzer greifen darauf zu. Die Autoren realisierten Zugriffsmöglichkeiten sowohl auf PC-Basis, auf Workstation und per Browser.

Smith et al. 1997 beschreiben den Anwendungsfall. Untersuchungen hatten gezeigt, dass der Entwurf neuer Bauteile häufig nicht erforderlich ist, da entsprechende Bauteile bereits existieren oder durch kleine Modifikationen aus bestehenden Teilen entwickelt werden können. Effizientes Retrieval ähnlicher Bauteile vermeidet zahlreiche Neuentwürfe, die erhebliche Kosten verursachen. Ziel ist es, bei Eingabe eines neu zu entwickelnden Bauteils, ein Cluster ähnlicher, bereits existierender Teile zu finden. Bei der Entwicklung eines solchen Systems fanden die Konstrukteure bei Boeing keine für alle Anforderungen gültige Definition von Ähnlichkeit, sondern nur sehr allgemeine Kriterien, die eine Rolle spielen. Dies ist durchaus typisch beim Entwurf von Systemen, die intuitive Expertenaufgaben simulieren. Escobedo et al. 1993 wählten in dieser Situation ein nicht überwachtes, selbstorganisierendes Netz, das in der Lernphase keine Zielvorgaben benötigt.

Die Bauteile werden per Computer Aided Design (CAD) digital entworfen, so dass alle Daten maschinell lesbar zur Verfügung stehen. Die Autoren leiten daraus die Repräsentation der Bauteile in Form von binären Vektoren ab. Die

Komponenten werden zunächst in eine definierte standardisierte Lage gebracht. Dann extrahiert das System für jedes Teil drei geometrische Repräsentationen, die Umriss, Lage der Befestigungslöcher und Lage der Kanten abbilden. Diese dreifache Repräsentation erlaubt es dem Benutzer, das für seine Situation passende Kriterium zu wählen und die Anzahl der gefundenen Teile in gewissen Grenzen zu modifizieren. Damit hat der Benutzer Möglichkeiten, die ein unüberwachter Klassifikations-Algorithmus in der Regel nicht bietet.

Escobedo et al. 1993 ermöglichen dies durch einen modularen und hierarchischen Aufbau aus zahlreichen ART-1-Modulen. Das erste Modul sortiert alle Bauteile nach seinem Umriss in Cluster. Diese Cluster unterteilt eine zweite Schicht mit zwei ART-1-Netzen in feinere Cluster. Davon greift jeweils eines auf die Repräsentation der Befestigungslöcher und das andere auf die Kanten zu. In einer dritten Schicht werden die Ergebnisse der Cluster für Löcher und Kanten kombiniert. Damit kann ein Benutzer ausgehend von einem Input-Muster mit mehreren Kriterien nach ähnlichen Bauteilen suchen. Will er ähnliche Formen finden, durchläuft das System das erste Modul und liefert eine relativ große Ergebnismenge. Weitere mögliche Kriterien sind Löcher, Kanten oder beides zusammen. Das gesamte System umfasst ca. 8000 ART-1 Netzwerke (Caudell et al. 1994). Auffällig ist, dass Löcher und Kanten nicht erstes oder einziges Kriterium sein können. Offensichtlich spielt dies im Anwendungsfall keine Rolle.

Die Schachtelung mehrerer Schichten von Klassifikatoren ähnelt dem Vorgehen von Chen et al. 1996 und dem WEBSOM-System, die große Bestände von Internet-Seiten mit Schichten von Kohonen-Netzen organisieren (cf. Abschnitt 4.4.2). Bei Escobedo et al. 1993 motivieren diese Schichten jedoch semantisch und schaffen daraus keine komplexe Benutzungsoberfläche, welche die Karte mit hierarchischen Steuerungselementen mischt. NIRS nutzt die Klassifikation nicht zur Interaktion, sondern liefert je nach Kriterium die Cluster auf einem bestimmten Level. Die Komplexität von NIRS erhöht nicht die Komplexität für den Benutzer.

Caudell et al. 1994 adaptieren NIRS für dreidimensionale Repräsentation. Die Verwendung von Pixeln erzeugt sehr große Mustervektoren. Das Lernverfahren von ART ist komplexer als der Kohonen-Algorithmus und erfordert viel Zeit. Caudell et al. 1994 komprimieren die Muster-Vektoren mit einem spezifischen Verfahren für binäre Vektoren und verringern so die Lernzeit erheblich.

NIRS ist ein vielversprechender Ansatz. Es erreicht durch seine Architektur eine Erweiterung der Standard-Klassifikationsverfahren, die dem Benutzer Interaktionsmöglichkeiten bietet. Das System hat sich in der Praxis an realen

Daten bewährt, gerade weil es sich an den spezifischen Anwendungsfall anpasst. Deshalb lässt es sich nicht ohne weiteres auf andere Bereiche übertragen. Eine Anpassung an Textretrieval fällt wie für alle Clustering-Verfahren schwierig. Trotzdem ist das Faktenretrievalsystem NIRS aufgrund der vagen Natur der Repräsentation und der Vagheit des Anwendungsfalls Ähnlichkeit ein typisches IR-System (cf. Definition der Fachgruppe IR in der GI, Abschnitt 1).

4.6 Backpropagation-Netzwerke

Der Backpropagation-Algorithmus ist eines der mächtigsten und am häufigsten eingesetzten neuronalen Netze (cf. Abschnitt 3.5.4). Im Information Retrieval wird er allerdings eher selten eingesetzt. Der folgende Abschnitt stellt die existierenden Systeme vor. Abschnitt 6.2 entwirft eine Systematik für mögliche Information Retrieval Modelle mit dem Backpropagation-Algorithmus.

4.6.1 Lernen als Gradientenabstieg

Der Backpropagation-Algorithmus gehört zu den Gradientenverfahren, die eine Fehlerfunktion orthogonal zur Gradiente minimieren (cf. Zell 1994:106). Auch andere lernende Systeme, die keine neuronalen Netze nutzen, setzen teilweise Gradientenverfahren ein.

Lewis et al. 1996 und Papka et al. 1996 befassen sich mit Lernen aus Relevanz-Feedback und vergleichen ein Gradientenverfahren mit den im Information Retrieval weit verbreiteten Algorithmen von Rocchio und Widrow-Hoff (cf. Papka 1996). Diese Relevanz-Feedback Algorithmen versuchen, den berechneten Fehler zwischen Anfrage und den als relevant eingestuften Dokumenten durch neue Gewichtung der Anfrage-Terme zu minimieren.

Die Lernverfahren bestimmen die Gewichtungen der Terme in der neuen Anfrage nach Relevanz-Feedback des Benutzers. Während der Ansatz von Rocchio die Gewichte in einem Schritt berechnet, nähert sich das Gradientenverfahren von Lewis et al. 1996 und Papka et al. 1996 wie der Backpropagation-Algorithmus in kleinen Schritten einer besseren Lösung an. Lewis et al. 1996:300 schlagen folgende Formel vor:

$$w_{t+,j} = \frac{w_{t,j} \exp(-2h(w_t x_i - y_i)x_{t,j})}{\sum_{J=1}^{d} w_{t,j} \exp(-2h(w_t x_i - y_i)x_{t,j})}$$

w_j	Gewicht von Term j
x_i	Trainingsdokument i
y_i	Relevanzurteil des Nutzers zu Dokument i
h	Lernrate

Lewis et al. 1996 und Papka et al. 1996 testen den neuen Ansatz mit Teilen der TREC-Kollektion und erreichen bei den meisten Experimenten bessere Ergebnissen als mit dem verbreiteten Rocchio-Algorithmus für Relevanz-Feedback (cf. Baeza-Yates/Ribeiro-Neto 1999:119). Diese positiven Ergebnisse sprechen dafür, den Backpropagation-Algorithmus für Information Retrieval weiter zu testen.

4.6.2 Anfrage-Dokumenten-Vektor-Modell

Dieser Abschnitt bespricht einige bestehende Systeme, Abschnitt 6.2.2 stellt das Anfrage-Dokumenten-Vektor-Modell in den Rahmen möglicher Information Retrieval Architekturen mit dem Backpropagation-Algorithmus und diskutiert Vor- und Nachteile.

Mori et al. 1990 schlagen das Anfrage-Dokumenten-Vektor-Modell vor, das eine Term-Schicht in eine Dokument-Schicht abbildet. Je ein Neuron repräsentiert sowohl Terme als auch Dokumente in der jeweiligen Schicht. Damit bildet dieses Modell die konsequente Erweiterung der Spreading-Activation-Netzwerke um eine versteckte Schicht. Einige der Schwächen der Spreading-Activation-Netzwerke, die Abschnitt 4.3.4 zusammenfasst, sollen durch die versteckte Schicht gelöst werden. Das Netz wird mit dem Backpropagation-Algorithmus trainiert und kann dadurch komplexere Funktionen abbilden als die Spreading-Activation-Netzwerke, die nur die Leistungsfähigkeit eines Perzeptrons erreichen (cf. Abschnitt 3.5.4.1).

Mori et al. 1990 trainieren das Netz mit den bei der Indexierung extrahierten Termen und den dazugehörigen Dokumenten. Weitere Lernpaare bilden danach die Terme von Anfragen und dazu im Relevanz-Feedback-Prozess als relevant identifizierte Dokumente. Das Netz umfasst sieben versteckte Schichten, was auf jeden Fall zu viel ist. Eine oder zwei versteckte Schichten gelten als völlig ausreichend. Mori et al. 1990 hegen die Hoffnung, eine der mittleren Schichten interpretieren zu können und so Cluster von Dokumenten

zu finden. Eine nachträgliche symbolische Interpretation von sub-symbolischen Repräsentationen in versteckten Schichten ist jedoch sehr schwierig.

Mori et al. 1990 präsentieren beispielhaft die Interaktion eines Benutzers mit ihrem System. Das Gesamtmodell nimmt vor dem Retrieval im neuronalen Netz eine Term-Expansion vor und fordert den Benutzer zur Bewertung der gefundenen Terme auf. Experimente zur Messung der Qualität des gesamten Systems werden nicht vorgestellt.

Auch Creput/Caron 1997 stellen ein Anfrage-Dokumenten-Vektor-Modell vor, das mit einer versteckten Schicht und einer Backpropagation-Variante arbeitet. Die Autoren zeigen zunächst an einem einfachen konstruierten Beispiel, das plausibel wirkt, dass eine Funktion von Termen zu Dokumenten nicht unbedingt linear trennbar sein muss. Damit kann diese Funktion nicht in jedem Fall von einfachen Spreading-Activation-Netzwerken implementiert werden, sondern erfordert eine versteckte Schicht und den Backpropagation-Lernalgorithmus. Ein Ergebnis der Analyse der Spreading-Activation-Netzwerke ist die Hypothese, dass sie die Komplexität des Retrieval-Prozesses nicht adäquat abbilden können. Die Argumentation von Creput/Caron 1997 stützt diese These.

Die von Creput/Caron 1997 vorgeschlagene Variante des Backpropagation-Verfahrens orientiert sich allerdings sehr an klassischer Logik und weniger an den Fähigkeiten neuronaler Netze. Creput/Caron 1997 trainieren durch einmalige Präsentation aller Beispiele, wobei der Algorithmus bei Bedarf neue versteckte Neuronen hinzufügt. Dabei werden die Trainingsbeispiele exakt ins Netz gespeichert. Dadurch leidet möglicherweise die Generalisierungsfähigkeit. Unklar ist auch, wie das Netz mit widersprüchlichen Trainingsbeispielen umgeht. Ein Standard-Backpropagation-Netzwerk zeichnet sich durch Fehlertoleranz aus. Ein falsch zugeordnetes Beispiel wird durch die zahlreiche Präsentation richtiger Beipiele gewissermaßen überschrieben, was bei Creput/Caron 1997 nicht möglich ist.

4.6.3 Transformations-Netzwerk

Crestani/van Rijsbergen 1997 erkennen als Schwäche der Spreading-Activation-Netze das Fehlen von sub-symbolischen Repräsentationen. Damit sei das konnektionistische Paradigma nicht ausgeschöpft. Die Autoren stellen ein Modell für adaptives Information Retrieval vor, das über sub-symbolische Repräsentation von Wissen verfügt. Ihr Modell beinhaltet ein typisches Backpropagation-Netz mit einer versteckten Schicht, die keine symbolische Deutung zulässt. Die Input-Schicht repräsentiert die Anfrage-Terme und die Output-Schicht die Dokument-Terme. Das Modell entspricht damit im We-

sentlichen dem Anfrage-Dokument-Profil-Modell, das im Rahmen des COSIMIR-Modells als eine Möglichkeit vorgestellt wird, Information Retrieval mit einem Backpropagation-Netzwerk zu integrieren (cf. Abschnitt 6.2.3) und das als Transformations-Netzwerk für die Heterogenitätsbehandlung eingesetzt wird (cf. Abschnitt 5.3.4).

Dieses Modell lernt optimale Kombinationen von Anfragen und Dokumenten und wendet das daraus gewonnene Wissen beim Retrieval auf nicht bekannte Anfragen an. Das Hauptproblem dieses Ansatzes besteht darin, dass in der Ausgabe-Schicht immer nur ein Dokument repräsentiert wird. Beim Training werden aus einer Anfrage mit n relevanten Dokumenten daher n Trainingbeispiele mit der gleichen Anfrage. Das Netz erhält so allerdings widersprüchliche Informationen, da der gleiche Input in Kombination mit verschiedenen Output-Werten trainiert wird. Im Retrieval-Fall findet das Netz auch immer nur eine Dokument-Repräsentation, was auf keinen Fall befriedigend ist. Dieses optimale Dokument ist in der Kollektion kaum vorhanden.

Crestani/van Rijsbergen 1997 versuchen beide Probleme zu lösen. Als alternatives Lernverfahren berechnen sie aus allen relevanten Dokumenten zu einer Anfrage einen Cluster-Repräsentanten und benutzen diesen zum Lernen. Sie nennen dies *horizontal learning*, während das Lernen mit allen Paaren von relevanten Dokumenten und Anfragen als *total learning* bezeichnet wird. Daneben testen die Autoren auch eine Zwischenstufe, das *vertical learning*, bei dem nur eine Untermenge der relevanten Dokumente als gewünschter Output für eine Anfrage eingesetzt werden. Letzteres Verfahren führte zu den besten Ergebnissen.

Beim Retrieval gehen Crestani/van Rijsbergen 1997 den umgekehrten Weg. Sie benutzen die vom Netz gefundene Dokument-Repräsentation wiederum als Anfrage und schicken sie an ein herkömmliches IR-System, das die Ähnlichkeit aller Dokumente mit dieser Anfrage berechnet und die ähnlichsten als Ergebnis zurückgibt. Die Output-Schicht repräsentiert während des Trainings also Dokumente, während des Recalls wird sie als Anfrage interpretiert. Während das Netz sub-symbolisch arbeitet, fallen im Gesamtsystem Schritte an, die mit herkömmlichen mathematischen Mitteln gelöst werden. Ein weiterer Schwachpunkt besteht in der binären Repräsentation der Dokumente. Die bisherige IR-Forschung hat gezeigt, dass gewichtete Repräsentationen in der Regel bessere Ergebnisse bringen. Sie lassen sich auch problemlos in den Ansatz von Crestani/van Rijsbergen 1997 integrieren.

Crestani/van Rijsbergen 1997 testen ihr System mit einem Ausschnitt der Cranfield-Kollektion, mit der auch das COSIMIR-Modell evaluiert wird (cf. Abschnitt 7.1). Sie benutzen die kleinere Cranfield-I Variante mit 200 Dokumente und 42 Anfragen mit Relevanzurteilen. Da die Autoren nur die in den

Anfragen und Dokumenten vorkommenden Deskriptoren benutzen, können sie ihr Netz auf 195 Input-Neuronen (Anfrage-Terme) und 1142 Output-Neuronen (Dokument-Terme) beschränken. Die versteckte Schicht besteht aus 100 Units. Dies führt zu einem Netz mit über 100.000 Verbindungen. Demgegenüber stehen bei *horizontal learning*, das die bessere Ergebnisse erbrachte, nur 42 Muster für Training und Test zur Verfügung. Selbst wenn alle Dokumente für alle Anfragen relevant wären, stünden bei auch *total learning* nur 800 Muster zur Verfügung. Von den Mustern nutzen die Autoren maximal 30% für das Training. Nach der Faustregel von Bigus 1996, sollte die Trainingsmenge für jede Verbindung im Netz mindestens zwei Beispiele enthalten. Bei Crestani/van Rijsbergen 1997 ist das Verhältnis Trainingsbeispiels zu Verbindungen eins zu 1000. Das Netzwerk ist also sehr stark unterspezifiziert. Die Ergebnisse sind besser als man bei dieser Situation erwarten könnte und sollten mit Vorsicht betrachtet werden. Eine Übertragung der Ergebnisse auf andere Datenmengen ist bei dieser Grundlage nicht möglich.

Die Zahl der Trainingszyklen beträgt konstant 300. Üblich ist es, das Training zu beenden, wenn die Generalisierungsfähigkeit in der Testmenge zu steigen beginnt (cf. Abschitt 3.5.4 ff.).

Die Experimente für *vertical learning* führen zu den besten Werten. Dabei bilden ein oder zwei Drittel der relevanten Dokumente zu einer Anfrage die Muster. Input ist die Anfrage und Output die verschiedenen Dokumente. *Vertical learning* übertraf den Standard-Ansatz. Dabei muss bedacht werden, dass dem lernenden Verfahren mit den Relevanzurteilen mehr Wissen zur Verfügung steht als dem Standard-IR-Verfahren. Dagegen ergab sich weder eine Verbesserung für *horizontal learning*, wobei ein Cluster-Repräsentant als Output dient, und *total learning*, bei dem alle relevanten Dokumente ins Training einfließen.

Interessant an den Experimenten ist, dass die adaptierten Anfragen sich teilweise sehr deutlich von den ursprünglichen Formulierungen unterschieden. In vielen Fällen waren Terme ganz verschwunden. Weiterhin überschneiden sich die Treffermengen kaum. D.h. beide Anfragen finden relevante Dokumente, aber jedes findet andere. Dies deckt sich mit den Ergebnissen der TREC-Konferenz, nach denen verschiedene Information Retrieval Systeme bei vergleichbarer Gesamtqualität unterschiedliche relevante Dokumente in der Ergebnismenge liefern (cf. Abschnitt 2.1.4.2 und Abschnitt 2.3.1.2, cf. Womser-Hacker 1997).

Crestani/van Rijsbergen 1997 haben auch Relevanz-Feedback in ihrem Netz realisiert. Aus den relevanten Dokumenten in der Ergebnismenge werden Terme extrahiert, die zur Anfrage hinzukommen. Diese Muster bestehend aus Original-Anfrage als Input und adaptierter Anfrage als Output lernt das Netz

zusätzlich. Die Ergebnisse sind schlechter als bei probabilistischem Relevanz-Feedback.

Auch wenn die Ergebnisse nicht ohne weitere Tests auf größere Textkollektionen übertragbar sind, so ist sehr vielversprechend, dass die modifizierten Anfrage relevante Dokumente finden, die das Vergleichssystem nicht findet. Kurzfassungen des Ansatzes und der Ergebnisse finden sich in Crestani 1993, 1993a, 1994, 1994a und 1995.

Cortez et al. 1995 schlagen ein Transformations-Netzwerk mit fast identischer Funktionalität vor. Ein Backpropagation-Netzwerk bildet die Anfrage-Terme auf die Index-Terme ab, wobei die Schichten allerdings unterschiedlich groß sind. Ein induktiver Lernalgorithmus analysiert vor dem Training die Dokumente und extrahiert die signifikantesten und am stärksten diskriminierenden Terme, die dann den Output bilden. Ein kleines Experiment mit diesem Verfahren nutzt die ADI-Kollektion mit 35 Anfragen und 82 Dokumenten, von denen die Titel vorliegen. Am Input wurden die 116 in den Anfragen vorkommenden Terme und in der Output-Schicht die 66 signifikanten Dokument-Terme angelegt. Die gesamte Kollektion wird zum Training des Backpropagation-Netzes und eine Teilmenge zum Test benutzt, wobei aus den Original-Anfragen durch das zufällige Entfernen von Termen unvollständige Fragen gebildet wurden. Die unvollständigen Anfragen aktivieren im Output 83% der Dokument-Terme, die sie mit den vollständigen Anfragen gelernt hatten. Aussagen zur Qualität des Retrievals erfolgen nicht.

Das Transformations-Netzwerk besitzt großes Potenzial im Information Retrieval. Dies muss weiter experimentell überprüft werden, insbesondere für die Behandlung von Heterogenität im Information Retrieval (cf. Abschnitt 5.3.4). Während das Netz für den Einsatz als IR-System ungeeignet ist und bei Crestani/van Rijsbergen 1997 zur Ausweichstrategien *total* und *horizontal learning* als führt, ist die Architektur für die Heterogenitätsbehandlung sehr gut einsetzbar. Das Transformations-Netzwerk ist ein vielversprechender hetero-assoziativer Ansatz, der auf dem Backpropagation-Algorithmus aufbaut. Allerdings sind weitergehende Tests mit realen Daten nötig.

4.7 Weitere Information Retrieval Modelle mit neuronalen Netzen

Während neuronale Netze in den oben ausführlich diskutierten Modellen eine zentrale Rolle spielen, decken sie in einigen Systemen nur Teilaspekte des Retrievals ab oder werden mit anderen Verfahren kombiniert. Der Vollständigkeit halber stellt dieser Abschnitt einige der wichtigsten Systeme vor. Die

assoziativen Thesauri und Systeme zur Term-Erweiterung auf der Basis von Hopfield-Netzen behandelt Abschnitt 4.2.

4.7.1 Neuronale Netze und Genetische Algorithmen

Neben der Fuzzy Logik und den neuronalen Netzen haben sich auch die genetischen Algorithmen als intelligente Wissensverarbeitungstechnik etabliert. Diese drei Methoden gehören zum Paradigma Soft Computing (cf. Kapitel 2) und werden inzwischen oft unter dem Begriff *Computational Intelligence* zusammengefasst (cf. z.B. Zimmermann 1998). Durch diese Abgrenzung vom Begriff *Artificial Intelligence* (Künstlichen Intelligenz, KI) wird versucht, die Debatte zu vermeiden, inwieweit intelligente Informationssysteme die intelligenten Fähigkeiten des Menschen implementieren oder nur teilweise imitieren.

Genetische Algorithmen beruhen wie die neuronalen Netzen auf einer Metapher aus dem Bereich der Biologie, nämlich auf dem Prinzip *survival of the fittest*, stehen jedoch der klassischen Künstlichen Intelligenz näher als der Konnektionismus. Eine Einführung in genetische Algorithmen bieten Beasley et al. (1993a, b).

Genetische Algorithmen sind heuristische Suchverfahren, die ähnlich wie andere Suchverfahren in der Künstlichen Intelligenz (cf. Rich/Knight 1991) eine Menge von möglichen Lösungen zu einem Problem generieren und diese dann auf ihre Qualität hin testen. Lösungen, die gut bewertet werden, berücksichtigt das System dann weiterhin. Das Grundidee genetischer Algorithmen besteht darin, die Eigenschaften dieser guten Lösungen an eine neue Menge von Lösungen weiterzugeben. Dabei werden die Eigenschaften neu kombiniert und in geringem Maße zufällig geändert. In der Terminologie der genetischen Algorithmen ist die Menge von Lösungen eine Population, die Qualität hinsichtlich einiger Probleme ist die *fitness* in einer Umwelt, Eigenschaften heißen Gene bzw. Chromosomen, die Weitergabe von Eigenschaften an die nächste Generation ist in der Metapher die Fortpflanzung, die mit einer gewissen Mutation verbunden ist. Wie in der Natur können sich nur gut angepasste Individuen fortpflanzen und ihre Gene in die nächste Generation einbringen. Bei einer erfolgreichen Anwendung konvergiert eine Population, d.h. ihre Mitglieder weisen hohe Qualitätswerte auf. Der beste Genotyp liefert die Lösung des Problems (cf. Beasley et al. 1993a). Genetische Algorithmen werden meist für Optimierungsprobleme eingesetzt. Sie weisen prinzipiell ähnliche Vor- und Nachteile auf wie andere heuristische Suchalgorithmen. Insbesondere durchsuchen sie den Suchraum möglicher Lösungen nicht vollständig und finden deshalb auch nicht unbedingt eine optimale Lösung. Der

Erfolg von genetischen Algorithmen hängt stark von einer guten *fitness*-Funktion ab, die vergleichbar ist mit der heuristischen Schätzfunktion im A*-Algorithmus (cf. Rich/Knight 1991:76). Auch KI-Suchverfahren, wie sie in einem typischen Schachprogramm zu finden sind, hängen stark von der Bewertungsfunktion für die berechneten Zustände ab.

Genetische Algorithmen können mit neuronalen Netzen zu hybriden Systemen kombiniert werden. Besonders für die Suche nach einer günstigen Architektur neuronaler Netze eignen sich genetische Algorithmen.

Chen 1994 kombiniert die beiden Verfahren in seinem System GANNET (Genetic Algorithms and Neural Nets System). Dabei bilden die Dokument-Vektoren die Gene. Ein Benutzer wählt fünf Dokumente mit für ihn großer Relevanz. Der genetische Algorithmus optimiert dann die Ähnlichkeit der Dokumente, wobei als *fitness*-Funktion ein Standard-Ähnlichkeitsmaß dient. Dadurch verändert sich die Repräsentation der Dokumente. Die *fitness* in der ausgewählten Menge steigt. Die in der optimierten Menge enthaltenen Terme aktivieren in einem Hopfield-Netzwerk (cf. Chen 1995, cf. Abschnitt 4.2) weitere Terme. Diese assoziierten Terme sind Grundlage einer Suche mit Standardverfahren. Eine begrenzte Anzahl der Ergebnisdokumente wird in die Grundmenge aufgenommen und der genetische Algorithmus optimiert wiederum die Repräsentation dieser Menge. Durch mehrere Zyklen kann die gesamte *fitness* in der Menge erhöht werden.

Für den Retrievalprozess ist dies aber von eingeschränktem Wert. Die Situation des Benutzers in GANNET entspricht nicht der eines Informationssuchenden, sondern der eines Juroren, der versucht, die Repräsentationen zu verbessern. Damit bietet sich GANNET eher für den Indexierungsprozess an. Dabei ist aber nicht klar, wie sich die Änderung der Repräsentationen in den kleinen, intellektuell bestimmten Mengen auf die Gesamtleistung des Systems auswirkt.

4.7.2 Benutzermodellierung

Neuronale Netze werden auch zur Benutzer-Modellierung und insbesondere als personalisierte Filter eingesetzt. Damit besteht eine enge Beziehung zum Information Retrieval. Einen Überblick über Benutzermodellierung bietet Roppel 1996.

Chen/Norico 1992 beschreiben ein System, das auf dem Backpropagation-Algorithmus beruht.

Jennings/Higuchi 1992 stellen ein nicht-überwacht lernendes Netz vor, das einen News-Service personalisiert. Pelletier et al. 1996 realisieren ein ähnli-

ches Netz, das sich wie viele Spreading-Activation-Netzwerke auf die Analogie zum probabilistischen Retrieval beruft.

4.8 Neuronale Netze bei TREC

Eine Messlatte für experimentelle und kommerzielle Information Retrieval Systeme ist die Text Retrieval Conference (TREC). In TREC können Forscher ihre Systeme an einem realistischen Korpus bei standardisierten Bedingungen testen (cf. Abschnitt 2.1.4.2).

An TREC nahmen von Anfang an einige Systeme teil, die auf neuronalen Netzen beruhen (z.B. in TREC I: Gallant et al. 1993, Kwok et al. 1993). Insgesamt nahmen vier Systeme teil, davon gehören drei zum Spreading-Activation-Ansatz. PIRCS (cf. Abschnitt 4.3.2.1, cf. Kwok/Grunfeld 1994/6, Kwok et al. 1999) und Mercure (cf. Abschnitt 4.3.2.4, cf. Boughamen/Soule-Dupuy 1997/8, Boughamen et al. 1999) haben beide bei ihrer ersten Teilnahme die kleinere Textmenge (Kategorie B) bearbeitet und haben dann innerhalb eines Jahres den Umstieg zur vollen Textmenge (Kategorie A) geschafft. Dies zeigt, dass diese Modelle ausgereift sind und auch größere Textmengen bewältigen. Die folgende Tabelle gibt einen Überblick über die eingesetzten neuronalen Netzwerk Systeme:

Tabelle 4-2: Systeme mit neuronalen Netzen bei TREC

Konferenz	**Ad-hoc Experimente**	**Routing/Filtering Experimente**	**Weitere Experimente**
TREC 1	PIRCS, MatchPlus	PIRCS	
TREC 2	PIRCS, MatchPlus	PIRCS, Boyd et al., MatchPlus	
TREC 3	PIRCS	PIRCS	
TREC 4	PIRCS	PIRCS	
TREC 5	PIRCS, Mercure	PIRCS, Mercure	PIRCS (Chinese)
TREC 6	PIRCS, Mercure	PIRCS, Mercure	Mercure (Cross-lingual), PIRCS (Chinese, High Precision)
TREC 7	PIRCS, Mercure	PIRCS, Mercure	PIRCS (High Precision)

Ein Vorteil der Spreading-Activation-Netze PIRCS und Mercure besteht in der gleichberechtigten Behandlung von Dokumenten und Anfragen. Die TREC Routing-Experimente ergeben sich daher ganz natürlich. Beim Routing liegen Interessensprofile vor, zu denen aus einem Strom von Dokumenten die relevanten Texte gefiltert werden. Das Modell von Kwok/Grunfeld 1994 und 1996 benutzt die neuen Dokumente als Input. Sie aktivieren die für sie relevanten Anfragen. PIRCS (Kwok/Grunfeld 1994/6) lernt darüber hinaus anhand von Relevanz-Feedback durch die Veränderung von Gewichten. Mercure erweitert die Anfrage durch Spreading-Activation (Boughamen/Soule-Dupuy 1997/8).

Boyd et al. 1994 (cf. auch Syu/Lang 1994) versuchen, semantisches Wissen aus einem Thesaurus in ihr Spreading-Activation-Netz zu integrieren. Der Thesaurus enthält thematische Rollen für Wörter wie z.B. *Beneficiary* oder *Cause*. Für die Category B Routing konnten die semantischen Experimente allerdings nicht durchgeführt werden. Das als Vergleichsmaßstab vorgesehene Standard-Spreading-Activation-Netz wurde zur TREC-Konferenz eingereicht. Die Ergebnisse sind nach Angaben der Autoren schlecht.

Der in MatchPlus eingesetzte Ansatz von Gallant et al. 1993 und 1994 ist nicht vollständig offengelegt, da die Firma HNC ihn kommerziell einsetzt. Daran wirken zwei neuronale Netze, wobei es sich höchstwahrscheinlich um Spreading-Activation-Ansatz und/oder Self-Organizing Maps handelt, da offensichtlich keine Daten für überwachtes Lernen benötigt werden. Das Modell verwendet sogenannte Kontext-Vektoren. Das Vektorraum-Modell im Information Retrieval verwendet Vektoren, in denen jede Komponente einen Deskriptor repräsentiert. Dadurch entstehen sehr lange Vektoren, von denen viele den Wert Null besitzen, die also spärlich besetzt sind. Je länger die Vektoren, desto höher ist die Rechenzeit für das Retrieval. Kontext-Vektoren sind ein semantischer Ansatz zur Dimensionalitätsreduktion (cf. Abschnitt 2.1.2.4.1). Ziel von Reduktionen ist nicht nur eine schnellere Verarbeitung der Muster, sondern auch bessere Verteiltheit der Muster und damit bessere Ähnlichkeitswerte. Die Komponenten in Kontext-Vektoren repräsentieren semantische Eigenschaften wie *human, day* oder *heavy*. Diese Eigenschaften werden entweder intellektuell oder aus hochfrequenten Begriffen gewonnen. Im Vektor für einen Term nehmen die Komponenten Werte an, die den Term semantisch beschreiben. Der Deskriptor *astronomer* hätte z.B. in der Komponente *human* den höchsten Wert und in der Komponente *day* einen niedrigen Wert. Dies kann intellektuell geschehen, Gallant et al. 1993 und 1994 berichten von Algorithmen, die Kontext-Vektoren halb- und vollautomatisch erstellen. Kontext-Vektoren repräsentieren also zunächst Deskriptoren. Dokumente und Anfragen ergeben sich als die gewichtete Summe der enthaltenen Einzelterme.

Einschränkungen von TREC haben sich wahrscheinlich auf die Gestaltung der teilnehmenden Systeme ausgewirkt. Sowohl PIRCS als auch Mercure haben an den Basistests teilgenommen. Relevanz-Feedback spielt lediglich in einem anderen Teil von TREC, dem Interactive Track, eine Rolle. Möglicherweise verzichten die beiden Systeme in ihren TREC Experimenten deshalb auf das den Spreading-Activation-Modelle inhärente Relevanz-Feedback.

Die Bewertung in TREC erfolgt mit den Maßen Recall und Precision. Die Precision wird für Recall-Niveaus zwischen 0,1 bis 0,9 berechnet und aus diesen Werten die durchschnittliche Precision bestimmt. Diese Zahl führt zu den fünf besten Systemen, die jedes in einem Überblicksartikel erwähnt werden. Eine Grafik zeigt die Recall-Precision-Kurve dieser Systeme in Recall-Schritten von 0,1 zwischen 0,1 bis 0,9.

In TREC 5 zählte PIRCS sowohl bei Ad-hoc mit der Kurzbeschreibung und automatischer Anfragen-Generierung, bei Ad-hoc mit manueller Anfrage-Generierung als auch bei den Routing-Experimenten zu den besten acht Systemen (cf. Voorhees/Harman 1997a). Beim Chinese-Track erzielte PIRCS das beste Ergebnis (cf. Wilkinson 1998) unter zehn Teilnehmern (cf. Smeaton/Wilkinson 1997).

In TREC 6 liegen PIRCS und Mercure in der Spitzengruppe der ad-hoc Systeme. Unter den 57 eingereichten Ergebnissen mit automatischer Anfragen-Generierung nutzten sechzehn die vollständige Topic-Beschreibung, 29 die Kurzbeschreibung und zwölf nur den Titel. Bei den Titeln und Langbeschreibungen gehörten PIRCS und Mercure jeweils zu den acht besten Systemen (cf. Voorhees/Harman 1998a). Dabei ist PIRCS besser als Mercure, allerdings liegen die acht ersten Systeme sehr eng zusammen und der Unterschied ist somit gering. Bei den Routing-Experimenten konstruierten 28 Systeme die Anfrage automatisch. PIRCS und Mercure zählen auch hier wieder zu den besten acht Systemen. Beim Chinese-Track erreichte PIRCS wie im Jahr vorher ein sehr gutes Ergebnis (cf. Wilkinson 1998). Im High Precision Track hatte PIRCS mit die schlechtesten Ergebnisse, erstellte aber als einziges System die Anfrage automatisch (cf. Buckley 1998).

In TREC 7 findet sich PIRCS nach wie unter den acht besten Systemen der ad-hoc-Systeme mit automatischer Anfragengenerierung, Mercure jedoch nicht mehr (cf. Voorhees/Harman 1999a).

Bei den Information Retrieval Systemen aus der Familie der neuronalen Netze bei TREC handelt es sich somit um Spreading-Activtion-Netzwerke. Insbesondere PIRCS zeigt die Leistungsfähigkeit dieser Systeme. Es nahm an allen bisherigen TREC-Konferenzen teil und erreichte häufig einen der Spitzenplätze. Mercure gelang bei drei Teilnehmen einmal ein Vorstoß in die Gruppe der acht besten Systeme einer Kategorie. Boyd et al. 1994 versuchen, ein se-

mantisch angereichertes Spreading-Activation-Netzwerk zu implementieren. Die Ergebnisse fielen allerdings schlecht aus. Andere Modelle als Spreading-Activation-Netzwerke spielen praktisch keine Rolle. Der Algorithmus von MatchPlus ist nicht offengelegt und nahm nur an den beiden ersten Konferenzen teil.

4.9 Fazit: Neuronale Netze im Information Retrieval

Von allen analysierten Klassen von Modellen werden drei für große Mengen von realen Daten eingesetzt:

- Assoziativspeicher wie z.B. Hopfield-Netzwerke (cf. Abschnitt 4.2) sind eine sehr interessante Klasse von neuronalen Netzen, deren Grundfunktionalität im Retrieval besteht. Auto-Assoziativspeicher liefern gespeicherte Muster zu unvollständigen Input-Mustern. SpaCAM ist ein Beispiel für die erfolgreiche Adaption und Anwendung dieser Technik im kommerziellen Umfeld. In Reinform lässt sich damit allerdings kein IR-Prozess implementieren. Hetero-assoziative Netze können nicht nur in einem Hopfield-Netzwerk implementiert werden, sondern auch in einem Backpropagation-Algorithmus oder einer Kohonen-Karte realisiert werden. Die Spreading-Activation-Netzwerke sind somit ein Sonderfall eines hetero-assoziativen Hopfield-Netzwerks.
- Spreading-Activation-Netzwerke (cf. Abschnitt 4.3) wurden bisher am häufigsten eingesetzt und mit großen Dokumentmengen evaluiert. Sie stellen nur bedingt eine eigene konzeptuelle Klasse von IR-Verfahren dar, sondern lehnen sich eng an die bekannten Modelle an. Nur mehrfache Aktivierungsschritte zwischen den Schichten führen zu eigenständigen Ergebnissen.
- Kohonen-SOM und ART-Netze (cf. Abschnitte 4.4 und 4.5) sind die Grundlage von Systemen mit großen Datenmengen. Allerdings handelt es sich um Clustering-Verfahren, die für wichtige Schritte im Information Retrieval nicht direkt anwendbar sind. Für den Vergleich zwischen Dokument und Anfrage sind Clustering-Verfahren nicht geeignet.

Alle diese Systeme haben folgende Nachteile:

- Fehlen von sub-symbolischer Verarbeitung:
  Diese Systeme nutzen die Mächtigkeit neuronaler Netze nicht aus. Der sehr häufig benutzte und mächtige Backpropagation-Algorithmus wird nicht eingesetzt.

- Lernfähigkeit bleibt weitgehend eingeschränkt:
  Wenn Lernen implementiert ist, beschränkt es sich meist auf Relevanz-Feedback. Die Auswirkungen für das gesamte Modell werden nicht quantifiziert und sind wohl minimal.
- Das am intensivsten erforschte Modell der Spreading-Activation-Netzwerke stellt keine konzeptuell neuartige Klasse von IR-Modellen dar.

Information Retrieval Modelle auf der Basis des Backpropagation-Algorithmus erscheinen sehr erfolgversprechend. Von den bestehenden Modellen ist besonders das Transformations-Netzwerk interessant. Es eignet sich für die Heterogenitätsbehandlung (cf. Abschnitt 5.3.4) und zur Vorverarbeitung der Anfrage, erfordert aber noch ein vollständiges Retrievalsystem. Das Transformations-Netzwerk wird in Abschnitt 7.2 mit einer realen Datenbasis evaluiert.

Ein neues Modell für Information Retrieval auf der Basis neuronaler Netze sollte die diskutierten Schwächen vermeiden. Damit gelten vor allem die folgenden Zielvorgaben, die sich aus einer stärkeren Berücksichtigung der Stärken neuronaler Netze ergeben:

- Die sub-symbolischen Fähigkeiten des Backpropagation Ansatzes erlauben die Implementierung einer großen Anzahl Klassen von Funktionen.
- Die Lernfähigkeit sollte stark ausgeprägt sein.
- Die im IR bewährten üblichen Indexierungsverfahren, Repräsentationen und Gewichtungsschemata sollten benutzt werden können.
- Der Kern des IR-Prozesses sollte auch den Schwerpunkt der Modellierung bilden.

Kapitel 6 stellt das COSIMIR-Modell vor, das auf den Ergebnissen dieser Analyse des state-of-the-art aufsetzt.

Keines der neuronalen Modelle für Information Retrieval implementiert ein neues Verfahren zur Inhaltsanalyse, sondern sie alle setzen eine der bestehenden Inhaltserschließungsmethoden voraus und setzen erst auf der gewonnenen Dokument-Term-Matrix auf. Wie Abschnitt 2.1 zeigt, sind diese vorhandenen Verfahren zwar unbefriedigend, da sie die Syntax von Texten weitgehend ignorieren und die Semantik nur durch die Isolation der im Text vorkommenden Wörter modellieren. Da jedoch keine effizienten semantischen Analyseverfahren für Massendaten zur Verfügung stehen, werden die im Information Retrieval üblichen Extraktions- und Frequenzanalyse-Algorithmen bei der

Inhaltsanalyse nach wie vor bevorzugt. Dementsprechend ist auch das dazu gehörende Konzept der Dokument-Term-Matrix die übliche Art der Wissensrepräsentation.

Davon weicht das COSIMIR-Modell nicht ab. Die Schwächen der Inhaltsanalyse werden aufgrund fehlender Alternativen akzeptiert und die Dokument-Term-Matrix bildet den Ausgangspunkt von COSIMIR.

5 Heterogenität und ihre Behandlung im Information Retrieval

Dieses Kapitel greift mit der Heterogenität einen Aspekt des Information Retrieval auf, den bereits Kapitel 2 erwähnt. Nach der Diskussion neuronaler Netze (cf. Kapitel 3) und ihrer Rolle im Information Retrieval (cf. Kapitel 4) steht bei der Problematik der Heterogenität und ihrer Behandlung die Anwendung vager Verfahren im Zentrum. Gleichzeitig ermöglicht dieses Kapitel das Verständnis einer Erweiterung des COSIMIR-Modells, das im nächsten Kapitel vorgestellt wird. Diese Erweiterung erlaubt die Verarbeitung heterogener Daten ohne eine explizite Transformation und wird in Kapitel 7 wie das COSIMIR-Modell und das Transformations-Netzwerk getestet.

Dieses Kapitel führt zunächst die Dimensionen und Probleme der Heterogenität vor und zeigt dann Lösungsansätze auf, die vorwiegend im Bereich der vagen Informationsverarbeitung liegen. Wichtig sind besonders Transformationen zwischen heterogenen Repräsentationen, wobei der Schwerpunkt auf der semantischen und nicht auf der technischen Integration liegt.

5.1 Probleme und Dimensionen der Heterogenität

Die zunehmende weltweite Vernetzung schafft für die Informationssuche völlig neue Möglichkeiten. Unterschiedlichste Datenquellen werden virtuell integriert und ein System kann auf diese Weise sehr große Mengen von Daten anbieten. Gleichzeitig stellen diese Möglichkeiten das Information Retrieval vor die Herausforderung, die Suche in stark heterogenen Umgebungen zu erleichtern.

Die Trennung der Informationsangebote und ihre Heterogenität wird aus politischen und inhaltlichen Interessen bestehen bleiben, da viele Informationsanbieter sich durch ihr individuelles Angebot profilieren. Auch werden sich kaum alle Datenbankproduzenten auf eine gemeinsame Basis bei der Inhaltserschließung verständigen. Zudem führt die Aufteilung in Spezial-Datenbanken zu einer höheren Kompetenz bei der Aufarbeitung dieser Bereiche und ist somit für viele Benutzerabsichten sehr sinnvoll.

In der Praxis kennt ein Benutzer oft mehrere potenziell relevante Quellen, wie etwa einzelne Datenbanken oder Internet-Angebote, die er dann sukzessive abfragt. Der Benutzer überführt also das gleiche Informationsbedürfnis mehrfach in eine Anfrage. In der Regel entstehen so während eines Informations-Prozesses abhängig von den Zwischenergebnissen iterativ mehrere Anfragen.

Dieser gesamte Prozess wird für jedes der Informationsangebote durchlaufen, da die Zwischenergebnisse in den verschiedenen Quellen unterschiedlich sind. Somit entsteht ein erheblicher zusätzlicher Aufwand.

Bei einer erschöpfenden Suche sollten so viele Quellen wie möglich durchsucht werden. In der Praxis kann ein Benutzer aber nicht beliebig viele Datenbanken abfragen. Eine mögliche Ausweichstrategie, die automatische Auswahl von vielversprechenden Datenquellen für eine Anfrage, hat sich in der Praxis als ist sehr schwierig erwiesen (cf. Rittberger 1995, Gövert 1997).

Für die Benutzer ist es also wünschenswert, viele getrennt erfasste und verwaltete Informationsangebote mit einer Anfrage zu durchsuchen. Dieser Wunsch erklärt einen Teil des Erfolgs von Internet-Suchmaschinen wie Northern Light (cf. Abschnitt 2.1.5). Suchmaschinen liefern Daten verschiedenen Typs, wenn diese innerhalb einer Seite im Format HTML (Hypertext Markup Language) kodiert sind. Diese beinhalten Bilder, Filme, Tabellen und meist Text, wobei verschiedenste natürliche Sprachen vorkommen. Die technischen Probleme der Heterogenität beim Internet löst weitgehend die gemeinsame Basis HTML. Gerade diese Chance zur Heterogenität gilt als ein Grund für den großen Erfolg des Internet, das nur die Darstellungssprache HTML vorgibt, aber keine standardisierte Inhaltserschließung oder -beschreibung (cf. z.B. Kuhlen 1999:138).

Vor allem entstehen durch die Heterogenität aber semantische Probleme, die bis heute weitgehend ungelöst sind und von bestehenden Systemen wie den Internet-Suchmaschinen ignoriert werden. Das Grundproblem sind unterschiedliche Begriffs-Schemata, die unterschiedliche Daten repräsentieren. Häufig erfordern Fachgebiete unterschiedliche Spezialthesauri, da Fachtermini unterschiedlich gebraucht werden. Je nach Kontext ändern Begriffe ihre Semantik. Ein spezieller Thesaurus verspricht daher eine bessere Retrievalqualität in Fachgebieten, in denen die Semantik eines Terms klar definiert ist und konsistent benutzt wird.

Bestrebungen, die Heterogenität durch Standardisierung zu lösen, haben wenig Aussicht auf Erfolg. Dazu gehört die immer wieder geforderte Verwendung von weltweiten Thesauri zur Verschlagwortung von Internet-Dokumenten durch die Autoren. Es ist fragwürdig, ob Autoren überhaupt eine Indexierung vornehmen. Wenn ja, dann haben sie in der Regel keine dokumentarische Ausbildung und kennen nicht alle den gesamten Thesaurus, um ihr Dokument richtig einzuordnen. Zudem spielen wirtschaftliche Interessen bei der Auffindbarkeit von Dokumenten eine große Rolle und beeinflussen die Vergabe von Termen.

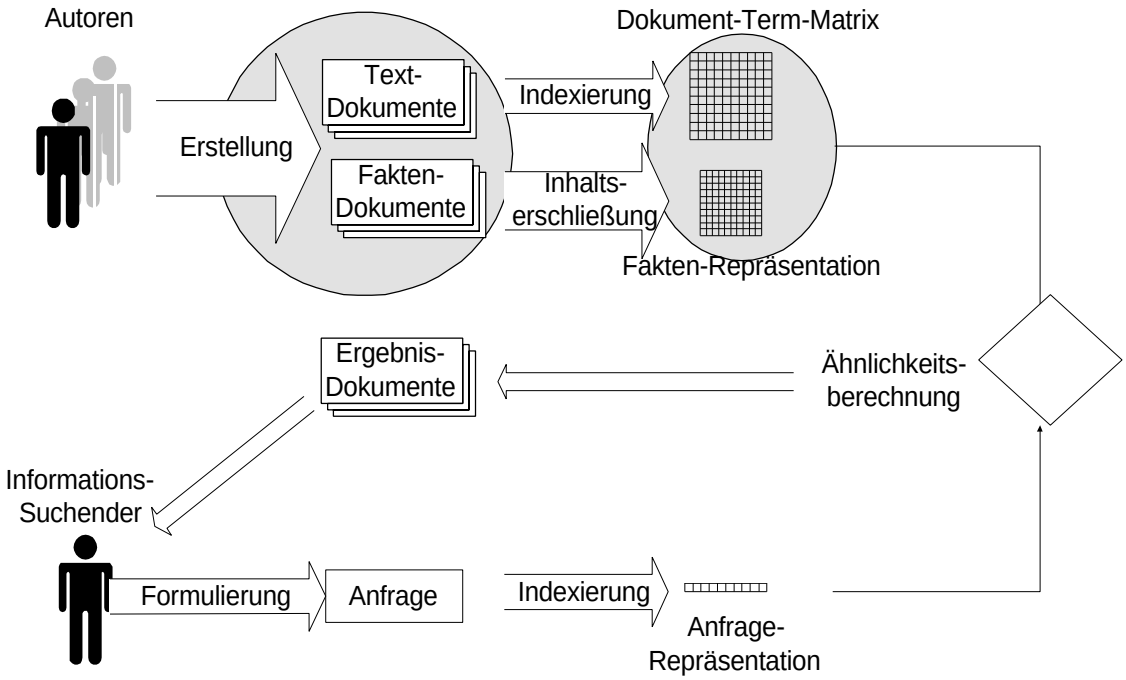


Abbildung 5-1: Der Information Retrieval Prozess bei heterogenen Datenbeständen (hier Texte und Fakten) und heterogenen Repräsentationen

Semantische Probleme treten auch bei Wortlisten aus der automatischen Indexierung auf. Ein Term tritt in verschiedenen Korpora mit unterschiedlichen Verteilungshäufigkeiten auf. Aus Sicht des Information Retrieval, in denen sich Bedeutung auf Vorkommenshäufigkeit reduziert, ändert sich dadurch die Bedeutung. Die Bedeutung eines Terms konstituiert sich aus den Dokumenten, auf die er verweist. Damit hat ein Term in jeder Kollektion eine unterschiedliche Bedeutung. Eine automatische Analyse aller Texte ist organisatorisch unmöglich und auch nicht unbedingt sinnvoll, da empirische Untersuchungen gezeigt haben, dass in unterschiedlichen Kollektionen unterschiedliche Varianten der Inhaltserschließung zur optimalen Retrievalqualität führen (cf. Womser-Hacker 1997:212 ff.). Dies ist plausibel, da ein Begriff in verschiedenen Fachgebieten oft unterschiedlich gebraucht wird.

Das Kernproblem der Heterogenität besteht also darin, Abbildungen zwischen verschiedenen Begriffs-Schemata - seien es Thesauri oder Wortlisten aus der automatischen Indexierung – zu erreichen. Diese Probleme gewinnen im Kontext digitaler Bibliotheken und unternehmensweiter Data-Warehouses verstärkt an Bedeutung.

Abbildung 5-1 zeigt den Information-Retrieval-Prozess bei heterogenen Quellen. Die Situation des Benutzers ist nach wie vor identisch mit der im

Standardfall (cf. Abbildung 2-1), das System muss die Anfrage jetzt aber mit heterogenen Dokumenten vergleichen. Dadurch ergeben sich die oben besprochenen Problembereiche, die in Abbildung 5-1 mit Ellipsen markiert sind.

Eine prägnante Zusammenfassung der semantischen Probleme der Heterogenität gibt Chen 1998. Zahlreiche aktuelle Beispiel für den Bereich unterschiedlich erschlossener Textdaten liefern Buckland et al. 1999. Viele weitere Arbeiten befassen sich mit der Lösung technischer Probleme bei der Integration. So untersuchen z.B. Gövert 1996 und Fuhr 1999 die logische und konzeptuelle Verbindung verschiedener Datenschemata.

Heterogenität entsteht aufgrund verschiedener Ursachen. Im den folgenden Abschnitten werden die Probleme heterogener Objekte, Qualität und Sprachen thematisiert. Diese voneinander unabhängigen Dimensionen von Heterogenität zeigt Abbildung 5-2.

5.1.1 Heterogene Objekte

Information Retrieval Systeme waren lange Zeit auf Text beschränkt. Wie Abschnitt 2.1 zeigt existieren inzwischen Systeme für verschiedenste Datentypen. Benutzer wissen zu Beginn eines Informationsprozesses häufig nicht, welche Art von Dokumenttyp ihnen bei der Problemlösung hilfreich sein wird. Im Vordergrund steht der Wunsch nach relevanten Objekten und nicht der formale Aspekt eines Datentyps. In der Praxis existieren heute getrennte Information-Retrieval-Systeme für Texte, Bilder, Grafiken (cf. del Bimbo 1999), Fakten oder Audio-Dokumente wie gesprochene Sprache (cf. Schäuble 1997:121ff.). Die Repräsentationen und Retrieval-Funktionen werden für einzelne Datentypen optimiert. In einem umfassenden Multimedia-Informationssystem sollte diese Schranke weitgehend aufgehoben sein und der Zugriff auf verschiedenste Datentypen mit einer Anfrage möglich sein.

Ein häufiger Anwendungsfall ist die Suche nach Fakten und Texten in einem Informationssystem. Dazu wird technisch ein Information Retrieval System in ein Datenbankmanagementsystem integriert (IR-DBMS Integration). Vage Verfahren aus dem Information Retrieval werden mit exakten Datenbank-Algorithmen kombiniert. Diese Problematik diskutieren z.B. Fuhr 1992 und Gövert 1996.

Inzwischen existieren hierfür kommerzielle Lösungen. Der Search Server der Firma FULCRUM ist ein klassisches Information-Retrieval-System, das die Textdaten in einem relationalen Schema abspeichert. Zusätzlich können Anwender weitere beliebige Tabellen mit strukturierten Fakten anlegen, die sie mit der Sprache SQL (Structured Query Language) für exakte Datenbankabfragen bearbeiten können. Die Text-Retrieval-Funktionalität ist durch eine

Erweiterung von SQL um die contains-Bedingung realisiert (cf. Krause/Mutschke 1999, PC DOCS/Fulcrum 1999). Ähnliche Konzepte bieten auch klassische Datenbankmanagementsysteme, die um Text-Retrieval-Komponenten erweitert wurden. Tabelle 5-1 bietet einen Überblick über solche Systeme.

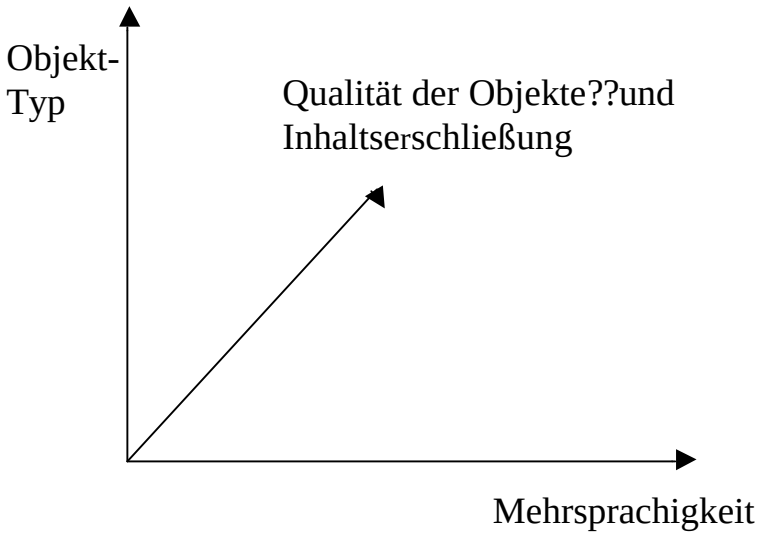


Abbildung 5-2: Dimensionen der Heterogenität

Neben den technischen Schwierigkeiten entstehen v.a. konzeptuelle Probleme. Einige werden im Folgenden am Beispiel von ELVIRA (Elektronisches Verbandsinformations-, Recherche- und Analysesystem, cf. Abschnitt 2.2.3.2) diskutiert. ELVIRA wurde bereits in als Beispiel für ein Fakten-Retrieval-System mit Erweiterungen für vages Retrieval vorgestellt und zeigt, wie Heterogenität die Vagheit erhöht.

Das größte Heterogenitätsproblem im Kontext von ELVIRA ist die Existenz verschiedener Produktnomenklaturen also Fakten-Thesauri, die jeweils einer speziellen Sichtweise auf die Elektroindustrie bzw. die gesamte Industrie entsprechen. Dies trifft bereits für das reine Fakten-Retrieval zu (cf. Abschnitt 2.2.3.2.1). Auf eine einheitliche Einteilung der Industrie konnten sich die beteiligten Verbände und Informationsanbieter nicht verständigen. Jede bestehende Nomenklatur repräsentiert Benutzersichten und ist damit gerechtfertigt. Aufgrund der Vertrautheit der Benutzer mit den bestehenden Nomenklaturen, ist es auch aus softwareergonomischer Sicht sinnvoll diese zu integrieren und so Altwissen auszunutzen. So heißen z.B. *Waschmaschinen* im Güterverzeichnis für Produktionsstatistiken *Waschvollautomaten*, womit die gleichen Produkte gemeint sind. In zahlreichen Fällen werden Produkte völlig unterschiedlich gruppiert, so dass diese Gruppen nicht vergleichbar sind (für weitere Beispiele cf. Mandl et al. 1998).

Tabelle 5-1: Datenbankmanagementsysteme mit integrierter Text-Retrieval-Funktionalität

Hersteller	**Datenbank-System**	**Text-Retrieval-Komponente**	**Ranking Verfahren**
Sybase	Adaptive Server[1]	Verity (Speciality Data Store)[2]	vorhanden
Informix	Informix Dynamic Server[3]	Verity Text DataBlade[4]	vorhanden
Informix	Informix Dynamic Server[5]	Excalibur RetrievalWare[6]	
Computer Associates	Objekt-orientierte Datenbank Jasmine[7]	Excalibur Text DataBlade Module[8]	
IBM	DB2[9]	IBM Intelligent Miner for Text[10]	vorhanden
ORACLE	ORACLE 8[11]	Oracle Intermedia[12]	vorhanden
Software AG	ADABAS[13]	ADABAS TEXT RETRIEVAL[14]	

In folgendem Beispiel, das aus einer empirischen Analyse realer Anfragen an den VDMA stammt (cf. Mandl et al. 1998), formuliert ein Benutzer seine An-

[1] http://www.sybase.com/products/databaseservers/ase

[2] http://www.sybase.com/detail/1,3693,1009236,00.html

[3] http://www.informix.com/informix/products/servers

[4] http://www.informix.com/informix/.../options/udo/datablade/dbmodule/verity1.htm

[5] http://www.informix.com/informix/products/servers

[6] http://www.excalib.com/partners/dir/informix.shtml

[7] http://www.cai.com/products/jasmine.htm

[8] http://www.informix.com/informix/products/options/udo/.../.../excalibur1.htm

[9] http://www-4.ibm.com/software/data/db2

[10] http://www-4.ibm.com/software/data/iminer/fortext/ibm_tse.html

[11] http://www.oracle.com/database/oracle8i/index.html

[12] http://www.oracle.com/intermedia

[13] http://www.softwareag.com/adabas/product/strategy.htm

[14] http://www.softwareag.com/adabas/add_on/text_management.htm

frage primär als Faktenanfrage, aber daneben interessieren ihn auch Texte. Die Anfrage zielt auf den Markt für Maschinen in Kolumbien und die Anteile deutscher Unternehmen. Die Marktbetrachtung ist mit den in ELVIRA zur Verfügung stehenden statistischen Daten allerdings nur eingeschränkt möglich. Da keine Produktionsdaten für Kolumbien zur Verfügung stehen, beschränkt sich der Benutzer auf den Import und Export. Der Benutzer müsste folgende Fakten-Anfrage in ELVIRA stellen:

> *Export und Import für Maschinenbau insgesamt zwischen Deutschland und Kolumbien und für Kolumbien insgesamt (Summe Berichtsländer)*

Ein optimales System liefert eine gemischte Ergebnisliste, in der Zeitreihen und Texte enthalten sind. Um bei dieser Zeitreihen-Anfrage auch interessante Texte finden zu können, muss das System die Anfrage intern in eine Textanfrage transformieren. In diesem Fall sollte das System zu folgender Textanfrage gelangen, wie sie in etwa ein Experte formulieren würde:

> *Quelle: Textdokumente / Land: Kolumbien / Freitext: Maschinen* und (Import oder Export)*

Der Nomenklatureintrag *Maschinenbau insgesamt* ist als Term für die Textsuche ungeeignet. Mit dem Term *Maschinenbau* oder *Maschinen** dagegen werden Dokumente gefunden. Aufgrund der Struktur der Außenhandelsstatistik ist der Term *Summe Berichtsländer* in einer Textsuche nicht sinnvoll.

5.1.2 Heterogene Erschließung und Qualität

Auch homogene Objekte können zu Heterogenität führen, wenn sie z.B. unterschiedlich inhaltlich erschlossen werden oder unterschiedlicher Qualität sind. Selbst die Anwendung des gleichen Indexierungsverfahrens führt bei verschiedenen Dokument-Mengen zu heterogenen Termlisten. Unterschiedliche Verfahren wiederum extrahieren aus gleichen Dokumenten verschiedene Terme, was etwa Fusionsansätze ausnutzen (cf. Abschnitt 2.3.1.2). In jedem Fall repräsentieren unterschiedliche Term-Räume oder Repräsentationsmechanismen die Dokumente.

Ein weiterer Aspekt ist die Qualität, die sich auf die Dokumente oder auf die Inhaltserschließung beziehen kann. Häufig erscheinen Benutzern Dokumente aus bestimmten Quellen besonders glaubwürdig, während sie z.B. Dokumenten von unbekannten Informationsanbietern aus dem Internet zunächst wenig Vertrauen schenken. Die Mechanismen für das Vertrauensmanagement in Informationssystemen wie z.B. Autoritätsbeweis oder Zertifizierung untersucht Kuhlen 1999. Anbietern mit höherer Glaubwürdigkeit traut man in vielen Fällen auch eine bessere Inhaltserschließung zu, bzw. man weiß, dass die In-

haltsanalyse professionell betrieben wird. Inhaltliche Kompetenz führt besonders bei der manuellen Indexierung eher zu besseren Ergebnissen. Dagegen sind die Dokumente kleiner Informationsanbieter teilweise nur schlecht oder gar nicht erschlossen.

In vielen Situationen wird ein Benutzer jedoch auch ein Dokument mit niedriger Qualität akzeptieren, wenn er ansonsten keine relevanten Dokumente erhält. Das Schalenmodell (cf. Krause 1996a, 1998) ist ein dezentraler Ansatz zur Bearbeitung dieser Problematik aus Sicht eines Fachinformationszentrums. Zentrale Informationsstellen haben bisher häufig monolithische Dokumenten-Kollektionen mit zentralen Thesauri aufgebaut. Die Forderung nach hoher Konsistenz und Homogenität der Daten führte zum Ausschluss potenziell relevanter Dokumente, die hinter diesen formalen Anforderungen zurückbleiben.

Das Schalenmodell berücksichtigt in einem Datenpool verschiedene Schalen von Dokumenten, die mit verschiedener Qualität erschlossen sind. Von einem inhaltlich möglichst detailliert erschlossenen Kernbereich nimmt die Qualität der Daten und die Übereinstimmung mit den Vorschriften zur Inhaltserschließung in äußeren Schalen ab. Je nach Möglichkeit oder Wunsch siedeln sich kleinere, externe Anbieter auf verschiedenen Schalen an. Das Schalenmodell führt damit Vagheit ein und lockert die bisher exakt verwaltete Konsistenz. Es ermöglicht dadurch auch die Integration automatisch und intellektuell indexierter Dokumente in einer Kollektion. Wird im Schalenmodell z.B. aus Kostengründen der Kernbereich verkleinert, kann die zentrale Stelle eine neue Schale mit automatisch erschlossenen Dokumenten einführen, um den bisherigen Umfang zu sichern.

Die zentrale Einrichtung erhält die Rolle eines Moderators, der den Kernbereich betreut und die Schalen definiert. Der Benutzer hat die Möglichkeit, beim Retrieval gezielt den Kernbereich oder bestimmte Schalen anzusteuern, je nachdem welche Qualitätskriterien er für diese Anfrage anlegt.

Auch bei Wirtschaftsinformationen wie etwa im Rahmen von ELVIRA (cf. Abschnitt 2.2.3.2) führen unterschiedliche Qualität und Glaubwürdigkeit der Daten zu Problemen. Nach Aussagen der Verbände sind viele statistische Daten von zweifelhafter Qualität. Trotzdem geben die Mitarbeiter sie weiter, wenn keine anderen Daten vorhanden sind. Interpretiert man dies im Rahmen des Schalenmodells, so wählt der Informationsvermittler automatisch die adäquate Schale, in der sich Dokumente befinden, und weist den Benutzer auf ein eventuell niedrigeres Qualitätsniveau hin.

5.1.3 Multilingualität

Eine der auffälligsten Folgen der weltweiten Vernetzung ist die Möglichkeit des Zugriffs auf Informationen in unterschiedlichen Sprachen. Die Menge der zur Verfügung stehenden Dokumente steigt so schnell an, dass eine intellektuelle Übersetzung aller Texte nicht in Frage kommt. Trotzdem wollen Benutzer auf den Nachweis relevanter Literatur in verschiedenen Sprachen nicht verzichten. Daher werden Informationssuchende zunehmend die Ergebnisdokumente nicht auf die Sprache der Anfrage einschränken. Selbst mittlere bis geringe passive Sprachkenntnisse reichen oft aus, um ein Dokument zu verstehen oder zumindest vorläufig über die Relevanz zu entscheiden und es eventuell übersetzen zu lassen. Dagegen sind nicht immer die entsprechenden Kenntnisse zur aktiven Produktion sprachlicher Äußerungen vorhanden, die für das Formulieren von Anfragen erforderlich sind. Gerade dann ist es sinnvoll, cross-linguales Retrieval einzusetzen, dem Benutzer also Dokumente in Sprachen vorzulegen, die nicht der Anfragesprache entsprechen.

Verschiedene Ansätze ermöglichen cross-linguales Retrieval. Häufig wird die Anfrage übersetzt, aber auch die Übersetzung der Dokumente ist denkbar. Daneben gibt es Verfahren, die ohne eine explizite Übersetzung direkt auf Kookkurrenzen aufsetzen. Dafür müssen die Dokumente teilweise übersetzt vorliegen und so ein Doppelkorpus bilden (cf. Abschnitt 5.3.2.3). Einen Überblick über multi- und cross-linguales Retrieval bieten Hull/Oard 1997, Oard 1997 und Braschler et al. 1999.

5.1.4 Lösungsansatz: Behandlung von Heterogenität durch Transformationen

Heterogenität führt meist zu heterogenen Beschreibungssprachen oder Indexierungsschemata für die Beschreibung der Retrieval-Objekte. Information-Retrieval-Systeme berechnen die Ähnlichkeit zwischen Anfrage und Dokument, um dem Benutzer die passendsten Dokumente für seine Anfrage zu liefern. Werden nun Anfrage und Dokument unterschiedlich repräsentiert, können die meisten Systeme keinen Vergleich durchführen. Das Ziel der Heterogenitätsbehandlung besteht in einer Vereinheitlichung der heterogenen Repräsentationen und damit gewissermaßen in einem Übersetzungsprozess zwischen verschiedenen Beschreibungssprachen. Wie bei natürlichen Sprachen beinhaltet dieser Prozess syntaktische und semantische Probleme. Bei den meisten Indexierungsschemata können die syntaktischen Probleme vernachlässigt werden.

Diese Arbeit beschäftigt sich überwiegend mit den semantischen Problemen. Die Beschreibungssprache besteht im Information Retrieval meist aus einer

Liste von Termen und Gewichten. Das Vektorraum-Modell interpretiert die Terme als Achsen eines Raumes. Demnach spannen die Terme einen Merkmalsraum auf, in dem die Dokumente platziert sind, wobei die Achsen oder Terme prinzipiell gleichberechtigt sind. Jedes weitere Indexierungsverfahren führt zu einem weiteren Merkmalsraum. Die Anfrage ist bei heutigen Systemen von den Benutzern frei formuliert, ohne dass die Anfrageterme auf eine bestimmte erlaubte Menge beschränkt sind. Aber auch restringierte Zugänge kommen in der Praxis vor. Erlaubt eine Benutzungsoberfläche nur die Auswahl von Termen aus einem Thesaurus oder einer Klassifikation, dann entsteht die Anfrage in diesem Term-Raum mit weniger Dimensionen.

Der Vergleich einer Anfrage mit den Dokumenten erfolgt bei den meisten Systemen auf einer einheitlichen Basis und damit innerhalb einer Repräsentationsform. Ein Retrieval-System muss dazu die heterogen repräsentierten Objekte in ein Repräsentationsschema bringen.

Einen Lösungsansatz bieten Transformationen, die den Umstieg von einer Beschreibungssprache in eine andere und damit eine Abbildung von einem Term-Raum in einen anderen leisten. Sie erstellen ausgehend von der Repräsentation eines Objekts in einem Raum seine Repräsentation in einem anderen. Kuhlen etwa spricht von einer „postkoordinierenden Ordnung durch transformierende Anpassung“ (Kuhlen 1999:138). Präkoordination wäre in diesem Fall eine Standardisierung der Inhaltserschließung, wie sie in der dokumentarischen Tradition oft angestrebt wird, die aber in weltweiten Datennetzen nicht durchsetzbar ist.

Grundsätzlich kann die Anfrage in alle Beschreibungssprachen transformiert werden, dann müssen die Ähnlichkeitswerte aus verschiedenen Term-Räumen verglichen werden. Sinnvoller ist es daher meist, alle Retrieval-Objekte in einem Termraum zu repräsentieren.

Folgende Verfahren für Transformationen sind bekannt:

- Exakte Verfahren
- Vage Verfahren
  - Statistische Verfahren
    - Berechnung von Assoziationen
    - Assoziationen auf der Basis von LSI-Repräsentationen
  - Neuronale Netze
    - Hopfield- und Spreading-Activation-Netzwerke
    - Transformations-Netzwerk

Die statistischen Verfahren sind verbreitet und werden häufig eingesetzt. Daneben erscheint besonders das Transformations-Netzwerk als erfolgversprechend. Es besteht aus einem Backpropagation-Netzwerk, das mächtiger ist als neuronale Netze ohne versteckte Schichten (cf. Abschnitt 3.5.4.1). Zusätzlich wird in den Abschnitten 5.3.5 und 6.4.4 das für Heterogenitätsbehandlung adaptierte COSIMIR-Modell vorgestellt.

Eine Unterscheidung der Verfahren für Transformationen ergibt sich auch aus der Art des benutzen Wissens:

- Exakte Verfahren nutzen bekanntes und explizit vorhandenes Wissen, das von Menschen formuliert und gepflegt wird (Terminologielisten, Regelwerke, Expertensysteme).
- Dagegen nutzen vage Verfahren Wissen, das in der benutzten Form nicht der Mensch formuliert, sondern das z.B. aus statistischen Zusammenhängen abgeleitet wird. Bei den meisten Verfahren gibt der Mensch Wissen in Form von Beispielen für die gewünschte Abbildung vor. Das System selbst implementiert daraus die Abbildungsfunktion.

5.2 Exakte Verfahren für Transformationen

Exakte oder deduktive Verfahren leiten die Repräsentation eines Objekts in dem Ziel-Merkmals-Raum nachvollziehbar her und setzen dafür auf bekanntes und formalisiertes Wissen ein.

5.2.1 Thesauri und Konkordanzen

Thesauri sind systematische Sammlungen von Begriffen zu einem Fachgebiet, die eine konsistente Benutzung des Vokabulars zum Ziel haben. Sie dienen als Hilfsmittel in allen Phasen des Information Retrieval, was auch die entsprechende DIN-Norm festhält:

> „Ein Thesaurus im Bereich Information und Dokumentation ist eine geordnete Zusammenstellung von Begriffen und ihren (vorwiegend natürlichsprachlichen) Bezeichnungen, die in einem Dokumentationsgebiet zum Indexieren, Speichern und Wiederauffinden dient.“ (Burkart 1997:160)

Ein verbreitetes Verfahren, Beziehungen zwischen heterogenen Repräsentationsschemata herzustellen, ist die intellektuelle Analyse und Zuordnung. Dies kann ein übergreifender Thesaurus leisten, der alle in den Nomenklaturen,

Terminologien oder Wortlisten vorkommenden Begriffe umfasst und sie den entsprechenden Synonymen oder Ober- und Unterbegriffen zuordnet.

Ein derartiger Thesaurus stellt eine für die Anwendung gültige Einteilung der Welt dar. Häufig gelingt insbesondere bei heterogenen Datenbeständen keine Einigung auf einen solchen Thesaurus. Da alle bestehenden Systeme jeweils einen Standpunkt darstellen, der aus seiner Perspektive gerechtfertigt ist und der bei manchen Benutzungssituationen überlegen sein kann, ist dieser Ansatz allein nicht sinnvoll. Außerdem fügt man damit in einem ohnehin heterogenen Umfeld den bestehenden Systematiken eine weitere hinzu.

Anstatt eine neue Systematik einzuführen, können auch Verbindungen zwischen bereits bestehenden analysiert und in einer Konkordanz festgehalten werden. Eine Konkordanz ermöglicht den exakten Überstieg von den Begriffen in einem Thesaurus in die entsprechenden Begriffe der anderen. Der intellektuelle Aufwand ist vergleichbar mit der Erstellung eines Thesaurus. Allerdings treten in der Praxis häufig Beziehungen auf, die nicht eindeutig sind.

Nikolai et al. 1998 und Kramer et al. 1997 schlagen Thesaurusverbünde vor, bei denen semantische Beziehungen zwischen verschiedenen Thesauri intellektuell erschlossen werden. Dabei sind die gleichen Typen von Beziehungen erlaubt wie innerhalb eines Thesaurus also v.a. Synonyme, Ober- und Unterbegriffe. Die einzelnen Thesauri bleiben im Verbund erhalten und ebenso eventuelle Widersprüche zwischen ihnen.

Ein Beispiel für ein durch soziale Interaktion zu schaffendes System ist der Ansatz von Sigel 1998. Darin werden Beziehungen zwischen verschiedenen Produktnomenklaturen zunächst statistisch oder intellektuell ermittelt. Im konkreten Anwendungsfall wurden das Yahoo-Branchenbuch, die Kategorien von „Wer liefert was?“ und die Gelben Seiten Deutschlands und Italiens in Beziehung gesetzt. An diesem Ausgangspunkt setzt das System bizzyB an, das Interaktion mit dem vorliegenden Wissen erlaubt, um so das statistische Wissen nach und nach um menschliches Wissen zu erweitern (cf. Sigel 1998). Dabei ist allerdings unklar, wie die parallele Arbeit von vielen Beteiligten sich auf die Qualität auswirkt.

5.2.2 Regelsysteme

Für die Transformationen in heterogenen Information-Retrieval-Systemen sind grundsätzlich auch Regelsysteme geeignet, die im Wesentlichen aus Wenn-Dann-Regeln bestehen. Wird die Regelbasis komplexer und umfangreicher, muss evtl. ein Expertensystem die Regeln verwalten und ihre Beziehungen kontrollieren.

5.2.3 Nachteile exakter Verfahren

Exakte Verfahren sorgen zwar für nachvollziehbare Transformationen, sie weisen jedoch auch Nachteile auf. Das schwerwiegendste Argument liegt in dem hohen intellektuellen Aufwand, der für große Datenmengen, wie sie in realen Anwendungen vorkommen, kaum geleistet werden kann. Weiterhin bilden Thesauri mit ihrer Kunstsprache immer nur einen Teil der Welt ab und umfassen nicht alle Begriffe. Ihre Erstellung und Pflege ist aufwendig, insbesondere in einem dynamischen und sich ändernden Gegenstandsbereich, der ständige Aktualisierung erfordert.

Auch Konkordanzen bereiten Probleme. Häufig sind die Beziehungen zwischen Begriffen nicht eindeutig und kontextabhängig, so dass die Sichtweise des Bearbeiters ausschlaggebend ist. Gerade bei unterschiedlichen hierarchischen Klassifikationen ist eine Zuordnung verschiedener Gruppen schwierig. Eine Konkordanz zwischen der von einem Volltext-Indexierungssystem erstellten Wortliste und einer Klassifikation kann intellektuell ohnehin kaum erstellt werden.

Transformationen müssen also auch mit wenig und unsicherem Wissen robust ablaufen.

5.3 Vage Verfahren für Transformationen

Vage Transformationen ergänzen exakte Verfahren wie Thesaurus-Erweiterungen und Konkordanzen und führen von einem oder mehreren Begriffen aus einem Begriffs-Schemata zu einem gewichteten Vektor von Begriffen in einem anderen Begriffs-Schemata.

Text-Retrieval setzt vermehrt vage Verfahren ein. So haben z.B. statistische Verfahren das traditionelle Boolesche Modell ersetzt und bilden heute die Basis zahlreicher kommerzieller Systeme. Auch neuronale Netze wurden in den letzten Jahren in Information-Retrieval-Systemen erfolgreich eingesetzt, wie der Überblick in Kapitel 4 zeigt.

Wie im Text-Retrieval läuft auch die Entwicklung der Heterogenitätsbehandlung hin zu Verfahren, die verstärkt Vagheit zulassen. Die folgenden Abschnitte zeichnen diese Entwicklung von erprobten statistischen Verfahren hin zu noch weitgehend experimentellen Verfahren auf der Basis neuronaler Netze. Bei diesen Verfahren ist kein intellektueller Aufwand für die Erstellung der Transformations-Funktion erforderlich. Statt dessen ist ein sogenanntes Doppelkorpus mit Trainings- oder Lerndokumenten erforderlich, die nach

beiden zu verbindenden Schemata erfasst wurden. Anhand dieser Datengrundlage erlernt oder modelliert das Verfahren die entsprechende Funktion.

Den größten Erfolg versprechen Expertenurteile über Zusammengehörigkeit von Dokumenten. Daneben kommen Benutzerurteile über Relevanz-Feedback in Frage. Eine geringere Sicherheit als Relevanz-Feedback-Urteile sind in einer Sitzung gemeinsam oder direkt hintereinander abgefragte Dokumente. Sowohl Experten- als auch Benutzerurteile haben den großen Nachteil, dass sie relativ langsam entstehen. Es kostet viel Zeit und Ressourcen, eine ausreichende Menge zu sammeln. Die neuen Technologien wie das Internet bieten auch in diesem Zusammenhang interessante Möglichkeiten. So berichten z.B. Mattox et al. 1999 von einer Anwendung, in der eine Firma die Interessenprofile von Mitarbeitern durch die Aufzeichnung der Zugriffe auf Internet-Seiten ermittelt. Eine Vorstudie ergab, dass die Verweildauer auf einer Seite mit dem Interesse und der Kompetenz zu den darin besprochenen Themen korreliert.

Im Anwendungsfall ELVIRA (cf. Abschnitt 2.2.3.2 und 5.1.1) stehen Doppelkorpora nicht zur Verfügung und wurden testweise auf heuristische Weise gewonnen (cf. Mandl 1999). Das Verfahren nutzt die doppelte Erschließung der BfAI-Texte aus, die intellektuell indexiert sind und zusätzlich automatisch erschlossen wurden. Im Volltext erfolgt eine Suche nach den Termen der Zeitreihenbeschreibung, so dass Volltext und Warenverzeichnis parallelisiert werden. Diese Daten besitzen dann nicht die Qualität eines Doppelkorpus, sie sollten aber je nach Anwendungsfall in Betracht gezogen werden. Die Implementierung dieser Transformation greift auf kommerziell verfügbare Software zurück. Damit wird eine kostspielige Eigenentwicklung vermieden.

Die Transformations-Ansätze zur Behandlung von Heterogenität eignen sich neben der Transformation von Anfragen für das Retrieval auch als Vorschlagmodus für die intellektuelle Inhaltserschließung. Dabei wird entweder eine Anfrage transformiert und der Benutzer erhält Vorschläge für andere Anfrage-Terme (cf. Schatz et al. 1996) oder ein Indexierer erhält ausgehend vom Volltext eines Dokuments einen Vorschlag für Indexterme aus einem kontrollierten Vokabular (cf. Chung et al. 1998).

5.3.1 Statistische Verfahren

Statistische Verfahren auf Basis des Vektorraum-Modells werden am häufigsten für vage Transformationen eingesetzt. Grundlage sind Assoziationswerte zwischen den Termen, die sich aus der Analyse von Kookkurrenzen ergeben. Die Gewichtungsformeln für Assoziationen ähneln den Ansätzen zur Gewichtung von Termen in Dokumenten (vgl. Abschnitt 2.1). Eine Assoziationsmatrix fasst das gemeinsame Auftreten von Termen in Dokumenten

zusammen. Auch IR Systeme für homogene Daten greifen auf solche Werte zurück, um Anfragen um zusätzliche Terme zu erweitern (Query-Expansion) oder um Cluster von Dokumenten zu bestimmen. Die Beziehungen zwischen den Termen lassen auch auf Beziehungen zwischen den Dokumenten schließen. Ein Überblick über Assoziationsmaße im Information Retrieval bietet Ferber 1997.

Ein typisches Beispiel präsentieren Chen/Martinez et al. 1998 mit der Analyse einer Untermenge der Literaturdatenbank INSPEC mit 400.000 Dokumenten. Der aus den Kookkurrenzen automatisch generierte Thesaurus wurde mit dem intellektuell erstellten Thesaurus der INSPEC Datenbank verglichen. In dem Experiment beurteilten Testpersonen wie gut die Thesauri bei Eingabe eines Terms damit in Beziehung stehende Terme findet. Dabei stellten Chen/Martinez et al. 1998 fest, dass der automatisch generierte Thesaurus beim Retrieval von Termen einen besseren Recall erreicht als der intellektuell erstellte Thesaurus, während die Precision auf gleichem Niveau liegt.

Der nächste Abschnitt zeigt, dass ähnliche Verfahren auch in anderen Bereichen und insbesondere in Datenbanken eine wichtige Rolle spielen. Die darauf folgenden Abschnitte stellen verschiedene Ansätze und Systeme für statistische Transformationen im Information Retrieval vor.

5.3.2 Assoziationen

Heterogenität bei Datenbanken entsteht z.B. durch die virtuelle oder reale Zusammenführung mehrerer Datenbanken. Dies führt häufig zu erheblichen Problemen. Scheuermann et al. 1998 behandeln die Unsicherheit globaler Objekt-Identität, mit der sich verteilte Daten eindeutig einem Objekt zuordnen lassen. In ihrem Ansatz sucht ein kombiniertes System mit Kohonen-SOM und einem Backpropagation-Netzwerk nach ähnlichen Objekten. Sciore et al. 1994 integrieren heterogene Datenbank-Schemata, wobei sie sich besonders mit verschiedenen Größenangaben für numerische Daten beschäftigen. Dieses Problem von formal aber nicht semantisch vergleichbaren Spalten lösen sie durch die Einführung semantischer Werte für einzelne Größen.

Auch Assoziationsmaße spielen in Datenbanken eine große Rolle, so etwa beim Data Mining oder bei vagen funktionalen Abhängigkeiten. In beiden Fällen wird ein Maß für eine Beziehung zwischen Objekten berechnet. Diese Objekte sind meist Fakten aus relationalen Datenbanken. Eine einheitliche Sichtweise auf fuzzy funktionale Abhängigkeiten und Assoziationsregeln entwickeln Delgado et al. 1999.

Fayyad 1997 definiert Data Mining als einen Schritt im iterativen und interaktiven Prozess des Knowledge Discovery in Databases. In den damit

verbundenen einzelnen Schritten „data warehousing, target data selection, cleaning, preprocessing, transformation and reduction, data mining, model selection (or combination), evaluation and interpretation, and finally consolidation and use of the extracted knowledge“ (Fayyad 1997:6) spielt Unsicherheit eine große Rolle. Das Bilden eines Modells aus sehr vielen Daten ist ein sehr vager Prozess.

Data Mining hat im Wesentlichen das Ziel, versteckte Zusammenhänge in Datenbanken zu finden. Dabei reicht die Abfrage exakter Fakten wie bei einer normalen SQL-Abfrage nicht aus. Vielmehr geht es um das Auffinden unsicheren Wissens, das so nicht explizit in der Datenbank abgespeichert wurde, sondern aus einer Gesamtsicht auf zahlreiche Tupel und Spalten entsteht. Damit kann Data Mining auch als Aggregation oder Reduktion gespeicherter Daten auf einige Regeln betrachtet werden. Eine wichtige Anwendung ist Marketing, wo z.B. das Kaufverhalten von Kunden analysiert wird, um einen höheren Erfolg von gezielten Werbemaßnahmen zu erreichen. Dem gemeinsamen Vorkommen von Termen entspricht im Marketing der Kauf von mehreren Produkten von einem Kunden. Einen Überblick über Data Mining bietet Nakhaeizadeh 1998. Wichtig ist in diesem Zusammenhang das Konzept des Data Warehouse, einer Plattform für eine Organisation, in der alle Daten zusammenfließen. Das Data Warehouse geht über die Leistungen einer Datenbank hinaus. Es integriert heterogene Daten aus allen Bereichen der Organisation, ist themenoriertiert und zeichnet alle zeitlichen Veränderungen in einer Datenbasis auf. Das Data Warehouse bildet die Grundlage für Data Mining und integriert Auswertungsverfahren in seiner Architektur (cf. z.B. Anahory/Murray 1997).

Wichtige Kennzahlen bei der Generierung einer Assoziations-Regel sind die Maße Support und Confidence. Support berechnet sich aus der Anzahl von Objekten, die eine Assoziations-Regel zwischen zwei Mengen von Attributen stützen, die also gleiche Werte für diese Attribute besitzen. Confidence beschreibt das Verhältnis von Support zur Anzahl der Objekte, bei denen die erste Menge von Attributen diese Werte annimmt. Confidence beschreibt also die Anzahl der übereinstimmenden Objekte zur Grundmenge an Objekten mit dieser Eigenschaft. Beide Maße sollten bestimmte Minima erreichen, bevor eine Regel akzeptiert wird (cf. z.B. Amir et al. 1997:333f.).

Da Data Mining oder Knowledge Discovery besonders bei großen Datenmengen interessant ist, spielt Effizienz bei der Implementierung eine wichtige Rolle. Amir et al. 1997 realisieren einen effizienten Algorithmus, der auch dynamisch auf Änderungen in der Datenbank reagiert.

Fuzzy funktionale Abhängigkeiten (Fuzzy functional dependencies, FFD, cf. Delgado et al. 1999) sind eine Verallgemeinerung von funktionalen Abhän-

gigkeiten aus dem relationalen Datenbankmodell. Eine funktionale Abhängigkeit besteht dann, wenn der Wert eines Attributs die Werte von anderen Attributen funktional bestimmt. Darauf baut formal das Konzept des Schlüssels in relationalen Datenbanken auf. Schlüssel sind Attribute, die einen Datensatz eindeutig identifizieren und von denen alle anderen Attribute funktional abhängig sind (cf. Heuer/Saake 1997:188ff.).

Fuzzy funktionale Abhängigkeiten dagegen erweitern dieses Konzept auf Vagheit und Unsicherheit, wie sie in realen Daten sehr häufig auftritt. Die wichtigsten Anwendungsgebiete sind vage Daten und vage Beziehungen. Bei vagen Daten dient das Konzept der fuzzy funktionalen Abhängigkeiten der Vermeidung von Redundanz und damit der leichteren Pflege und der effizienten Speichernutzung. Die Analyse von vagen Beziehungen entspricht der Suche nach Assoziationsregeln im Data Mining. Werden hinreichend gute Regeln gefunden, können diese auch für die Reduktion der Daten dienen. Dazu wird eine Spalte, die sich aus einer anderen herleiten lässt, gelöscht.

Auch in den Sozialwissenschaften spielen statistische Transformationen zwischen den Eigenschaften von Objekten eine Rolle. Unter dem Begriff Datenfusion wird dort die Übertragung von Datensätzen aus Umfragen auf andere Umfragen behandelt (cf. Gabler 1997). Dabei sind die Objekte oft nicht identisch wie im Bereich Data Mining oder Information Retrieval, da die Anonymisierung in den sozialwissenschaftlichen Umfragen sehr wichtig ist. Für solche Fälle tritt die Datenfusion als vages Verfahren an die Stelle der exakten Methoden. Die Identität wird von einer weitgehenden Übereinstimmung der Attributen, die in beiden Umfragen gemessen wurden, abgelöst. Die für einzelne Umfragen spezifischen Attribute werden dann für die weitgehend übereinstimmenden Objekte von einer Umfrage auf die andere übertragen. Die Vagheit ist dabei natürlich besonders hoch. Wie Gabler 1997 anhand von Beispielen und einer theoretischen Analyse zeigt, können durch diese Technik viele der für Sozialwissenschaftler interessanten Zusammenhänge verlorengehen. Insgesamt schätzt er die Rolle der Datenfusion für diesen Bereich eher pessimistisch ein (Gabler 1997:89).

5.3.2.1 Text Categorization

Der Begriff Text Categorization steht meist Ansätze, die zwischen dem Vokabular aus einer automatischen Indexierung und einem kontrollierten Thesaurus abbilden. Damit bildet Text Categorization eine Mischform aus automatischer und intellektueller Indexierung. Der Text wird zwar automatisch analysiert, den Texten werden aber Terme aus einer kontrollierten Liste

zugewiesen, wie es sonst bei intellektueller Verschlagwortung üblich ist. Text Categorization kann folgende Aufgaben erfüllen:

- Vorschlagmodus
  Die Indexierer erhalten einen automatisch erstellten Vorschlag, den sie bei ihrer Arbeit berücksichtigen können.
- Information Filtering
  In diesem Anwendungsfall kehrt sich die Rolle von Dokumenten und Anfragen um. Die Anfragen sind stabile, lange gültige Profile, die ein Benutzer festlegt. Die Dokumente sind flüchtige Elemente eines Stroms, aus dem nur die zu dem Profil passenden ausgefiltert werden. Solche Anwendungen sind etwa bei Presse-Agenturen, Nachrichtentickern oder anderen Arten von flüchtigen Dokumenten sinnvoll. Eine Anwendung für die Ausschreibung von Forschungsprojekten bietet Yasdi 1999. Filtering erfreut sich auch im Internet immer größerer Beliebtheit (cf. z.B. Schirmer/Müller 1999) und entspricht den Routing-Aufgaben im TREC-Kontext (cf. Abschnitt 2.1.4.2).
- Behandlung von Heterogenität
  Aus Kostengründen wird oft selbst bei Instituionen, die intellektuell indexieren, diese Art der Inhaltserschließung nicht auf alle oder nicht auf neue Datenbestände angewandt. In solchen Fällen weist ein Text-Categorization-System die Thesaurus-Terme automatisch zu.

In den Bereich Vorschlagmodus fällt das System AIR/PHYS, das im Rahmen des Darmstädter Indexierungs-Ansatzes entstand (cf. Biebricher et al. 1988). Ziel war es, die Indexierungsqualität der intellektuellen Indexierung durch einen automatisch erstellten Indexierungsvorschlag zu erhöhen. Während die automatische Indexierung fast alle im Text vorkommenden Begriffe benutzt, ist die intellektuelle Indexierung auf einen Thesaurus beschränkt. Dessen Begriffe tauchen häufig in dieser Form nicht in den Texten auf. AIR/PHYS versucht daher Beziehungen zwischen dem automatischen Indexierungsergebnis und der intellektuellen Indexierung herzustellen. Dazu werden bereits intellektuell verschlagwortete Dokumente zusätzlich automatisch indexiert. Zwischen den beiden Repräsentationen ergeben sich Assoziationsfaktoren:

$$z\,(t,s) = \frac{h(t,s)}{f(t)}$$ *(Biebricher et al. 1988:334)*

t	*Freitext-Term*
s	*Thesaurus-Term*
f(t)	*Zahl der Dokumente, die t enthalten*
h (t,s)	*Zahl der Dokumente aus f(t), denen s zugeordnet wurde*

Der Thesaurus umfasst ca. 22.000 Begriffe. AIR/PHYS berechnet dafür 800.000 Assoziationsfaktoren, von denen es die 350.000 wichtigsten ins System übernahm. Diese Faktoren beschreiben quasi die Eigenschaften von Dokumenten, denen bestimmte Thesaurus-Begriffe zugeordnet sind. Die Übergangsfunktion enthält weitere Thesaurus-Beziehungen, wie z.B. um 50.000 USE-Relationen und 170.000 BROADER TERM-Relationen. In vielen Fällen wurden so über die automatische Indexierung zufriedenstellende Deskriptoren aus dem Thesaurus zugewiesen.

AIR/PHYS ist ein gutes Beispiel für statistische Abbildungen zwischen heterogenen Repräsentationen. Es treten alle typischen Probleme auf, wie die Erstellung von doppelten Repräsentationen für viele Dokumente, als Basis für statistische Verfahren. Pro Deskriptor wurden nur durchschnittlich siebzehn Assoziationsfaktoren gespeichert, die auch nur aus dem Abstract gewonnen wurden, was ein zu geringer Ausschnitt aus den relevanten Eigenschaften sein kann.

Einen analogen Anwendungsfall stellt Ferber 1997 vor. Die dort geschilderten Experimente unterstreichen die Wichtigkeit der empirischen Überprüfung der angewandten Formel. Die verwendete Formel enthielt zwei Parameter, die den Einfluss der Häufigkeit von Thesaurus-Termen und Freitext-Begriffen im Korpus regelten. Die Veränderung dieser Parameter wirkte sich sehr stark auf die Qualität der Abbildung aus (Ferber 1997:244ff.).

Ein weiteres System für die Generierung von Indexierungsvorschlägen aus einem kontrollierten Vokabular stellen Plaunt/Norgard 1998 vor. Eine Menge von 4626 Abstracts aus der INSPEC Datenbank bildet die Grundlage für das Experiment. Nach dem Training konnte in einer 460 Abstracts großen Testmenge die Übereinstimmung mit menschlichen Indexierern gemessen werden. Zwar liegen die Werte nicht sehr hoch, jedoch halten Plaunt/Norgard 1998 die Ergebnisse für sehr ermutigend, da z.B. die Indexierungsvorschriften oft zu einem spezifischeren Term führen als das automatische Verfahren.

Apté et al. 1994 stellen ein induktives Verfahren vor, das statistische Assoziationsregeln erstellt. Aufbauend auf Entscheidungsbäumen und anderen induktiven Methoden aus dem Bereich Künstliche Intelligenz entwickeln

Apté et al. 1994 einen eigenen Algorithmus Swap-1, der die gefundenen Regeln oder Assoziationen nachträglich optimiert. Nachträglich wird eine gefundene Menge von Regeln statistisch und durch Pruning-Verfahren weiter verbessert. Diese Text-Categorization klassifiziert deutsche und englische Nachrichtenagenturtexte.

Yang 1995 erhöht die Effizienz der Assoziationsberechnungen durch Dimensionsreduktion auf mehreren Ebenen. Er führt zunächst Wortstärke als Maß dafür ein, wie gut ein Term zwei in Beziehung stehende Dokumente identifiziert:

$$Wortstärke_t = \frac{Anzahl\ der\ Paare\ bei\ denen\ t\ in\ beiden\ Dokumenten\ vorkommt}{Anzahl\ der\ Paare\ bei\ denen\ t\ im\ ersten\ Dokument\ vorkommt}$$

Yang 1995:258

Die Wortstärke eliminiert die schwächeren Terme. Während die durchschnittliche Term-Precision nahezu unverändert bleibt, spart das Verfahren ganz erheblich Rechenzeit. Je nach Kollektion eliminiert Yang 1995 bis zu 77% aller Terme bei gleichbleibender Precision. Zusätzlich führt Yang 1995 eine Singular Value Decomposition der verbleibenden Dokument-Term-Matrix durch wie sie Latent Semantic Indexing (LSI, cf. Abschnitt 2.1.2.4.3) einsetzt. Wieder entstand ein ähnlicher Effekt. Fast ohne Qualitätsverlust konnten 80% der Singular Values vernachlässigt werden. Im Vergleich zu dem Einsatz von LSI im normalen Information Retrieval ohne vorhergehende Eliminierung von wenig aussagekräftigen Termen ist der Anteil allerdings noch hoch. In der Regel verbleiben 100 bis 300 LSI-Dimensionen, so dass bei einem Ausgangsvokabular von über 30.000 eine Reduktion um 99% erfolgt.

Die Kombination der Methoden ergab in drei kleineren Test-Kollektionen jeweils eine geringfügige Verbesserung der durchschnittlichen Precision und eine Reduktion der Rechenzeit um zwischen 50% und 90%. Allerdings optimierte Yang 1995 den Anteil der reduzierten Terme oder Singular Values durch eine vollständige Analyse, was in einer realen Anwendung äußerst schwierig ist.

Lam/Ho 1998 präsentieren ein nicht lineares und lernendes Verfahren, das auf dem k-nearest-neighbor Algorithmus basiert. Eine Kategorisierung von Internet-Dokumenten auf der Basis von Kohonen Karten stellt Schatz 1998 vor (cf. Abschnitt 4.4).

5.3.2.2 Teilweise überlappende Fachgebiete

Einen etwas anders gelagerten Anwendungsfall stellen Chen/Martinez et al. 1996 vor. Im vorigen Abschnitt wurden Assoziationen zwischen verschiedenen Begriffsschemata errechnet, die alternative Zugriffsmöglichkeiten mit gleichem Stellenwert darstellen. Die Transformation von Chen/Martinez et al. 1996 schlägt eine Brücke zwischen verschiedenen Spezialgebieten mit teilweise überlappenden Fachbegriffen. Zwei unabhängig voneinander entwickelte Systeme für Biologen, die sich mit einer bestimmten Spezies Wurm bzw. Fliege beschäftigen besitzen ca. 30% gleiches Vokabular. Häufig ist interessant, wie Experten eines anderen Teilgebiets ein Thema behandeln. Eine Kookkurrenzanalyse der Texte ermöglicht die Transformationen.

Fachleute bewerteten das entstandene kombinierte System positiv (cf. Chen/Martinez et al. 1996:20f.). In einem Experiment gaben Fachleute ihre Assoziationen für einzelne Fragestellungen wieder und suchten diese dann im übergreifenden Thesaurus. Dabei fanden sich im automatisch erstellten Thesaurus nur ca. 8% der von den Experten geäußerten Assoziationen. Trotz dieses relativ niedrigen Wertes zeigen die Retrievalexperimente, dass der kombinierte Thesaurus den Recall erhöht, während die Precision konstant bleibt (cf. Chen/Martinez et al. 1996:19). Das Experiment zeigt, dass die reine Transformationsqualität nicht der Retrievalqualität entspricht. Bei Experimenten sollten Experten also in keinem Fall nur die Transformation bewerten, vielmehr muss die Transformationsfunktion im Kontext des Retrievals betrachtet werden.

Schatz et al. 1996 führen ein Beispiel für eine Anfrage an, die einen ähnlichen Transfer zwischen benachbarten Gebieten erfordert. Ein auf Brücken spezialisierter Bauingenieur will zum Einfluss von Wind auf lange Strukturen recherchieren. Da beim Verlegen von Unterseekabeln ähnliche Probleme durch die Strömung auftreten, soll seine Anfrage für diesen Bereich umgeformt werden. Schatz et al. 1996 unterstützen den Benutzer durch eine Kombination von Thesaurus und Kookkurrenzliste bei der Suche nach Termen. Der Thesaurus dient aber nur als Wortliste und nicht für die Indexierung. Nach der Suche nach Begriffen im relativ kleinen Thesaurus, der die Orientierung durch semantische Beziehungen erleichtert, erweitert der Benutzer diese Begriffe mit Hilfe der Kookkurrenzliste.

Allerdings ist fragwürdig, ob Benutzer bereit sind, vor dem ersten Suchschritt ein mehrstufiges Verfahren allein zur Identifikation von Termen durchzuführen. Die Terme sagen noch nichts über die Relevanz der damit verbundenen Dokumente aus. Zudem widerspricht dies dem primären Informationsbedürfnis, das auf Dokumente ausgerichtet ist und nicht auf Terme. Aus der IR-Forschung ist weiterhin bekannt, dass die Verbesserung der Anfrage durch

Relevanz-Feedback die besten Ergebnisse bringt. Deshalb ist es meist sinnvoller, eine Anfrage zu stellen und diese nach und nach anhand der Ergebnisdokumente zu verbessern. Daraus folgt auch, dass der Benutzer Term-Erweiterungen und die Transformation von Anfragen innerhalb heterogener Umgebungen nicht unbedingt vor der Anfrage sehen muss.

5.3.2.3 Multilinguales Retrieval

Sheridan/Ballerini 1996 setzen Assoziationsmaße für multilinguales Retrieval ein. Neben den lexikon-basierten Ansätzen, die z.B. die Anfrage maschinell übersetzen (cf. z.B. Hull/Grefenstette 1996), realisieren sie damit ein rein statistisches Verfahren. Auch dafür ist ein Doppelkorpus erforderlich, d.h. identische Dokumente liegen zumindest für eine Trainingsmenge in beiden Sprachen vor. Sheridan/Ballerini 1996 fassen in ihrem System die Dokumente als Eigenschaften der Terme auf und nicht umgekehrt wie sonst im Information Retrieval üblich. Die Bedeutung eines Terms besteht damit aus seiner Vorkommenshäufigkeit in den Dokumenten.

Dadurch bilden Sheridan/Ballerini 1996 zunächst einen monolingualen Ähnlichkeitsthesaurus, der auf Kookkurrenzen beruht. Eine Anfrage mit mehreren Suchtermen kann nun in ihre Eigenschaften, die Dokumente, überführt werden. Da die Dokumente in der anderen Sprache ebenfalls vorliegen, wird diese Repräsentation in die Suchterme der anderen Sprache transformiert. Der Ansatz nutzt die gesamte Dokument-Term-Matrix und somit jede Eigenschaft der Terme.

Sheridan/Ballerini 1996 haben mit diesem Verfahren u.a. einen Korpus von 93.000 Dokumenten bearbeitet, die sowohl in Italienisch als auch in Deutsch vorliegen. Das Ergebnis des multilingualen Retrieval ist zwar erwartungsgemäß schlechter als das des besten einsprachigen italienischen Vergleichsversuchs, aber die Qualität ist noch zufriedenstellend.

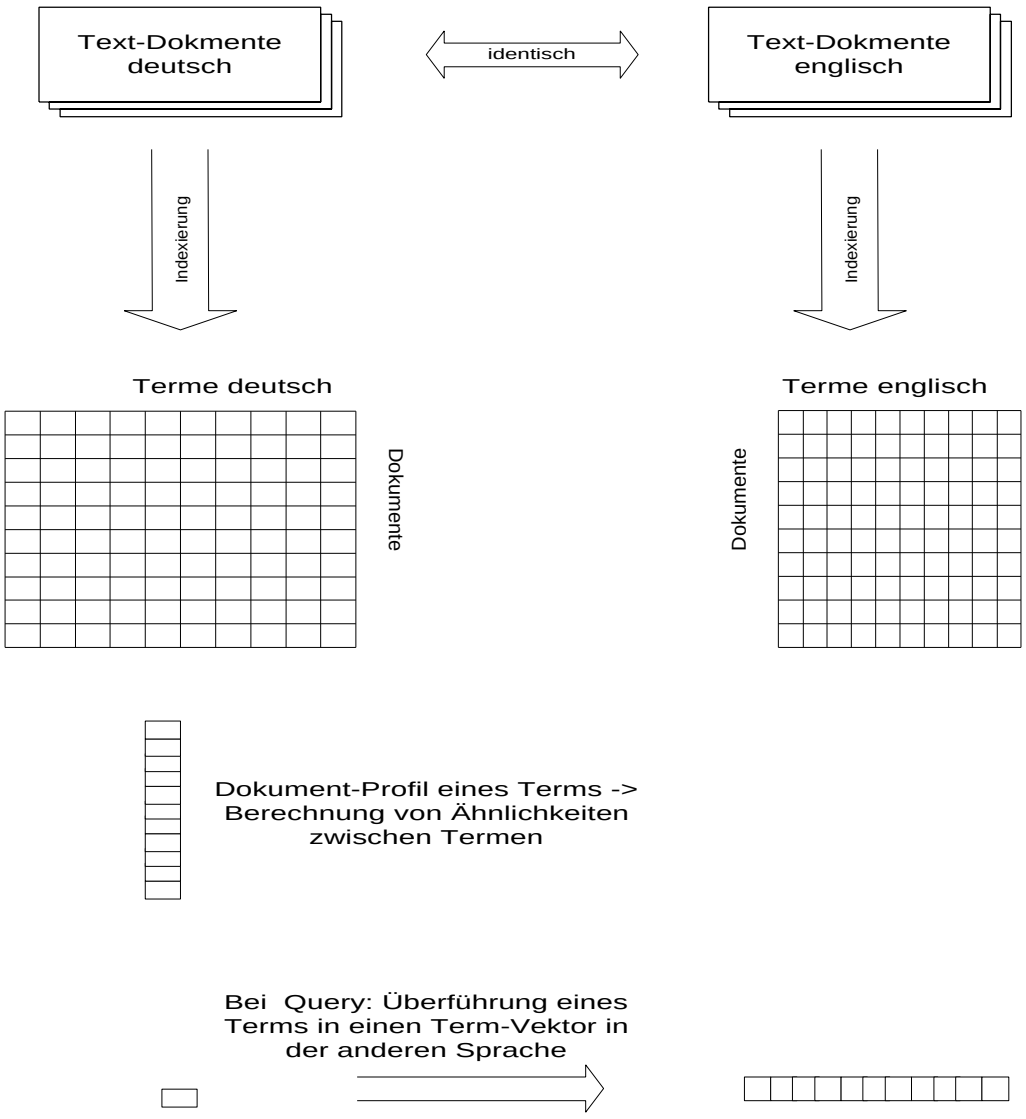


Abbildung 5-3: Schematische Darstellung des multilingualen Retrieval nach Sheridan/Ballerini 1996

Einen Überblick über neueste Evaluierungsergebnisse zum cross-lingualen Information Retrieval im Rahmen der TREC-Konferenz (cf. Abschnitt 2.1.4.2) bieten Braschler et al. 1999.

5.3.2.4 Latent Sematic Indexing für Transformationen

Latent Semantic Indexing (LSI) ist ein Verfahren zur Reduktion der Dimensionalität, das im IR mit Erfolg eingesetzt wird (cf. Abschnitt 2.1.2.4.3). Auch für Transformationen wird LSI benutzt, so z.B. für multilinguales Retrieval (cf. Young 1994). Dabei erfolgt die Transformation nach der Reduktion auf den Raum mit niedriger Dimensionalität. Wie bei LSI als IR-Verfahren be-

steht die Hoffnung, dass durch die Reduktion unwichtige Teile der ursprünglichen Matrix verloren gehen und nur die semantisch relevantesten Anteile übrig bleiben. Die Transformation soll dann auf Basis der wichtigen Strukturen besser gelernt werden.

Eine einfache Termerweiterung innerhalb des gleichen Vokabulars mit LSI testen Schütze/Pedersen 1997 und gelangen für die Tipster Kollektion (aufgegangen in TREC) zu besseren Ergebnissen als mit einem Vergleichsexperiment ohne Term-Erweiterung. Die Assoziationsmatrix zwischen den 450.000 Termen wurde schrittweise auf eine Matrix mit 450.000 mal 20 Elementen reduziert.

Dumais et al. 1997 beschreiben eine vergleichbare Anwendung für multilinguales Retrieval.

5.3.3 Hopfield- und Spreading-Activation-Netzwerke

Auch konnektionistische Modelle wurden bereits für Transformationsaufgaben getestet. Hopfield Netzwerke sind assoziative Speicher, die Retrieval aus unvollständigen Mustern erlauben (cf. Abschnitt 3.5.3). Als Information Retrieval System speichern Hopfield-Netze die Dokumente als Energie-Minima und die Anfrage wird als unvollständiges Muster eingegeben, von dem aus das Netz zum nächstliegenden Dokument konvergiert (cf. Abschnitt 4.2).

Chen 1995 speichert Muster aus verschiedenen Thesauri in einem Hopfield-Netzwerk. Benutzer definieren Cluster von zusammengehörenden Begriffen, die aus verschiedenen Thesauri stammen und definieren so Transformationen. Im Retrievalfall liefert das Netz nach Eingabe von Termen ähnliche Terme aus mehreren Thesauri. Ein weiteres Hopfield-Netzwerk nach diesem Prinzip stellen Lin/Chen 1994 im Bereich multilinguales Retrieval vor. Es leistet einen Umstieg zwischen chinesischen und englischen Termen.

Allerdings weicht diese Implementierung stark vom Konzept der Hopfield-Netzwerke ab. Anstelle der Hopfield-Lernregel, die garantiert, dass jedes Muster als Energie-Minimum abgespeichert wird, werden die Verbindungsstärken aus der Indexierung übernommen. Damit ähnelt das Netz eher einem Spreading-Activation-Netzwerk mit nur einer Schicht als einem Hopfield-Netzwerk.

Auch die in Abschnitt 4.3 beschriebenen Spreading-Activation-Netzwerke können für Transformationen adaptiert werden. Diese Systeme bestehen meist aus zwei Schichten, zwischen denen Aktivierung fließt. Eine davon repräsentiert die Anfrage-Terme und die zweite die Dokumente. Die Stärke der Verbindungen beschreibt die Wichtigkeit des Terms für das jeweilige Dokument. Beim Einsatz für Transformationen repräsentiert jede Schicht eine Begriffssystematik und die Verbindungsstärken geben die Gewichte der As-

soziationen zwischen den Termen in den unterschiedlichen Systematiken an. In einem solchen System verschwimmt die Grenze zwischen einem auto- und einem hetero-assoziativen Netzwerk. Da auch innerhalb der Term-Schicht eines Spreading-Activation-Netzwerks Verbindungen möglich sind, lässt sich das oben besprochene zweischichtige Netzwerk formal kaum davon abgrenzen. Vielmehr kann es als Sonderfall einer Term-Schicht mit Verbindungen innerhalb der Schicht betrachtet werden, bei dem nur zwischen bestimmten Gruppen Verbindungen zugelassen sind. Die Gruppen sind die unterschiedlichen Begriffs-Systematiken.

Die Abgrenzung der Begriffs-Systematiken kann durch zwei Verfahren verdeutlicht werden. Zum einen kann zwischen zwei Indexierungsvokabularen eine Schicht mit Objekten eingezogen werden, die von beiden Vokabularen beschrieben wird. Ein solches Netz, wie es Abbildung 5-4 skizziert, schlagen Lee/Dubin 1999 vor. In diesem Ansatz verbindet eine Transformation zweifach intellektuell indexierte Dokumente zur Astrophysik. Eine Liste von Termen stammt vom *Astrophysical Journal* und eine zweite von der NASA. Die erste Liste führte zu einer Schicht mit 4120 Neuronen und die zweite zu einer mit 2305 Neuronen. Dazwischen liegt eine Dokument-Schicht mit ca. 15.000 Neuronen. Die Autoren analysieren Kollektion und Ergebnisse in mehrerer Hinsicht. Allerdings berechnen sie nicht die Standard-Maße Term-Recall und Term-Precision und sind somit nicht mit anderen Experimenten vergleichbar.

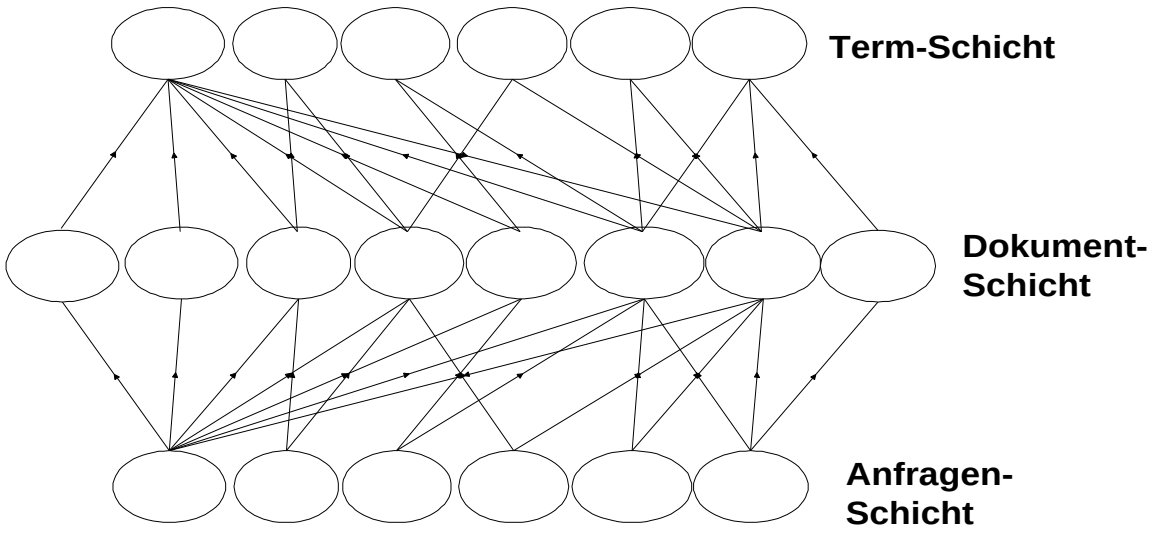


Abbildung 5-4: Mögliche Architektur eines Spreading-Activation-Netzwerks für Transformationen

Die Architektur zeichnet sich durch hohe Flexibilität aus, da sowohl Terme aus beiden Vokabularen als auch ein Objekt in der mittleren Schicht als Input dienen können. Ein Schritt in diese Richtung erfolgt bereits durch die dreischichtigen Spreading-Activation-Netzwerke, die zusätzlich zur Dokument- und Term-Schicht eine Autoren-Schicht besitzen. Formal bilden die Autoren auch Eigenschaften der Dokumente. Damit bilden Terme und Autoren die Begriffs-Systematiken, die mit den Dokumenten verbunden sind und die sich formal sogar teilweise überlappen könnten (vgl. Abschnitt 4.3.3).

5.3.4 Transformations-Netzwerk

Die Übertragung der Abbildung zwischen zwei Repräsentations-Verfahren auf ein Backpropagation-Netzwerk trennt die Objekte und ihre Eigenschaften deutlicher, was aber auch einen Verlust an Flexibilität bedeutet. Trotzdem ist der Einsatz des Backpropagation-Verfahrens aus verschiedenen Gründen sinnvoll. Die Spreading-Activation-Netzwerke sind den statistischen Ansätzen zum Information Retrieval sehr ähnlich und bilden keine konzeptuelle Neuerung (cf. Abschnitt 4.3.4). Backpropagation-Netzwerke hingegen enthalten versteckte Schichten, die Informationen auf sub-symbolischer Ebene repräsentieren. Sie bilden somit eine wesentliche Verbesserung. Das Transformations-Netzwerk kann als eine Erweiterung des oben skizzierten Spreading-Activation-Netzwerks mit zwei Schichten um eine oder mehrere versteckte Schichten betrachtet werden. Da das Backpropagation-Netzwerk prinzipiell mehr Klassen von Funktionen implementieren kann (cf. Abschnitt 3.5.4.1), erhöhen sich die Chancen auf eine erfolgreiche Transformation. Das Backpropagation-Netzwerk lässt sich so auf die Heterogenitätsproblematik anwenden. Als Trainingsdaten dienen Doppelkorpora, also Dokumente, die in beiden Repräsentationen vorliegen. Eine Repräsentation wird als Input angelegt und das Netz lernt, die andere Repräsentation zu bestimmen.

5.3.4.1 Das Transformations-Netzwerk von Crestani/van Rijsbergen

Crestani/van Rijsbergen 1997 und Crestani 1995 stellen ein Backpropagation-Netzwerk vor, das Abbildungen zwischen identischen Repräsentationen ausführt. Sie nutzen dieses bereits im state-of-the-art zu neuronalen Netzen im Information Retrieval ausführlich besprochene System (cf. Abschnitt 4.6.3), um eine initiale Benutzeranfrage in eine für das Informationsproblem optimierte Anfrage zu transformieren. Die optimale Anfrage ermitteln sie aus einer Modifikation der originalen Anfrage durch Relevanz-Feedback. Aufgrund des geringen Umfangs der Testdaten sind die Ergebnisse nicht sehr

aussagekräftig. Das Transformations-Netzwerk ist damit weniger erprobt und etabliert als die statistischen Verfahren. Weiterführende Experimente stellt Kapitel 7 vor.

Die Beschränkung von Crestani/van Rijsbergen 1997 auf zwei identische Term-Räume ist nicht notwendig. Das Transformations-Netzwerk ist flexibel und lässt sich für verschiedenste Anwendungsfälle adaptieren, sobald genügend Trainingsdaten in der Form eines Doppelkorpus vorliegen. Die Input- und Output-Schicht werden dann an die Term-Räume für die entsprechenden Repräsentations-Verfahren angepasst.

5.3.4.2 Transformation von Werkstoffen

Mandl 1994 und Ludwig/Mandl 1997 übertragen das Transformations-Netzwerk auf Faktendaten aus dem Bereich Werkstoffinformation (cf. Abschnitt 2.2.3.1). Dabei leistet das Netz eine Abbildung zwischen zwei Repräsentationen eines Werkstoffs. Das System wird im Rahmen der Evaluierung von COSIMIR in Abschnitt 7.4.1 aufgegriffen, da die verwendeten Daten in weitere Experimente eingingen.

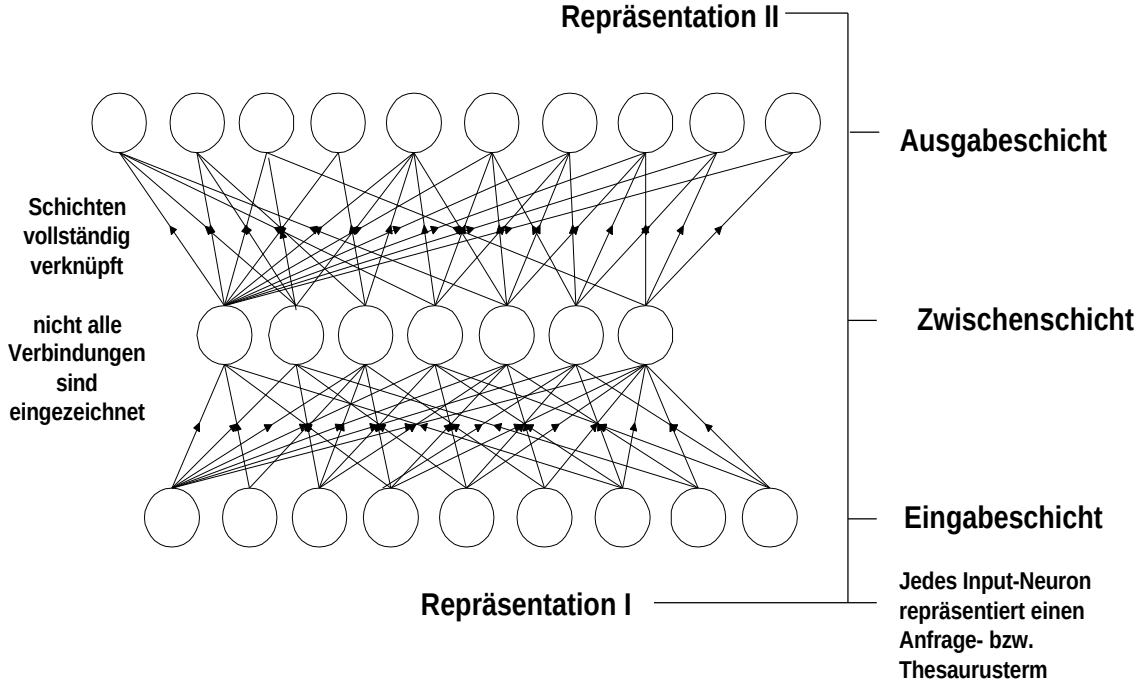


Abbildung 5-5: Das Transformations-Netzwerk

5.3.5 COSIMIR-Modell für heterogene Repräsentationen

Einen Sonderfall bildet das COSIMIR-Modell für Heterogenitätsbehandlung. Das COSIMIR-Modell, das im Folgenden Kapitel vorgestellt wird, ist ein Verfahren für Information Retrieval. Die Flexibilität des Modells lässt eine Adaption für heterogene Daten zu, in der ohne Transformations-Schritt das Retrieval direkt von heterogenen Repräsentationen ausgehend abläuft.

Das COSIMIR-Modell ist ein generelles Information Retrieval Modell, in dem ein neuronales Backpropagation-Netzwerk die Aufgabe der Ähnlichkeitsberechnung übernimmt. Die in Neuron ein, das die Relevanz repräsentiert und setzt es in. Als Input dienen Paare von Dokumenten und Anfrage. Trainingsdaten sind gewichtete Relevanzurteile zu diesen Paaren.

Das COSIMIR-Modell berechnet die Ähnlichkeit zwischen den zwei parallel anliegenden Input-Vektoren. Im Gegensatz zu mathematischen Ähnlichkeitsfunktionen (z.B. Kosinus, Dice, cf. Abschnitt 2.1.3) setzt es nicht voraus, dass die zwei zu vergleichenden Vektoren gleich aufgebaut sind und gleich viele Elemente besitzen. Somit lässt sich COSIMIR direkt auf heterogene Daten anwenden. Die beiden Input-Dokumente (oder Dokument und Anfrage) gehen in heterogenen Repräsentationen ins Netz ein. Das Netz lernt anhand von Beispielen, wie sich die Ähnlichkeit aus den einzelnen Elementen errechnet. Ein expliziter Transformationsschritt ist nicht erforderlich. Voraussetzung sind allerdings genügend Trainingsdaten, also Relevanzbewertungen zu Dokumenten in heterogenen Repräsentationen (cf. Abschnitt 6.4.4).

Das COSIMIR-Modell und seine Adaption für heterogene Daten wird in Kapitel 6 eingeführt und die Evaluierungsergebnisse stellt Kapitel 7 vor.

5.4 Fazit: Heterogenität und ihre Behandlung im Information Retrieval

Informationssysteme müssen u.a. aufgrund der weltweiten Vernetzung immer häufiger heterogene Daten integrieren, um für den Benutzer tolerant zu wirken. Die Heterogenität von Erschließungsverfahren führt v.a. zu semantischen Problemen. Die Ansätze zur Behandlung von Heterogenität lassen sich in sich in exakte und vage Verfahren unterteilen. Die exakten, deduktiven Verfahren wie etwa Konkordanzen zwischen Begriffssysteme erfordern einen hohen intellektuellen Aufwand, der nicht in allen Anwendungsfällen erbracht wird.

Soweit möglich, soll für die Transformationen formalisiertes menschliches Wissen in Form von Konkordanzen, Thesauri und deduktiven Regeln ausgenutzt werden. Wo dieses Wissen nicht mehr ausreicht oder überhaupt nicht zur Verfügung steht, was in der Praxis sehr häufig der Fall ist, greifen vage

Verfahren ein, die auch Wissen geringerer Sicherheit nutzen. Beispiele hierfür sind neuronale Netze und statistische Verfahren, die im Information Retrieval bereits erfolgreich eingesetzt werden.

Die folgende Übersicht fasst die Verfahren für vage Transformationen zusammen:

- Statistische Verfahren
    - Familie ähnlicher Verfahren basierend auf Kookkurrenzen
    - statistische Verfahren im IR allgemein sehr erfolgreich
    - für Transformationen in realen Anwendungen bereits erfolgreich eingesetzt
- Hopfield- und Spreading-Activation-Netzwerke
    - einfache neuronale Netze ohne sub-symbolische Fähigkeiten
    - den statistischen Verfahren sehr ähnlich
- Transformations-Netzwerk
    - basiert auf dem Backpropagation-Algorithmus
    - eine experimentelle Anwendung für Transformationen innerhalb des gleichen Vokabulars bekannt
- COSIMIR-Netzwerk für Transformationen
    - experimentelles IR-Verfahren basierend auf dem Backpropagation-Algorithmus
    - für Transformationen adaptiert
    - erste Experimente mit kleinen Datenmengen erfolgreich

Die statistischen Verfahren sind erprobt und in den meisten Projekten mit großer Sicherheit einsetzbar. Aber auch das Transformations-Netzwerk hat eine hohe Plausibilität und wurde bereits in kleinen Experimenten getestet. COSIMIR ist ein experimenteller Ansatz. In den Experimenten in Kapitel 7 werden das Transformations-Netzwerk und das COSIMIR-Modell für Heterogenität mit realen Daten getestet. Die Experimente mit dem Transformations-Netzwerk umfassen große und reale Text-Korpora.

6 Das COSIMIR-Modell

Das COSIMIR-Modell (COgnitive SIMilarity Learning in Information Retrieval) entstand aus der Analyse des state-of-the-art neuronaler Netze im Information Retrieval (cf. Kapitel 4). Es vermeidet die Schwächen bestehender Systeme und realisiert hohe Lernfähigkeit in einem Backpropagation-Modell. Eine Variante des COSIMIR-Modells eignet sich für die Behandlung heterogener Repräsentationen, die zunehmend eine Herausforderung für Information Retrieval Systeme werden (cf. dazu Kapitel 5).

Durch die gewählte Architektur implementiert das COSIMIR-Modell den zentralen Prozess im Information Retrieval, den *match* zwischen Anfrage- und Dokumentrepräsentation in einem einfachen Backpropagation-Netzwerk anhand von Beispielen. Fast alle nicht Booleschen IR-Systeme entscheiden sich aufgrund von bestimmten Heuristiken für eine Ähnlichkeitsfunktion. Diese Entscheidung ist beim COSIMIR-Modell nicht erforderlich.

Der folgende Abschnitt stellt den Kern des COSIMIR-Ansatzes vor. Abschnitt 6.2 beschreibt weitere denkbare Modelle, die Information Retrieval in der Architektur eines Backpropagation-Netzwerks abbilden. Sie haben jedoch einige formale Schwächen und eignen sich nur für Teilaspekte des gesamten Information Retrieval Prozesses. In einem erweiterten Modell, das in Abschnitt 6.4.3 vorgestellt wird, werden einige davon mit dem Kern des COSIMIR-Modells verbunden. Daneben zeigt Abschnitt 6.4 noch einige Erweiterungen und Modifikationen des COSIMIR-Grundmodells. Davon ist besonders das Modell für die Verarbeitung heterogener Repräsentationen interessant. Abschnitt 6.3 referiert einige Ansätze, die mit COSIMIR vergleichbar sind.

6.1 COSIMIR-Basismodell

Das COSIMIR-Modell realisiert den zentralen Prozess im Information Retrieval, den Abgleich zwischen Anfrage- und Dokument-Repräsentation in einem Backpropagation-Netzwerk. Input für das COSIMIR-Modell sind gleichzeitig eine Anfrage und ein Dokument. Über eine oder mehrere versteckte Schichten wird die Aktivierung bis zur Ausgangsschicht propagiert, die nur aus einem Neuron besteht und die Relevanz bzw. Ähnlichkeit zwischen den beiden Input-Objekten repräsentiert. Im Training wird die Relevanz von verschiedenen Kombinationen von Dokumenten und Anfragen gelernt.

6.1.1 Funktionsweise

Das COSIMIR-Modell greift auf eine durch Indexierung gewonnene Dokument-Term-Matrix zurück. Es basiert also wie die anderen in Kapitel 4 besprochenen Modelle auf einem Indexierungsverfahren aus dem Information Retrieval. Das COSIMIR-Modell setzt ein, sobald die Objekt-Repräsentationen vorhanden sind.

Der Kern jedes Retrievalprozesses besteht darin, auf Basis der in der Indexierung gewonnenen Repräsentationen ein Dokument und eine Anfrage zu vergleichen und die Ähnlichkeit zwischen beiden als Maß für die Relevanz zu berechnen. Für diesen Schritt wählt in der Regel ein Entwickler eines Information Retrieval Systems eine Ähnlichkeitsfunktion. Dafür kommen die mathematischen Ähnlichkeitsfunktionen wie Kosinus oder Dice in Frage (cf. Abschnitt 2.1.3). Die Entscheidung für eine bestimmte Ähnlichkeitsfunktion wird fast nie aus den Eigenschaften dieser Funktion begründet. Vielmehr fällt diese Entscheidung meist aufgrund heuristischer Überlegungen oder einer empirischen Überprüfung der Ergebnisse. An dieser Stelle im Information Retrieval Prozess setzt das COSIMIR-Modell an, in dem diese heuristische Entscheidung nicht mehr nötig ist.

Wie Abbildung 6-1 zeigt, werden an der Input-Schicht von COSIMIR gleichzeitig eine Dokument- und eine Anfrage-Repräsentation angelegt. Jedes Neuron repräsentiert dabei einen Term. Über eine oder mehrere Zwischen-Schichten breitet sich die Aktivierung aus und das Netz berechnet den Output. Die Output-Schicht besteht aus nur einem Neuron, das die Relevanz des Input-Dokuments für die Input-Anfrage repräsentiert.

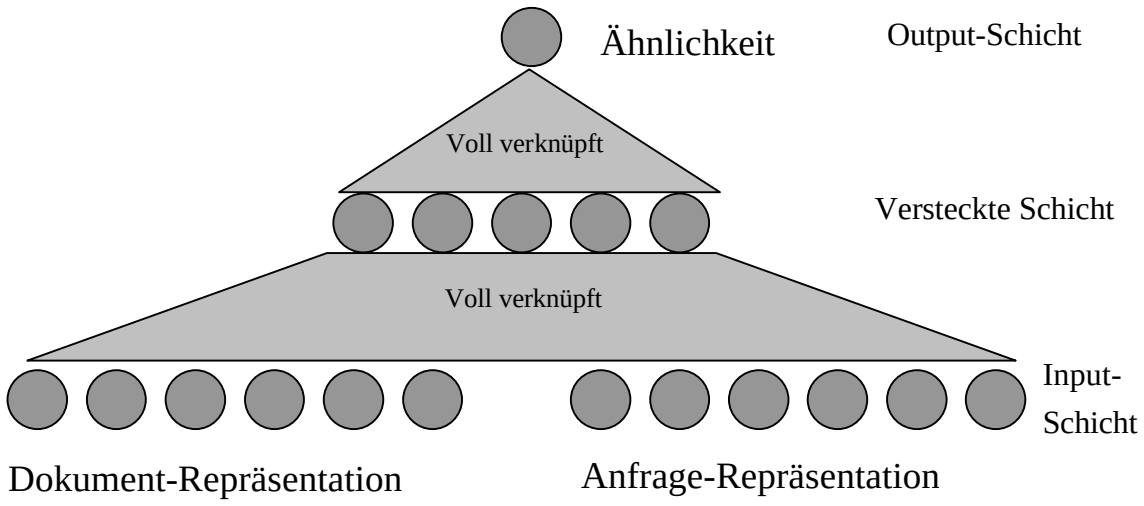


Abbildung 6-1: Das COSIMIR-Modell

Wie jedes Backpropagation-Netzwerk wird auch das COSIMIR-Modell zunächst trainiert und dann für neue Daten eingesetzt. Im der Trainingsphase lernt COSIMIR die Relevanz von verschiedenen Kombinationen von Dokumenten und Anfragen. Die von Benutzern vorgegebene Relevanz dient dabei als Maß für die Ähnlichkeit zwischen Anfrage und Dokument. Dafür benötigt das COSIMIR-Modell viele Trainingsdaten, die aus Urteilen von Benutzern über die Relevanz von Dokumenten zu Anfragen bestehen. Die Verbindungen, die von der Input-Schicht ausgehen, werden nur ausreichend trainiert, wenn auch jeder Term in zumindest einigen der Trainingsdokumente enthalten ist und somit in der Trainingsmenge vorkommt.

Benutzt man einen Thesaurus für die Dokumentrepräsentationen, so sollte also jeder Thesaurusterm in einem Dokument vorkommen. Der Thesaurus des Informationszentrum Sozialwissenschaften z.B. verfügt über ca. 6000 Begriffe. Bei Freitext für die Anfrageformulierung oder bei automatischer Indexierung sind die Repräsentationsvektoren noch größer. Die notwendige Größe von COSIMIR-Netzwerken für reale Datenbestände kann zu Problemen führen. Eine Lösungsstrategie, die auch für das Transformations-Netzwerk eingesetzt wird, ist die Komprimierung der Repräsentationen. Nach Vorstellung des Transformations-Netzwerks (cf. Abschnitt 6.2.1) wird die Komprimierung mit Latent Semantic Indexing in Abschnitt 6.4.2 besprochen.

Die Trainingsmenge muss auch Trainingsbeispiele für Nicht-Relevanz umfassen, da das System ansonsten lernt, in allen Fällen eine hohe Relevanz zu berechnen. Dadurch steigt die Zahl der potenziell verwendbaren Daten erheblich, da alle nicht relevanten Dokumente als negative Trainingsbeispiele benutzt werden können. Nicht relevante Dokumente und die dazugehörige Anfrage sollen im Output von COSIMIR zur Ähnlichkeit Null führen. Jedes Benutzerurteil schafft also zahlreiche Beispiele. Neben binären Urteilen über Relevanz können auch graduelle Urteile ausgenutzt werden, bei denen der Benutzer eine Relevanz zwischen Null und Eins vergibt. Die zu lernende Ähnlichkeit ist dann nicht Eins oder Null sondern ein Wert dazwischen.

In dieser Phase destilliert COSIMIR das Wissen aus den Ähnlichkeitsurteilen und speichert es in den Verbindungsstärken. Darin steckt das Wissen, wie das Vorkommen oder Nicht-Vorkommen von Termen die Ähnlichkeitsbewertung beeinflusst. Um in der Einsatzphase die Ähnlichkeit möglichst genau berechnen zu können, muss COSIMIR generalisieren und die Ähnlichkeitsfunktion von den Trainingsbeispielen auf die Beispiele im Einsatz übertragen.

In der Einsatzphase formuliert der Benutzer eine neue Anfrage. Nach Erstellung des Repräsentationsvektors erhält das COSIMIR-Modell die Anfrage und je ein Dokument als Input. Für jedes in der Datenbasis enthaltene Doku-

ment berechnet COSIMIR die Ähnlichkeit zur Anfrage. Die ähnlichsten Dokumente werden dem Benutzer als Ergebnis präsentiert.

Da das COSIMIR-Modell zahlreiche Benutzerurteile erfordert, wird es typischerweise für eine Benutzergruppe erstellt. Die Urteile verschiedener Benutzer werden also in der Trainingsmenge zusammengefasst und COSIMIR erstellt auf diese Weise ein Ähnlichkeits-Modell über zahlreiche Benutzer und ihre Interessen. Daneben ist es auch denkbar, dass ein einzelner Benutzer sein eigenes personalisiertes COSIMIR-Modell trainiert.

6.1.2 Wissensbasis

Das COSIMIR-Modell greift auf folgende im Information Retrieval vorhandene Wissensquellen zu:

- Die übliche und bewährte Repräsentation einer Kollektion in der Dokument-Term-Matrix. Dabei kann COSIMIR sowohl binäre als auch gewichtete Repräsentationen verarbeiten.
- Urteile von Benutzern über Relevanz von Dokumenten zu Anfragen

Die Benutzerurteile nutzen nur Information Retrieval Verfahren, die Relevanz-Feedback einsetzen (cf. Abschnitt 2.3.1.1). Dabei werden die Informationen meist anders eingesetzt als in COSIMIR. Manche Relevanz-Feedback-Verfahren werten die Informationen des Benutzers für die Anfrage-Optimierung aus und verwerfen sie dann wieder, während andere Systeme daraus lernen. Kwok 1989 etwa modelliert in seinem Spreading-Activation-Netzwerk, die Relevanz-Feedback Information als Aktivierung von Dokument-Neuronen durch den Benutzer. Das vorgesehene Lernverfahren soll nun die Gewichtungen der Verbindungen des Spreading-Activation-Netzwerks so verändern, dass die vom Benutzer positiv bewerteten Dokumente bei erneuter Eingabe der Anfrage eine höhere Aktivierung vom System erhalten. Da die Verbindungsgewichte die bei der Indexierung gewonnenen Gewichte der Dokument-Term-Matrix enthält, verändert das Lernverfahren die Repräsentation der Dokumente im System. Da sich die Dokumente natürlich nicht geändert haben und für jede Anfrage eine andere Repräsentation optimal sein kann, ist dieses Vorgehen problematisch. Die beiden oben genannten Wissensquellen werden bei derartigen lernenden Systmen vermischt. Die Ausgangsdaten in Form der Dokument-Repräsentationen und die Einflüsse des Lernens aufgrund der Benutzerurteile lassen sich nachträglich nicht mehr unterscheiden.

Das COSIMIR-Modell dagegen trennt diese beiden Informationsquellen. Zum einen belässt es die Dokument-Term-Matrix als Repräsentation der Kollektion

unverändert. Die Repräsentation eines Dokuments bleibt also immer gleich und da sich das Dokument ja auch nicht ändert, erscheint dies angemessen. Die Benutzerurteile bilden die Grundlage für das Lernen der Ähnlichkeitsfunktion auf Basis der jeweiligen Dokument-Term-Matrix. Die Benutzerurteile zeigen dem System gewissermaßen, wie die Ausgangsdaten zu bearbeiten sind, ohne diese zu verändern. Während also die Repräsentation von Dokumenten und Anfragen gleich bleiben, kann das lernende Netzwerk daraus zu verschiedenen Zeitpunkten unterschiedliche Ergebnisse ableiten. Die Ähnlichkeit eines Dokument-Anfrage-Paares kann sich während des Lernprozesses ändern. Wie Abbildung 6-2 zeigt, wirkt die Adaptivität bei COSIMIR also nicht auf der Ebene der Objektrepräsentationen, sondern in der Ähnlichkeitsberechnung.

6.1.3 Kognitive Ähnlichkeitsfunktion

Die Trennung der Wissensquellen bei COSIMIR scheint zunächst eine rein formale Eigenschaft zu sein, sie hat aber erhebliche Konsequenzen. Während bei Information Retrieval Verfahren mit Relevanz-Feedback die Urteile des Benutzers meist ausschließlich die Gewichte der Anfrage-Terme und nicht das System verändern, trägt COSIMIR das Wissen des Benutzers in den Kern des Systems, wo sonst mathematische Modelle vorherrschen. Durch die Relevanzurteile der Benutzer lernt das COSIMIR-Modell, eine kognitive Ähnlichkeitsfunktion zu implementieren. Die Verbindungsstärken eines COSIMIR-Netzwerks speichern die Urteile des Benutzers in Form von Wissen über die potenziell sehr komplexen Zusammenhänge zwischen dem Vorkommen von Termen in Anfragen und Dokumenten und den daraus resultierenden Einschätzungen der Ähnlichkeit.

Ein weiteres Argument dafür, dass COSIMIR eine kognitiv adäquatere Ähnlichkeitsfunktion implementiert als traditionelle Information Retrieval Systeme, liefert die Psychologie. Die Untersuchungen von Tversky 1977 (cf. auch Abschnitt 2.1.3) beschäftigen sich mit der menschlichen Bewertung von Ähnlichkeit. In verschiedenen Experimenten erwies sich die kognitive Ähnlichkeitsfuktion als äusserst komplex. Demnach nehmen Menschen Ähnlichkeit weder als transitiv noch als symmetrisch wahr. Fast alle mathematischen Ähnlichkeitsfuktionen erfüllen aber diese Bedingungen und sind damit stark eingeschränkt. Sie können das menschliche Vorbild nur unzureichend abbilden. Und die Sichtweise des Benutzers soll für ein Information Retrieval System natürlich im Vordergrund stehen.

Das COSIMIR-Modell dagegen ist nicht auf transitive oder symmetrische Funktionen beschränkt. Je nach Trainingsdaten kann das neuronale Back-

propagation-Netzwerk eine symmetrische oder eine nicht symmetrische, ebenso wie eine transitive oder eine nicht transitive Funktion implementieren.

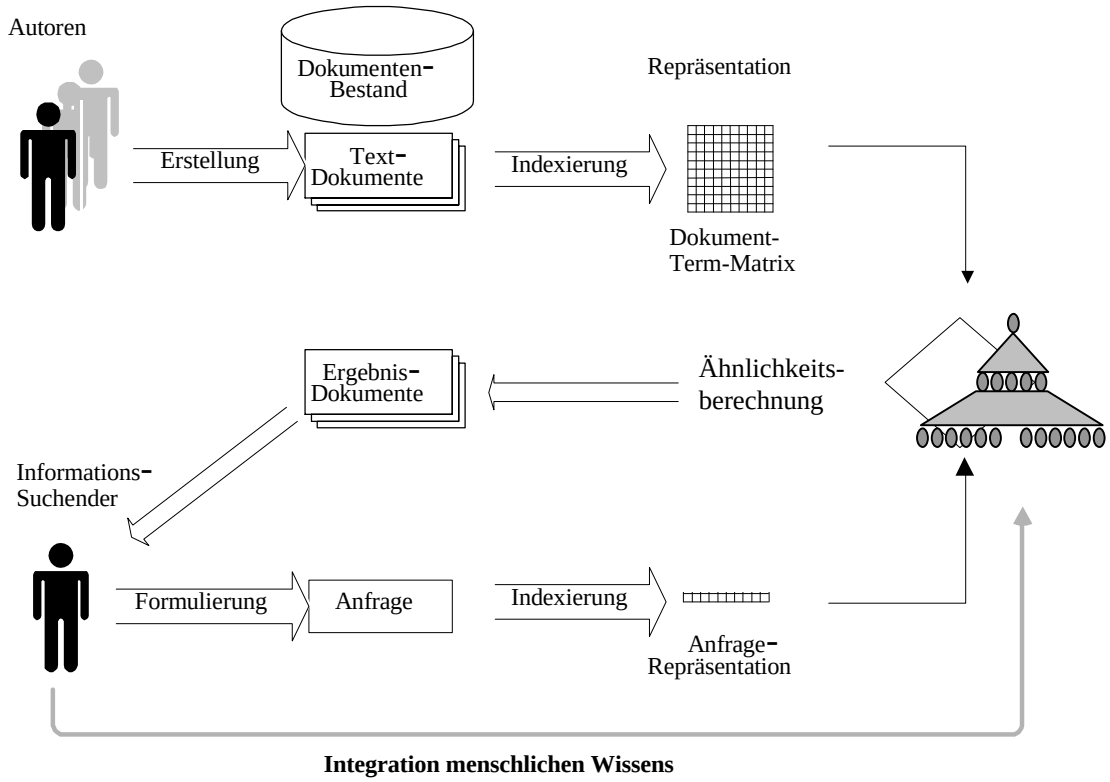


Abbildung 6-2: Das COSIMIR-Modell im Information Retrieval Prozess

Viele Information Retrieval Modelle erfordern aus formalen Gründen die paarweise Unabhängigkeit von Termen; eine Forderung, welche die Daten in den meisten Fällen offensichtlich nicht erfüllen. So wird in einem Dokument mit dem Term *Auto* der Term *PKW* mit höherer Wahrscheinlichkeit vorkommen als der Term *Elefant*. Das probabilistische Modell nimmt paarweise Unabhängigkeit zwischen Termen an, um die Komplexität bei der Berechnung bedingter Wahrscheinlichkeiten zu vermeiden. Ohne diese Annahme wäre das Modell nicht implementierbar. COSIMIR verzichtet auch auf diese Annahme und modelliert im Idealfall wie oben diskutiert die komplexen Zusammenhänge und Abhängigkeiten im Backpropagation-Netzwerk. Ebenso tragen in traditionellen Systemen alle Terme mit der gleichen Wertigkeit zum Ergebnis bei. Je nach Anfrage können manche Terme natürlich eine geringere Bedeutung und weniger Einfluss auf die menschliche Entscheidung haben. Das COSIMIR-Modell macht auch über die Wertigkeit der Terme keine Vorannahmen und greift lediglich auf die Benutzerurteile zu. Durch die Generalisierungsfähigkeit von Backpropagation-Netzwerken können die Zusammen-

hänge und Abhängigkeiten, die das Modell während der Trainingsphase lernt, auf andere Anfragen und Dokumente übertragen werden.

Neuronale Netze basieren auf einem einfachen Modell der menschlichen Kognition und ahmen die in vielen Bereichen menschliche kognitive Fähigkeiten besser nach als traditionelle Verfahren der Künstlichen Intelligenz (cf. Kapitel 3). Durch die Wahl des mächtigen Backpropagation-Algorithmus, der bisher kaum für Information Retrieval eingesetzt wurde, kann das COSIMIR-Modell formal mehr Klassen von Funktionen implementieren als z.B. ein Spreading-Activation-Netzwerk. Durch die versteckte Schicht von Neuronen und das dazugehörige Lernverfahren sind Backpropagation-Netzwerke leistungsfähiger als Netze ohne versteckte Schicht (cf. Abschnitt 3.5.4.1, cf. Zell 1994:100f.). Die Neuronen der versteckten Schicht haben keine symbolische Bedeutung, sie verweisen nicht auf Eigenschaften der beteiligten Objekte. Damit können sie auch vom Benutzer nicht interpretiert werden. Gerade diese sub-symbolische Repräsentationen erhöhen die Leistungsfähigkeit der Backpropagation-Netzwerke. Das Fehlen sub-symbolischen Repräsentationen erkennen z.B. Crestani/van Rijsbergen 1997 als Schwäche der Spreading-Activation-Systeme.

6.1.4 Gewinnung von Trainingsdaten

Das COSIMIR-Modell erfordert viele Trainingsdaten, die in realen Kontexten nicht ohne größeren Aufwand und die damit verbundenenen Kosten zur Verfügung stehen. Grundsätzlich müssen Urteile von Benutzern zur Relevanz von Ergebnis-Dokumenten zu ihren Anfragen gesammelt werden. Einige heuristische Verfahren tragen zur Erhöhung der Anzahl der verwendbaren Trainingsbeispiele bei.

- Den Input bilden zwei Vektoren, die beide die gleichen Terme repräsentieren. An einem wird die Aktivierung für die Anfrage angelegt und an dem anderen die des Dokuments. Die Rollen dieser Objekte sind austauschbar, auch das Dokument kann eine Anfrage sein, die auf ähnliche Dokumente zielt. Geht man davon aus, dass weitgehend Symmetrie besteht, sollte ein Dokument als Anfrage das gleiche Ergebnis im Vergleich zu einer Anfrage erzielen wie umgekehrt. Nutzt man diese Heuristik und akzeptiert die damit verbundene Unterstellung einer symmetrischen Ähnlichkeitsfunktion, verdoppelt sich die Anzahl der potenziellen Trainingsbeispiele. Ein Paar erscheint dann sowohl der Reihenfolge Dokument-Anfrage als auch in der Reihenfolge Anfrage-Dokument im Input.
- An der Input-Schicht von COSIMIR können die Objekte also die Reihenfolge wechseln. Als weitere Verallgemeinerung kann auch zweimal der

gleiche Objekttyp den Input bilden. In der Regel gilt, dass Identität den höchsten Grad der Ähnlichkeit darstellt. Somit kann jedes Objekt ein Trainingsbeispiel mit der Ähnlichkeit zu sich selbst bilden. Am COSIMIR-Netzwerk muss es lediglich zweimal parallel die Input-Neuronen aktivieren.

- Besteht die Anfrage aus einem Dokument, dann werden ähnliche Dokumente gesucht. Akzeptiert man die heuristische Annahme, dass Dokumente, die zu einer Anfrage relevant sind, untereinander sehr ähnlich sind, entstehen zusätzliche Trainingsbeispiele. Sind z.B. zu einer Anfrage n relevante Dokumente bekannt, dann führt dies im Standard-Modell zunächst zu n positiven Trainingsbeispielen. Geht man davon aus, dass die n Dokumente auch untereinander sehr ähnlich sind, dann kann jedes Dokument als eine Anfrage betrachtet werden, die als Ergebnis die anderen n-1 Dokumente liefern soll. Damit entstehen n (n-1) weitere Trainingsbeispiele.

6.1.5 Tolerantes Information Retrieval

Das COSIMIR-Modell ist in mehrerer Hinsicht ein tolerantes Information Retrieval Verfahren. Zum einen wurde mit neuronalen Netze bereits für die Modellierung eine tolerante Technik gewählt, die wenig störungs- und fehleranfällig sind. So führen kleine Veränderungen an den Eingangs-Daten in der Regel nicht zu großen Veränderungen am Ergebnis. Beim Lernprozess toleriert COSIMIR daher unterschiedliche und gar widersprüchliche Urteile über Ähnlichkeit. Das System formt daraus eine umfassende Ähnlichkeitsfunktion, die möglichst viele Aspekte und Benutzerstandpunkte integriert. Bei widersprüchliche Aussagen setzt sich je nach Datenlage entweder die häufiger vorkommende Variante oder die besser zu dem Rest der Daten passende Variante durch. Wichtig ist, dass diese Probleme nicht zu einem Scheitern des gesamten Verfahrens führen, sondern vom Backpropagation-Netzwerk verarbeitet werden.

Vor allem toleriert COSIMIR, wie oben diskutiert, die Eigenschaften der menschlichen Ähnlichkeitsbewertung und versucht nicht, diese mit inadäquaten Methoden wie den bekannten, einfachen Ähnlichkeitsfunktionen zu modellieren.

Ein weiterer entscheidender Vorteil von COSIMIR ist, dass es im Input auch Vektoren verschiedener Länge vergleichen kann. Dies ist interessant für die Behandlung heterogener Repräsentationen im Information Retrieval (cf. Kapitel 5). Dokument- und Anfrage-Vektor können somit in verschiedenen Repräsentationsschemata oder –sprachen vorliegen solange nur genügend Trainingsdaten vorhanden sind. Diesen Aspekt behandelt Abschnitt 6.4.4.

COSIMIR ist auch als generelles Ähnlichkeitswerkzeug einsetzbar. Es macht keinerlei Annahmen über die formalen Eigenschaften der Ähnlichkeit. Weder Transitivität noch Symmetrie sind Voraussetzungen für COSIMIR. Solange die Ähnlichkeitsfunktion im jeweiligen Anwendungsfall für ein Backpropagation-Netzwerk überhaupt erlernbar ist, kann sie beliebige formale Eigenschaften besitzen.

6.2 Backpropagation-Architekturen für Information Retrieval

Dieser Abschnitt fasst weitere Möglichkeiten für den Einsatz des Backpropagation-Netzwerks im Information Retrieval zusammen. Die Diskussion dieser möglichen Modelle zeigt, dass gerade die Architektur des COSIMIR-Modells sich gut für Information Retrieval eignet. Das Backpropagation-Netzwerk leistet Abbildungen zwischen verschiedenen Term-Räumen und ist damit ein hetero-assoziatives Modell. Dieser Abschnitt zeigt die Möglichkeiten, diese Eigenschaft für die Anforderungen des Information Retrieval zu nutzen. Die bereits in Abschnitt 4.6 diskutierten und bereits realisierten Ansätze zur Verwendung von Backpropagation im Information Retrieval werden den Architekturen zugeordnet.

Die Neuronen der Modelle bilden in verschiedenen Varianten die am Information Retrieval Prozess beteiligten Objekte wie Anfragen, Dokumente und Terme ab. Der entscheidende Vorteil der Architektur von COSIMIR liegt in der Berücksichtigung und Repräsentation der Relevanz in einem eigenen Neuron.

Alle diese Modelle bauen wie COSIMIR auf bestehenden Information Retrieval Methoden auf. Sie setzen ein Indexierungsverfahren voraus und die Dokumente und Anfragen werden als Vektoren dargestellt. Die Anzahl der Elemente der Vektoren ist identisch mit der Anzahl der vorkommenden Terme und jedes Vektorelement enthält ein Gewicht, das die Häufigkeit des Vorkommens und damit die Wichtigkeit des Terms für Anfrage oder Dokument ausdrückt.

6.2.1 Transformations-Netzwerk

Das Transformations-Netzwerk modelliert nicht den zentralen Prozess im Information Retrieval, sondern transformiert die Repräsentation eines Dokuments zwischen verschiedenen Term-Räumen, die z.B. durch unterschiedliche Indexierungverfahren entstehen. Erst nach der Transformation beginnt in der Regel mit dem Vergleich zwischen einer Anfrage und den Dokumenten der

entscheidende Teile des Retrieval-Prozesses. Einen Einblick in den Stand der Forschung zum Transformations-Netzwerk bietet Abschnitt 4.6.3.

Das Transformations-Netzwerk kann im Bereich der Heterogenitätsbehandlung eingesetzt werden und zwischen verschiedenen Term-Räumen abbilden. Die Problematik und Behandlung von Heterogenität im Information Retrieval behandelt Kapitel 5, wobei Abschnitt 5.3.4 das Transformations-Netzwerk in den Kontext der verschiedenen Verfahren zur Heterogenitätsbehandlung stellt. Das Transformations-Neztwerk erlaubt z.B. die Abbildung automatisch indexierter Dokumente auf ein intellektuelles Indexierungsschema. Die intellektuelle Indexierung wird dadurch maschinell nachgebildet. Ein derartiges Modell erhält als Input eine automatisch erstellte Repräsentation eines Dokuments. Der Input-Vektor verfügt über eine Dimension für jeden in der Kollektion vorkommenden Term. Während des Trainings und zur Laufzeit wird an der Input-Schicht je ein Dokument-Vektor angelegt. An jedes Neuron wird der Wert der jeweiligen Dimension in dem Dokument abgetragen. Dabei kann es sich z.B. je nach Vorkommen des Begriffs um Null oder Eins handeln, aber auch um reelle Werte aus dem Intervall zwischen Null und Eins, die z.B. die inverse Dokument-Frequenz repräsentieren.

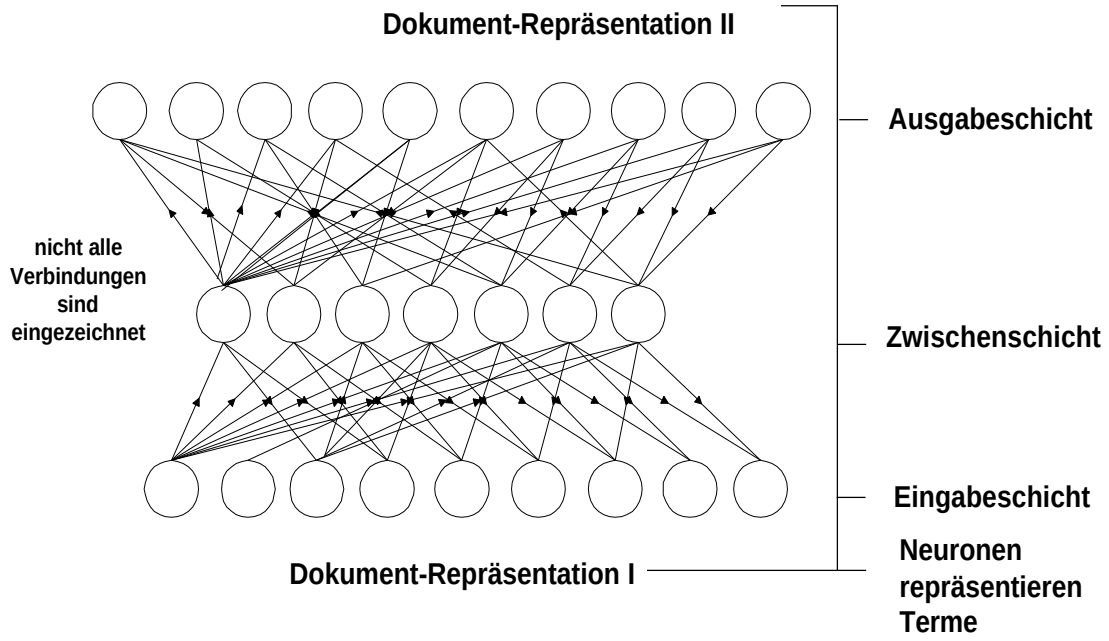


Abbildung 6-3: Transformations-Netzwerk

Wie Abbildung 6-3 zeigt, besteht der Output des Netzes in einer Repräsentation des gleichen Objekt innerhalb eines anderen Indexierungsschemas und damit eines anderen Term- oder Merkmalsraum. Der Output-Vektor besitzt

eine Dimension für jeden Term im zweiten Indexierungsschema. Dies kann z.B. ein Thesaurus mit kontrollierten Termen sein. In der Trainingsphase lernt das Netz die Transformations-Funktion zwischen den beiden Term-Räumen anhand von Beispielen, für die beide Repräsentationen vorhanden sind. Das Korpus muss also mit zwei Verfahren indexiert worden sein und alle Trainings-Dokumente liegen in beiden Merkmalsräumen vor. Die Erstellung eines Doppelkorpus erfordert oft einen hohen Aufwand. Nach Möglichkeit sollten bei der Planung eines Transformation-Netzwerks vorhandene Daten erschlossen werden.

Anhand der zweifach indexierten Dokumente des Doppelkorpus lernt das Transformations-Netzwerk die komplexen Zusammenhänge zwischen dem Vorkommen von Termen in beiden Term-Mengen. Beim erfolgreichen Training erwirbt das Netz die Fähigkeit zu generalisieren und die Funktion auch auf bisher unbekannte Dokumente anzuwenden. Damit kann es nach dem Training in der Einsatzphase ausgehend von einer Repräsentation die andere erstellen.

Ein weiterer wichtiger Anwendungsfall für das Transformations-Netzwerk ist die Abbildung zwischen zwei intellektuell erstellten Repräsentationen mit unterschiedlichen Thesauri. Bibliotheken oder Informationsservicestellen schließen sich heute mehr und mehr zu digitalen Bibliotheken zusammen, um so ihren Benutzern den Zugriff auf größere Datenmengen zu erleichtern. Oft benutzt ein Informationsanbieter einen eigenen Thesaurus, an den sich die Benutzer gewöhnt haben. Durch automatische Transformationen werden die Dokumente anderer Bibliotheken oder Informationsservicestellen mit dem eigenen Thesaurus zugänglich gemacht. Voraussetzung ist ein Doppelkorpus, den die Informationsanbieter bei einer Überlappung der Korpora eventuell in den eigenen Daten vorfinden. Falls dies nicht der Fall ist, kann eine Menge gezielt zusätzlich erschlossen werden. Auf diesen Anwendungsfall zielen die Experimente mit dem Transformations-Netzwerk in Abschnitt 7.2 und Abschnitt 7.3.

Diese Experimente sind erforderlich, da die bisher durchgeführten Experimente von Crestani/van Rijsbergen 1997 und Crestani 1995 nur auf sehr kleinen Datenmengen basierten. Zudem besteht der Anwendungsfall bei Crestani/van Rijsbergen 1997 und Crestani 1995 etwas anders gelagert und besteht in der Transformation zwischen identischen Repräsentationen oder Term-Räumen. Ziel ist es, die Anfrage auf eine durch Relevanz-Feedback abzubilden (cf. Abschnitt 4.6.3). Damit stellt der hier diskutierte Einsatz des Transformations-Netzwerks eine Verallgemeinerung des Ansatzes von Crestani/van Rijsbergen 1997 und Crestani 1995 dar.

Das Transformations-Netzwerk ist ein vielversprechender Ansatz für die Heterogenitätsbehandlung, der auf dem Backpropagation-Algorithmus basiert und aufgrund dessen Mächtigkeit auch komplexe, nicht linear trennbare Probleme erlernt (cf. Abschnitt 3.5.4.1). In den Abschnitten 7.2 und 7.3 wird experimentell überprüft, inwieweit diese Eigenschaften zu einer guten Qualität der Transformation führt.

6.2.2 Anfrage-Dokumenten-Vektor-Modell

Ein weiteres mögliches Information Retrieval Modell mit dem Backpropagation-Netzwerk ist das Anfrage-Dokumenten-Vektor-Modell, das z.B. von Mori et al. 1990 vorgeschlagen wird (cf. auch Abschnitt 4.6.2). Dieses Modell implementiert den zentralen Schritt im Retrievalprozess, indem es den Spreading-Activation-Ansatz einfach durch die Einführung von verstecketen Schichten zu einem Backpropagation-Netzwerk erweitert und dadurch weitere Schwächen dieses Ansatzes zeigt. Es besteht aus einer Anfrage-Schicht als Input und einer Dokument-Schicht als Output. In der Anfrage-Schicht repräsentiert jedes Neuron einen Thesaurus-Term und in der Output-Schicht ein Dokument. Die Input- und Output-Schicht entsprechen den Spreading-Activation-Netzwerken wie etwa dem Modell von Layaida 1994, die Abschnitt 4.3 ausführlich diskutiert. Mit Input- und Output-Schicht ist nicht nur die Architektur teilweise identisch, sondern auch die gewünschte Abbildung von Termen und damit Eigenschaften auf Objekte wie Abbildung 6-4 zeigt.

Zwischen diesen beiden Schichten liegt jedoch zusätzlich eine versteckte Schicht. Wie bei den Spreading-Activation-Netzwerken gibt der Benutzer seine Anfrage ein, indem er die Thesaurus-Terme gewichtet. Die Gewichtung kann mit den binären Werten Null oder Eins aber auch mit reellen Zahlen erfolgen. Das System propagiert diese Aktivierung durch das Netz und aktiviert in der Output-Schicht die Dokumente entsprechend ihrer Relevanz. Als Lerndaten werden von Experten oder Benutzern als optimal eingeschätzte Kombinationen von Anfragen und Ergebnis-Dokumenten eingesetzt. Nach der Generalisierung soll das Netz aus dem Vorkommen von Termen in der Anfrage auf relevante Dokumente schließen.

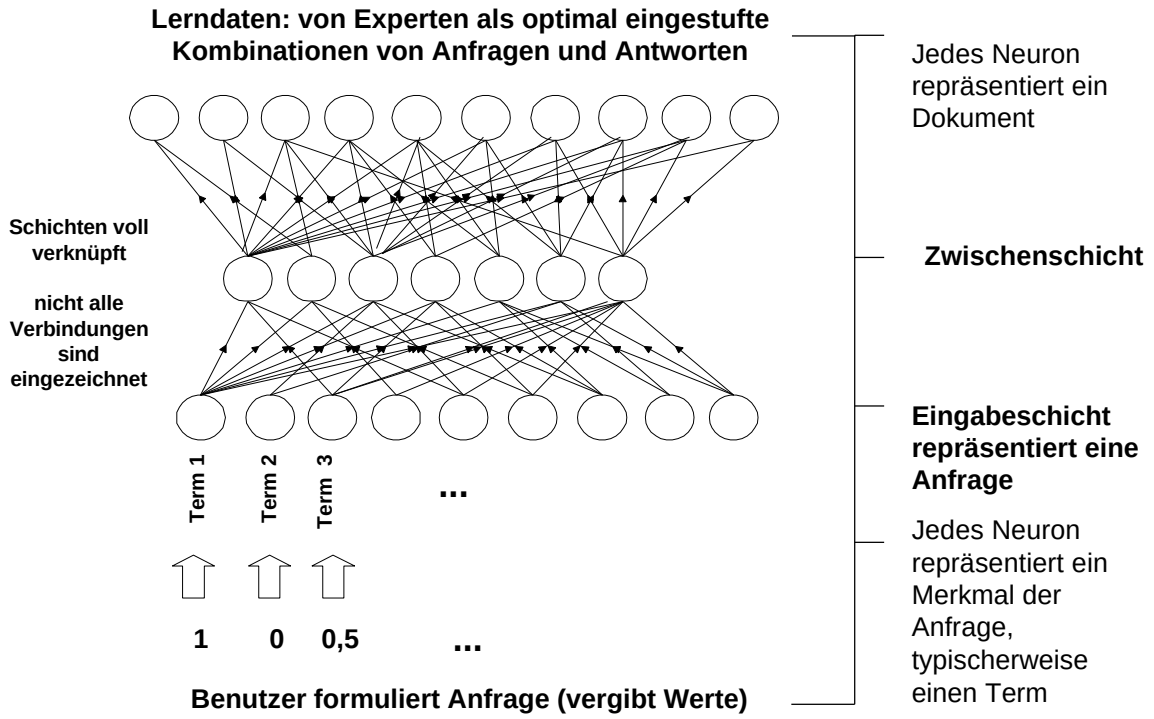


Abbildung 6-4: Anfrage-Dokumenten-Vektor-Modell

Ein genauerer Blick auf das Modell zeigt, dass die weitgehende Identität der Input- und Output-Schicht und damit der primären Abbildung zunächst irreführt und dass sich das Anfrage-Dokumenten-Vektor-Modell und das Spreading-Activation-Netzwerk beim Ablauf des Retrieval-Prozesses doch stark unterscheiden. Die Möglichkeit der Spreading-Activation-Netzwerke, die Aktivierung beliebig oft zwischen der Term- und Dokument-Schicht hin und her laufen zu lassen und so automatisch eine Term-Erweiterung zu erzielen, besteht beim Anfrage-Dokumenten-Vektor-Modell aufgrund des zugrundeliegenden Backpropagation-Algorithmus nicht. Diese Netzwerk-Architektur erlaubt nur eine einmalige Aktivierungsausbreitung in Richtung der Output- und damit der Dokument-Schicht. Durch die Einführung der versteckten Schicht und damit des Backpropagation-Algorithmus geht allgemein der Vorteil der großen Flexibilität der Spreading-Activation-Netzwerke verloren. Es entsteht ein Feed-Forward-Netz, bei dem Aktivierungsausbreitung von der Dokument-Schicht in die Anfrage-Schicht nicht definiert ist. Im Einsatz kann weder ein Dokument als Eingabe dienen, noch kann Relevanz-Feedback nur durch Aktivierungsausbreitung realisiert werden. Beides ergibt sich im Spreading-Activation-Netzwerken automatisch. Das Backpropagation-Netzwerk kann davon lediglich Relevanz-Feedback durch Lernen realisieren.

Ein weiteres Problem bei diesem Modell besteht in der Größe der Output-Schicht, die zu einer großen Anzahl von notwendigen Trainingsdaten führt.

Da jedes Dokument in der Output-Schicht durch ein Neuron vertreten ist, bestehen für jedes Dokument zahlreiche Verbindungen zur versteckten Schicht. Um dem Netz zu ermöglichen, diese Verbindungen richtig einzustellen, müßte jedes Dokument mindestens einmal in einer Antwortmenge vorkommen. Diese Menge an Trainingsdaten kann für eine reale Datenbank nicht beschafft werden. So beinhaltet die Datenbank SOLIS des Informationszentrum Sozialwissenschaften etwa 200.000 Dokumente aus dem Bereich der Sozialwissenschaft (cf. Zimmer 1998, 1998a). Diese Kritik trifft teilweise auch die Spreading-Activation-Modelle für Information Retrieval, in denen es eine Dokument-Schicht gibt. Der Nachteil wirkt sich dort allerdings nicht gravierend aus, da die Verbindungsstärken direkt eingestellt werden und nicht von Grund auf aus Trainingsbeispielen gelernt werden müssen wie es beim Backpropagation-Netzwerk der Fall ist.

Das Anfrage-Dokumenten-Vektor-Modell greift nicht auf die Eigenschaften der Dokumente zu, es kommt deshalb ohne Indexierung und ohne die Beschreibung der Dokumente durch Merkmale (Terme) aus. Es kann ausschließlich mit Trainingsdaten in der Form von Paaren aus Anfrage-Termen und Dokumenten arbeiten. Da die Eigenschaften der Dokumente und damit die Indexierung oder Inhaltserschließung eine entscheidende Rolle in jedem Information Retrieval Verfahren spielen, erscheint das Anfrage-Dokumenten-Vektor-Modell als kaum sinnvoll. Auch lernt dieses Netz nichts über die Eigenschaften der Dokumente, sondern lediglich, wie die Dokumente auf die Aktivierung in der Anfage-Term-Schicht reagieren.

Daneben ergeben sich erhebliche Probleme beim Update, da bei jedem neuen Dokument ein neues Neuron hinzugefügt werden muss. Damit muss das Netz neu trainiert werden und vor allem aber muss auch das neue Dokument bereits in einem Trainingsbeispiel vorhanden sein. In der Praxis ist diese Forderung unrealistisch. Durch die Generalisierungsfähigkeit sollen gerade auch Dokumente gefunden werden, die relativ neu sind und noch nicht in von Benutzern vorgegebenen Beispielen enthalten sind.

Die Probleme des Anfrage-Dokumenten-Vektor-Modells rühren größtenteils aus der Repräsentation von Objekten in der Output-Schicht her. Diese Grundidee widerspricht auch der verteilten Repräsentation, die zu den Stärken der neuronalen Netze zählt. Im Anfrage-Dokumenten-Vektor-Modell repräsentiert ein Neuron ein einzelnes von möglicherweise Millionen Dokumenten. Das Modell im Folgenden Abschnitt reagiert auf diese Schwäche und führt in der Output-Schicht eine verteilte Repräsentation der Dokumente ein.

6.2.3 Anfrage-Dokument-Profil-Modell

Das Anfrage-Dokument-Profil-Modell ist eine weitere Möglichkeit, Information Retrieval in einem Backpropagation-Netzwerk abzubilden. Die Input-Schicht ist identisch mit der im Anfrage-Dokumenten-Vektor-Modell. Die Output-Schicht besteht aus Neuronen für Terme, die als Merkmalmuster ein Dokument repräsentieren. Damit wird die verteilte Repräsentation, die schon die Input-Schicht realisiert, auf den Output übertragen. Damit steht im Anfrage-Dokument-Profil-Modell nicht jedes Neuron für ein Dokument, sondern ein Aktivierungs-Vektor über die Terme steht für ein Dokument. Abbildung 6-5 zeigt die Architektur des Modells.

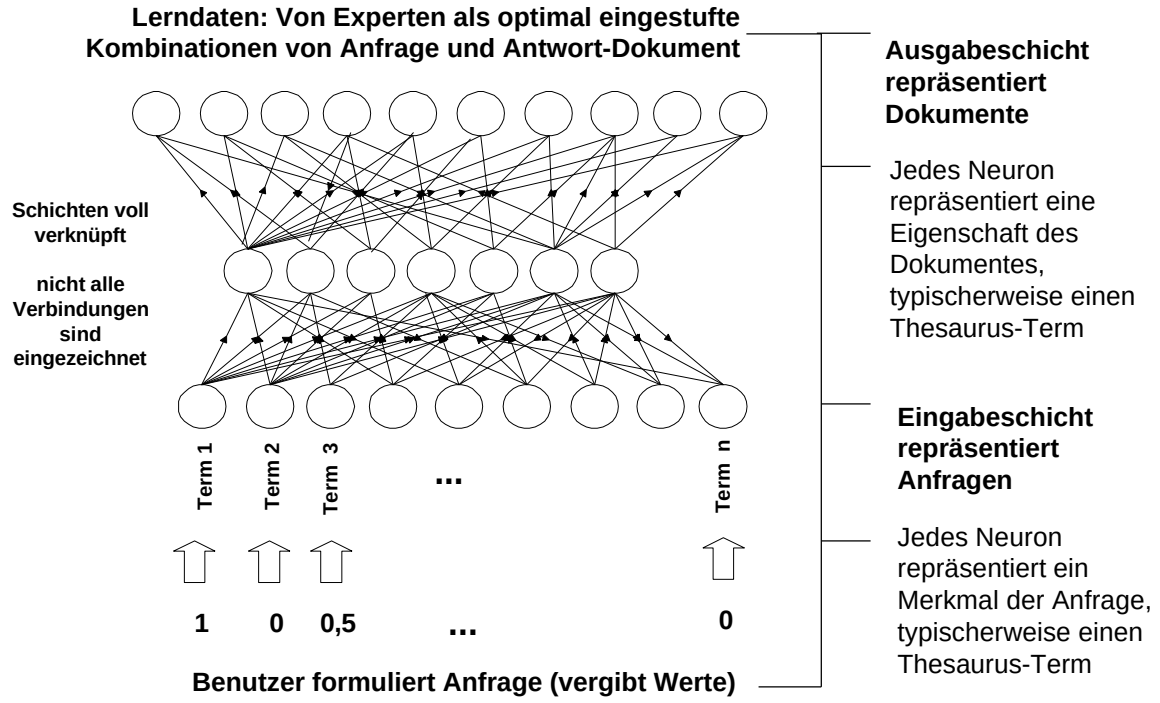


Abbildung 6-5**:** Anfrage-Dokument-Profil-Modell

Lerndaten bilden wieder optimale Kombinationen von Anfragen und Dokument. Dieses Modell lernt aber immer nur mit einem Dokument in der Output-Schicht. Gibt es also für eine Anfrage mehrere relevante Dokumente, was natürlich in der Praxis die Regel ist, so entstehen für diese Anfrage mehrere Trainings-Beispiele mit verschiedenen Dokumenten. Dies erhöht die Schwierigkeit der Lernaufgabe, da das Netz widersprüchliche Informationen erhält. Es lernt für den gleichen Input verschiedene gültige Output-Muster. Ein weiteres Problem besteht in der Integration gewichteter Relevanzurteile. Das Anfrage-Dokument-Profil-Modell kennt in dieser Form nur binäre Relevanzurteile. Jedes in der Trainingsmenge enthaltene Beispiel in Form eines Anfrage-Dokument-Paares ist gleichermaßen relevant, während alle nicht

enthaltenen Paare als nicht relevant gelten. Eine Möglichkeit, gewichtete Relevanzurteile einzubringen, besteht nur in der Variation der Präsentationshäufigkeit. Das Netz erhält die Anfrage-Dokument-Paare unterschiedlich häufig, wobei die Häufigkeit von der Relevanz für das Anfrage-Dokument-Paar abhängt. Dieses Verfahren könnte aber dazu führen, dass von den ohnehin unterschiedlichen Output-Mustern, die das Modell zu einem Input-Muster und damit einer Anfrage erhält, die am häufigsten präsentierten Beispiele die anderen beim Training völlig unterdrücken. Der Backpropagation-Algorithmus stellt die Verbindungen möglicherweise so ein, dass die weniger häufig präsentierten Beispiele überschrieben werden.

Das Hauptproblem des Anfrage-Dokument-Profil-Modells zeigt sich in der Einsatzphase. Nach dem Lernen errechnet das Netz aus einer Anfrage einen Aktivierungsvektor über alle Terme und liefert damit ein ideales Dokument. Output ist somit immer ein Dokument-Profil und kein Ranking aller Dokumente. Zum einen wünscht sich ein Benutzer in der Regel natürlich mehrere Dokumente als Ergebnis. Zum anderen ist nicht garantiert, dass dieses Dokument überhaupt in der Datenbasis existiert. Bei gewichteten Repäsentationsschemata mit reellen Zahlen ist es sogar äußerst unwahrscheinlich, dass der Output genau ein existierendes Dokument trifft. Im Einsatz müßten zum gefundenen, idealen Dokument-Profil in einem weiteren Schritt ähnliche Dokumente aus der Datenbasis gesucht werden. Ebenso kann der Output dieses Modells wieder als eine Anfrage betrachtet werden, die den Einstieg in einen Retrieval-Prozess bietet. Damit entspricht das Anfrage-Dokument-Profil-Modell dem ursprünglichen Einsatz des Transformations-Netzwerks bei Crestani/van Rijsbergen 1997, die eine Anfrage auf eine durch Relevanz-Feedback optimierte Anfrage abbildeten (cf. Abschnitt 4.6.3). Der Unterschied zum (in Abschnitt 6.2.1) vorgestellten Einsatz des Transformations-Netzwerks besteht darin, dass dort die Aufgabe in der Abbildung zwischen unterschiedlichen Repräsentationen liegt, während sowohl das Transformations-Netzwerk von Crestani/van Rijsbergen 1997 und das Anfrage-Dokument-Profil-Modell eine Abbildung innerhalb des gleichen Raumes von Merkmalen leisten. Durch die Generalisierung und Adaption auf die heterogenen Repräsentationen entsteht ein mächtiges Werkzeug für die Heterogenitätsbehandlung im Information Retrieval (cf. Abschnitt 5.3.5).

Entscheidend ist, dass die Architektur des Transformations-Netzwerks sich in Gestalt des Anfrage-Dokument-Profil-Modells nicht gut für die Modellierung von Information Retrieval Prozessen eignet.

6.2.4 Fazit: Backpropagation-Architekturen für Information Retrieval

Die vorgestellten Architekturen haben entscheidene Nachteile. Im Anfrage-Dokumenten-Vektor-Modell wirkt sich die lokale Repräsentation der Dokumente sehr nachteilig aus. Das Anfrage-Dokument-Profil-Modell bringt als Ergebnis lediglich ein Dokument und muss mit anderen Verfahren kombiniert werden. Dagegen lässt sich die gleiche Architektur für die Abbildung zwischen Merkmalsräumen sehr gut für heterogene Repräsentationen anpassen. Das daraus resultierende Transformations-Netzwerk ist ein vielversprechendes Modell für die Heterogenitätsbehandlung.

Die Diskussion dieser Modelle zeigt, dass deren Architekturen nicht gut für Information Retrieval geeignet sind. Die innovative Idee, im COSIMIR-Modell die Relevanz als eigenes Neuron einzuführen, führt zu den entscheidenen Vorteilen.

6.3 Mit COSIMIR vergleichbare Ansätze

Das COSIMIR-Modell wurde im Rahmen der Information Retrieval Forschung noch nicht vorgeschlagen. Es gibt aber Ansätze aus verschiedenen Bereichen, die mit der Funktionsweise oder dem Ergebnis von COSIMIR vergleichbar sind. Der bereits in Abschnitt 4.3.2.6 besprochene Ansatz von Wong et al. 1993 weist ebenfalls Ähnlichkeit mit dem COSIMIR-Modell auf. Die einzige Output-Unit berechnet einen Präferenz-Wert für eine Anfrage und einen Differenz-Vektor aus zwei Dokumenten. Diese Präferenz-Werte können als spezifische Form der Ähnlichkeit interpretiert werden. Jedoch verfügt das Netz von Wong et al. 1993 nicht über versteckte Schichten und realisiert somit keine sub-symbolischen Repräsentationen. Indem COSIMIR diese Fähigkeit des Backpropagation-Algorithmus ausnutzt, erhöht es seine Mächtigkeit und kann komplexere Funktionen implementieren, die nicht linear trennbar sind (cf. Abschnitt 3.5.4.1).

6.3.1 Retrieval von ähnlichen Prozessen

Am nähesten kommt dem COSIMIR-Modell der Ansatz von de Jong et al. 1996. Ziel der Autoren ist der Aufbau eines Case-Based-Reasoning Systems, das ähnliche industrielle Prozesse findet, so dass ein Ingenieur letztendlich energiesparendere Prozesse entwerfen kann. Die Prozesse werden durch Vektoren von Eigenschaften repräsentiert. Der Anwendungsfall ähnelt stark dem Problem von Escobedo et al. 1993, die ein System für die Suche nach ähnlichen Bauteilen implementiert haben (cf. Abschnitt 4.5).

De Jong et al. 1996 zeigen sich unzufrieden mit den üblichen Methoden zur Bestimmung von Ähnlichkeit im Case-Based-Reasoning, wo meist mathematische Ähnlichkeitsfunktionen und (fuzzy) Regelsysteme eingesetzt werden. Sie experimentieren mit verschiedenen Distanz-Funktionen wie der Euklidischen und der Hamming-Distanz. Diese Methoden erfüllen die speziellen Anforderungen des Anwendungsfalles nicht und erfassen nicht alle wichtigen Dimensionen der Ähnlichkeit.

De Jong et al. 1996 sammelten Expertenurteile zur Ähnlichkeit von Prozessen. Diese Daten nutzten sie als Lerndaten für ein Backpropagation-Netzwerk, dessen Architektur dem COSIMIR-Netzwerk gleicht. Zwei Prozesse, die miteinander verglichen werden, gehen als Input in das Modell, Output ist ein Ähnlichkeitswert. Die Qualität des Netzes schätzen due Autoren als gut ein. Angaben zur Zahl der Eigenschaften und Qualität der erreichten Abbildung fehlen. Die Zahl der Merkmale scheint jedoch niedrig zu ein. De Jong et al. 1996 liefern damit einen Hinweis für die Adäquatheit der Architektur des COSIMIR-Modells als Ähnlichkeitsfunktion in komplexen Anwendungsfällen. Die Anpassung an die Objekte im Information Retrieval, wobei eine Anfrage und ein Dokument als Input dienen, leisten die Autoren nicht. Ebenso lässt der Ansatz von de Jong et al. 1996 keine Aussagen zu über die Qualität des Modell im Information Retrieval und seine Eignung für umfangreiche Daten.

6.3.2 Gedächtnismodell

Barnden 1994 schlägt ein ähnliches Modell wie das COSIMIR-Modell in einem anderen Kontext vor, von einer Implementierung berichtet er jedoch nicht. Barnden 1994 beschäftigt sich mit der kognitionswissenschaftlichen Frage des Verhältnisses von Kurz- und Langzeitgedächtnis. Er glaubt, dass beim Retrieval aus dem Langzeitgedächtnis zunächst im Kurzzeitgedächtnis der Auslöser des Erinnerns geladen wird. Durch diese Annahme ergeben sich einige Probleme. Als plausiblen Ansatz stellt Barnden 1994 ein Modell vor, das dem COSIMIR-Modell ähnelt. Zwei Repräsentationen im Input werden durch das Netz verglichen und als Output ergibt sich ein Ähnlichkeitswert. Ein Teil des Input ist das Kurzzeitgedächtnis und ein weiterer Teil sind abwechselnd verschiedene Repräsentationen aus dem Langzeitgedächtnis.

6.3.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) wird ausführlich in Abschnitt 2.1.2.4.3 dargestellt. LSI modelliert nicht den Vergleich von Dokument und Anfrage im

Information Retrieval, sondern erstellt eine optimierte Repräsentation in Form einer reduzierten Matrix, in der die Dokumente nicht mehr von den Termen beschreiben werden, sondern durch 100 bis 300 LSI-Dimensionen. Diese neuen Eigenschaften, die LSI-Dimensionen oder Faktoren bilden nicht einzelne Terme oder Gruppen von Termen ab, sondern stehen für komplexe Kombinationen von Termen, die nachträglich nicht symbolisch interpretierbar sind. Damit führt LSI zu einem ähnlichen Ergebnis wie ein Backpropagation-Netzwerk. Daneben gibt es jedoch große Unterschiede. Während Backpropagation ein überwachtes Lernverfahren ist, steckt hinter LSI ein mathematisches Verfahren, das die Daten analysiert und keine Lernvorgabe benötigt. Zwar kann man beide Verfahren als Abbildung von einem vieldimensionalen Raum in einen zweiten vieldimensionalen Raum betrachten, jedoch ist LSI dabei wenig flexibel. Der zweite Raum ist eine reduzierte Version des ursprünglichen Merkmals-Raumes, während das Backpropagation-Netzwerk je nach Anwendungsfall Abbildungen zwischen heterogenen Räumen realisiert.

Latent Semantic Indexing stellt kein Retrievalsystem dar, sondern dient der Vorverarbeitung. Da COSIMIR den Retrievalprozess modelliert, ergänzen sich die beiden Ansätze. Die Ähnlichkeitsberechnung für die LSI-Repräsentationen kann von einem COSIMIR-Netz übernommen werden. Da große Merkmals-Räume für COSIMIR zu Schwierigkeiten führen, bietet Latent Semantic Indexing durch die Reduktion der Dimensionalität eine vielversprechende Ergänzung (cf. Abschnitt 6.4.2).

6.4 Erweiterungen des COSIMIR-Modells

Das COSIMIR-Modell erweist sich als vielfältig modifizierbar und damit als sehr flexibel. Neben verschiedenen Methoden zur Erzeugung von Mustern (cf. Abschnitt 6.1.4) kann v.a. die Architektur des Basismodells von COSIMIR an verschiedene Erfordernisse angepasst werden.

6.4.1 Modifikation der Verbindungsmatrix

Im Rahmen des COSIMIR-Modells sind verschiedenste Verknüpfungsmuster denkbar. Neben der üblichen vollständigen Verknüpfung zwischen allen Schichten ist es möglich, zunächst die Neuronen für identische Terme in einer Art Differenz-Schicht zu kombinieren. Diese Differenz-Schicht stellt im Backpropagation-Modell technisch die erste versteckte Schicht dar, die genau halb so groß ist wie die Eingangs-Schicht. Die Dokument- und die Anfrage-Repräsentation werden so verarbeitet, dass zunächst die Neuronen der Diffe-

renz-Schicht die Aktivierung zu einem Term aus Anfrage und Dokument kombinieren wie Abbildung 6-6 zeigt. Dies entspricht der Funktion der meisten mathematischen Ähnlichkeitsformeln, in denen oft die Gewichte für Terme mit gleichem Index aus Anfrage und Dokument multipliziert werden (cf. Abschnitt 2.1.3). Aus der Forschung zu neuronalen Netzen ist bekannt, dass Backpropagation-Netzwerke durch inhaltlich geleitete Reduktion von Verbindungen teilweise verbessert werden (cf. le Cun 1989, cf. Abschnitt 3.5.4.2). Die hier vorgestellte Modifikation in Form der Differenz-Schicht stellt eine Sonderform der Reduktion dar.

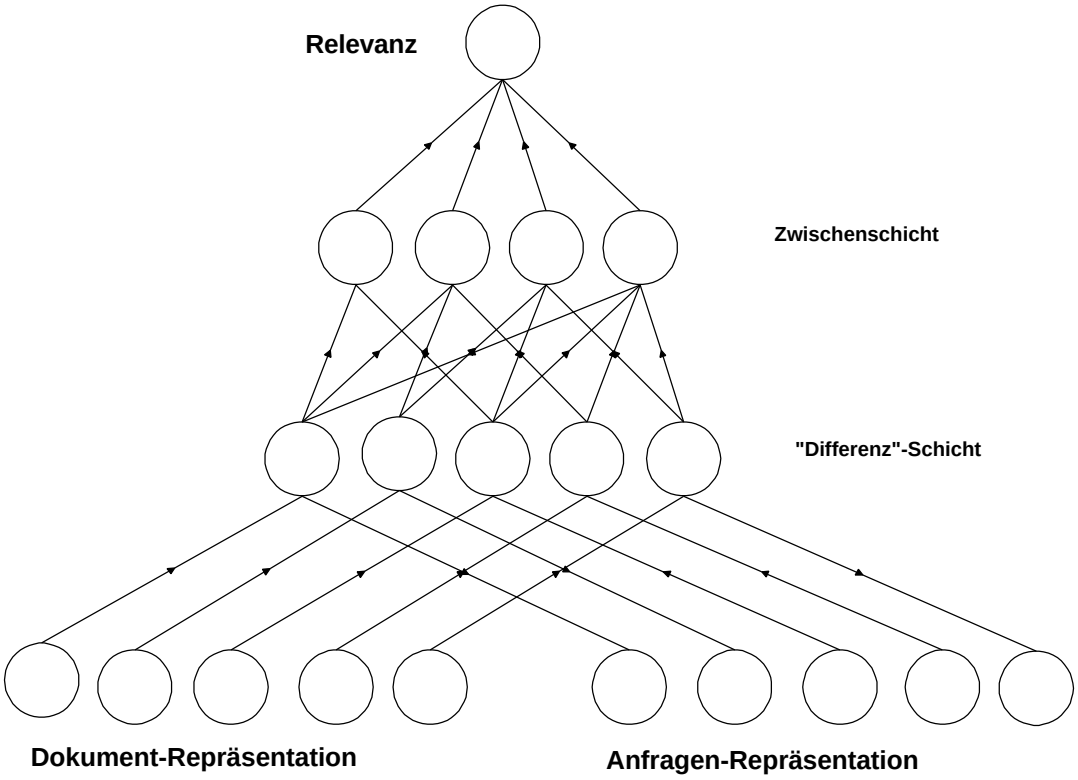


Abbildung 6-6**:** COSIMIR-Netz mit „Differenz“-Schicht

Die Anzahl der Verbindungen im COSIMIR-Modell und im COSIMIR-Modell mit Differenz-Schicht ergeben sich bei anonsten vollständiger Verknüpfung zwischen den Schichten nach folgenden beiden Formeln:

COSIMIR:	$v = (2\ t\ h) + h$
COSIMIR mit Differenz-Schicht:	$v = 2\ t + h\ (t+1)$

t Anzahl der Terme
h Anzahl der Neuronen in der versteckten Schicht vor dem Output

Wie die Beispiele in Tabelle 6-1 zeigen, reduziert die Einführung der Differenz-Schicht die Anzahl der Verbindungen erheblich.

Tabelle 6-1: Vergleich der Anzahl der Verbindungen

Anzahl der Terme (t)	**Anzahl der Neuronen in der in der versteckten Schicht vor dem Output (h)**	**Anzahl der Verbindungen im COSIMIR-Modell**	**Anzahl der Verbindungen im COSIMIR-Modell mit Differenz-Schicht**
100	10	2010	310
1000	20	40.020	22.020

Allerdings kann die Architektur mit der Differenz-Schicht auch dazu führen, dass die Zusammenhänge zwischen den verschienden Termen schlechter gelernt werden können. Wird COSIMIR mit einer Komprimierung des Merkmals-Raumes kombiniert, dann repräsentiert ein Neuron eine Kombination aus mehreren Termen. In diesen Fällen ist fraglich, ob der Einsatz der Differenz-Schicht zu einer Verbesserung der Abbildung führt.

6.4.2 Komprimierung von Repräsentationsvektoren

Die Grösse der Merkmals-Räume stellt ein Problem für alle Information Retrieval Verfahren mit neuronalen Netzen dar: „How do we overcome problems of scale limitations of connectionist implementations in real-life, large-scale operational systems?“ (Doszkocs et al. 1990:243).

Im COSIMIR-Modell erfordert der Informationsfluss eine starke Reduktion. Beim Retrieval wird sehr viel Information aus Neuronen für jeden Term im Input über versteckte Schichten zu nur einem Neuron im Output verdichtet. Beim Rücklauf der Fehlerinformation in der anderen Richtung kehrt sich das Verhältnis um. Vom Fehler eines Neurons lernen alle Verbindungen. Zum anderen bestehen die Dokument- und Anfrage-Vektoren bei den üblichen Indexierungsmethoden hauptsächlich aus Nullen. Diese spärlich besetzen Vektoren bilden auch ein Problem für andere Verfahren der Ähnlichkeitsberechnung.

Datenkomprimierung ist geeignet, dieses Missverhältnis und das Größenproblem zu entschärfen. Im Information Retrieval wurden bereits einige Verfahren erprobt (cf. Abschnitt 2.1.2.4). Besonders Latent Semantic Inde-

xing (LSI) scheint von den oben vorgstellten Systemen für COSIMIR und das Transformations-Netzwerk geeignet zu sein. Bei der in Abbildung 6-7 skizzierten Komprimierung mit Backpropagation sind Input und Output identisch (cf. auch Abschnitt 2.1.2.4.2). Bei spärlich besetzten Matrizen scheint diese Komprimierung schwer erlernbar, da das Netzwerk als erwünschten Output sehr häufig Null reproduzieren soll und dabei leicht übergeneralisiert und immer Null liefert. Die Komprimierung durch den Kontext-Vektor erfordert hohen intellektuellen Aufwand, der für ein generelles IR System vermieden werden soll (cf. Abschnitt 2.1.2.4.1).

Latent Semantic Indexing wurde bereits für die Komprimierung des Inputs eines neuronalen Information Retrieval Systems eingesetzt. Der Spreading-Activation-Ansatz von Syu et al. 1996 wird näher in Abschnitt 4.3.2.5 beschrieben.

LSI-Repräsentationen im Input sorgen für eine bessere Verteiltheit. Wie Kapitel 4 zeigt, repräsentiert in den meisten Modellen ein Neuron einen Term, so dass eine lokalisierte Repräsentation vorliegt. Legt man eine LSI-Repräsentation an ein Netz an, so verteilt sich die Infomation zu einem Term auf mehrere Neuronen.

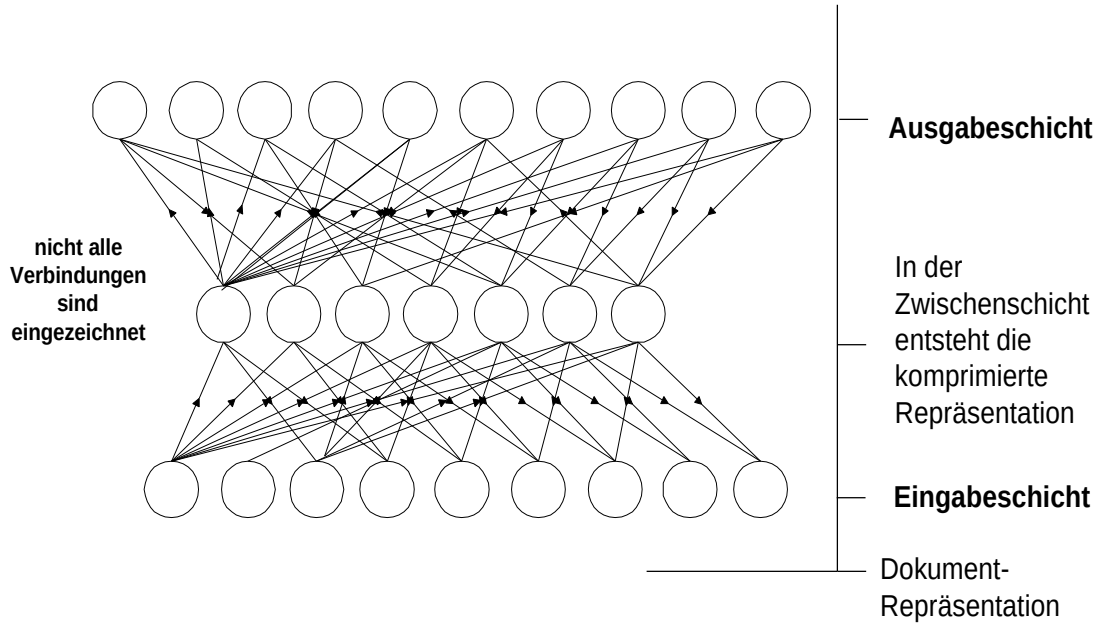


Abbildung 6-7: Komprimierung von Information Retrieval Daten mit einem Backpropagation-Netzwerk

In der Forschung zu neuronalen Netzen wird Singular Value Decomposition (SVD) unabhängig vom Anwendungsfall Information Retrieval als Vorverar-

beitungsmechanismus für spärlich besetzte Matrizen empfohlen. SVD ist der mathematische Kern von LSI. Kanjilal/Banerjee 1995 setzen SVD ein, um die Eingangsdaten zu komprimieren, die Größe der versteckten Schicht zu optimieren und die Konvergenz zu prüfen. Zur Komprimierung der Eingangsdaten extrahieren sie die Singular Values der gesamten Matrix und bilden dann die reduzierte Matrix. Die Frage wieviele der Singular Values und dementsprechend wieviele Dimensionen berücksichtigt werden, lösen sie heuristisch. Als Anhaltspunkt dazu dient der Betrag der Singular Values, die monoton fallen. Je größer eine Singular Value, desto wichtiger ist die entsprechende Dimension. Umgekehrt gilt, je kleiner eine Singular Value, desto eher kann sie vernachlässigt werden.

Zur Optimierung der Anzahl der versteckten Neuronen, trainieren Kanjilal/Banerjee 1995 zunächst ein zu großes und überparametrisiertes Netz. Der Output der ersten versteckten Schicht bildet für alle Muster eine Matrix. Auf diese Matrix wird wieder SVD angewandt, um festzustellen, welche der Dimensionen wichtig sind. Die Anzahl der versteckten Neuronen wird dann auf diese Anzahl reduziert.

Kanjilal/Banerjee 1995 schlagen vor, neben den üblichen Fehlermaßen bei neuronalen Netzen wie dem durchschittlichen quadrierten Fehler, die Konvergenz mit zusätzliche Maßen zu ermitteln. Auch dafür setzen sie Singular Value Decomposition ein. In jeder Epoche wird die Änderung an den Gewichten als Vektor gespeichert. SVD zerlegt die entstehende Matrix über Epochen und Gewichtsveränderungen. Die Singular Values bilden die Basis eines Maßes, das die Konvergenz als Energiefunktion beschreibt.

Die Komprimierung der Eingangsdaten ist für COSIMIR der wichtigste Anwendungsfall von LSI. Komprimierung allgemein und mit LSI ist eine erfolgreich angewandte Technik im Information Retrieval. Das größte Problem von COSIMIR ist die Größe der entstehenden Matrizen. Die Kombination von COSIMIR und Komprimierung liegt also auf der Hand.

6.4.3 Komplexes COSIMIR-Modell mit Kontext-Informationen

Eine interessante Möglichkeit der Erweiterung von COSIMIR besteht darin, als Input neben Dokument und Anfrage auch Kontext- und Benutzerinformationen zu benutzen. Dies ist in Domänen wichtig, in denen die Relevanz nicht feststeht, sondern abhängig von Benutzer und Situation sehr unterschiedlich interpretiert ausfallen kann. Liegen genug Trainings-Beispiele dafür vor, wie die Relevanz abhängig von Benutzer oder Kontext unterschiedlich bewertet wird, kann ein solches Netz trainiert werden. In Abbildung 6-8 ist diese Mög-

lichkeit durch die Einführung von zusätzlichen Input-Neuronen für Kontext- oder Benutzer-Informationen mit berücksichtigt.

Das COSIMIR-Modell kann auch zu einem aus mehreren Netzwerken zusammengesetzten komplexen Modell erweitert werden. Dabei dient das Transformations-Netzwerk der Vorverarbeitung und bildet heterogene Repräsentationen der Anfrage und der Dokumente auf eine gemeinsame Basis ab. Wie Abbildung 6-8 zeigt, ist dies sowohl bei der Anfrage als auch beim Dokument denkbar. Dabei würden beide aus verschiedenen Repräsentationsverfahren in ein drittes transformiert und darin verglichen. Ebenso kann das Verfahren auf die Anfrage oder das Dokument beschränkt werden. Diese Erweiterung lässt sich auch unabhängig von der Integration der Benutzer- oder Kontext-Informationen implementieren. Die mögliche Komprimierung ist dabei nicht berücksichtigt.

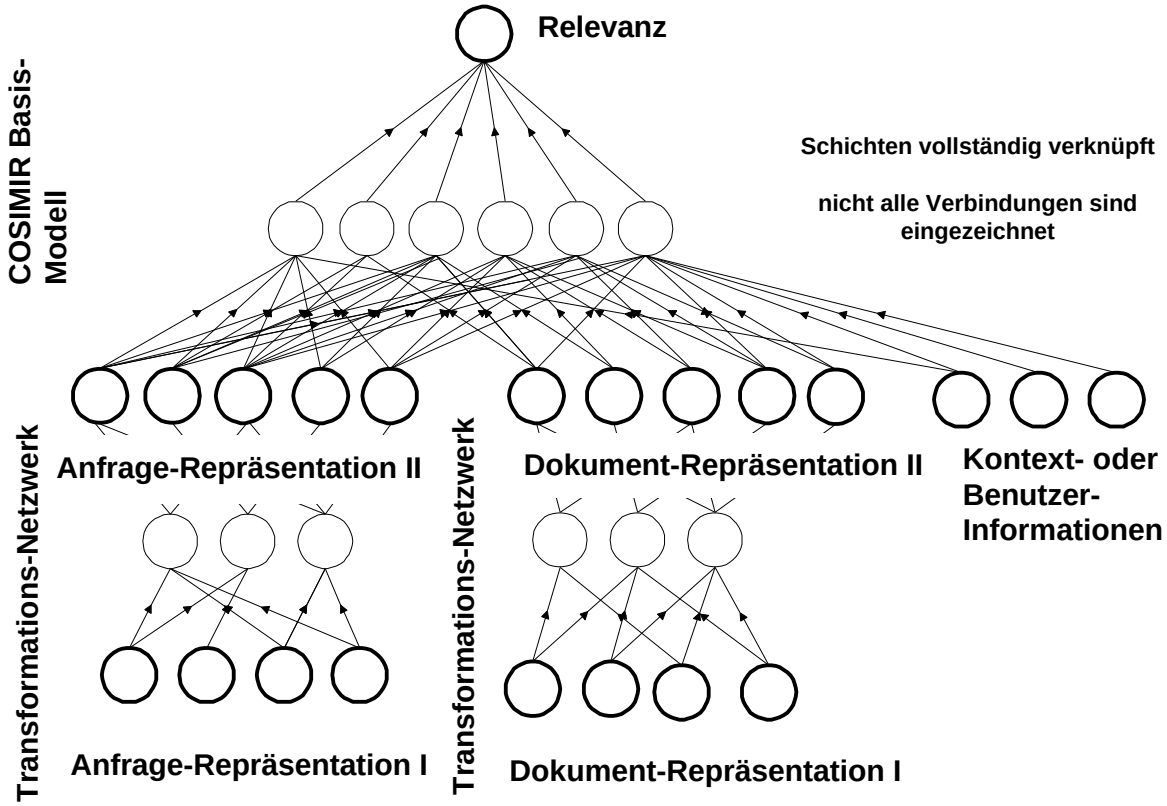


Abbildung 6-8: Erweitertes COSIMIR-Modell

Damit können heterogene Anwendungsbereiche bearbeitet werden. Es ist denkbar, dass multilinguale Anfragen und Dokumente in eine gemeinsame Repräsentations-Sprache transformiert werden und dann in den Retrieval-Prozess eingehen. Neben multilingualen können auch multimediale Doku-

mente-Repräsentationen, die z.B. aus Fakten oder Bildern auf eine einheitliche Dokument-Repräsentation abgebildet werden. Die einzelnen Netzwerke des komplexen Systems für Heterogenitätsbehandlung werden nach wie vor einzeln trainiert.

6.4.4 COSIMIR für heterogene Repräsentationen

Das COSIMIR-Modell kann, wie im letzten Abschnitt gezeigt, in einem heterogenen Kontext mit dem Transformations-Netzwerk kombiniert werden. Daneben lässt sich COSIMIR auch mit jedem anderen Transformations-Verfahren kombinieren. Wie im letzten Abschnitt gezeigt, wird der Transformation entweder das Dokument oder die Anfrage vor dem Vergleich im COSIMIR-Netzwerk in einen anderen Term-Raum abgebildet. Der Output eines Transformations-Netzwerks dient als Input eines COSIMIR-Modells.

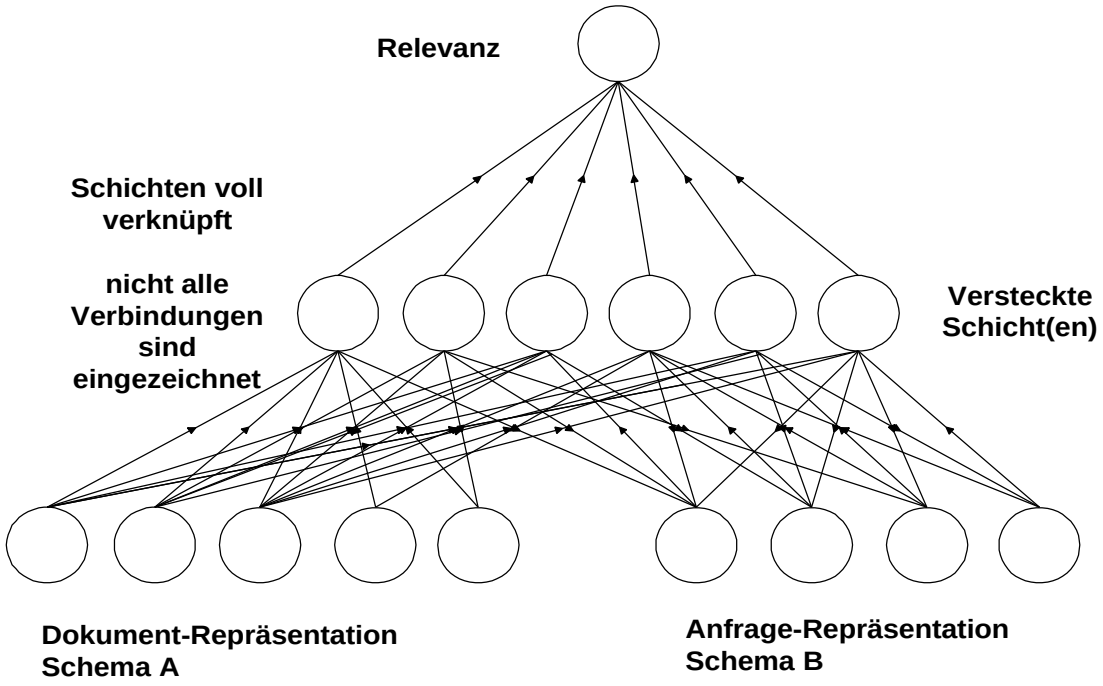


Abbildung 6-9: COSIMIR-Modell für Heterogenitätsbehandlung

COSIMIR ermöglicht aber noch eine weit flexiblere Reaktion auf heterogene Begriffsschemata. Wie schon in Abschnitt 6.1 erläutert, macht COSIMIR keine Annahmen über die mathematische Form der kognitiven Ähnlichkeitsfunktion, die es implementiert. Auch über die Input-Daten macht COSIMIR keine Annahmen und so müssen beide Seiten des Inputs nicht homogen sein

und aus identischen Term-Räumen stammen. Eine gleichförmige Repräsentation von Dokument und Anfrage ist nicht erforderlich, um mit dem COSIMIR-Modell die Ähnlichkeit zu bestimmen. Voraussetzung sind lediglich genügend Trainingsdaten in Form von Benutzerurteilen über heterogen repräsentierte Objekte. In diesem Fall greift der Input direkt auf die heterogenen Vektoren zu und berechnet daraus in einem Netzwerk die Ähnlichkeit wie Abbildung 6-9 zeigt.

Der explizite Transformationsschritt entfällt in diesem Modell. Statt dessen lernt COSIMIR anhand der Benutzerurteile direkt, die Ähnlichkeit aus den Gewichten der Terme der unterschiedlichen Indexierungverfahren abzuleiten.

6.5 Fazit: COSIMIR-Modell

Das COSIMIR-Modell ist ein neuartiges und einfaches Verfahren, das den Vergleich zwischen Dokument und Anfrage im Information Retrieval vollständig auf der Basis von Benutzerurteilen lernt. Dadurch wird im einem IR-Modell die heuristische Wahl einer mathematischen Ähnlichkeitsfunktion vermieden. Statt dessen implementiert das COSIMIR-Modell die Ähnlichkeitsfunktion auf der Basis der Benutzerurteile und realisiert so eine kognitiv angemessenere Ähnlichkeitsberechnung. COSIMIR eignet sich auch für die Ähnlichkeitsberechnungen in anderen Kontexten wie etwa Case Based Reasoning oder Data Mining.

Mehrere voneinander unabhängige Argumentationslinien führen zum COSIMIR-Modell:

- Die menschliche Ähnlichkeitsbeurteilung kann unsymmetrisch und nicht transitiv sein. Diese Eigenschaften decken die meisten mathematischen Ähnlichkeitsfunktionen wie etwa die verbreitete Kosinus-Funktion nicht ab (cf. Abschnitt 2.1.3).
- Die bestehenden Information Retrieval Modelle auf der Basis neuronaler Netze besitzen erhebliche Schwächen wie etwa mangelnde Lernfähigkeit. Sie nutzen auch die Mächtigkeit neuronaler Netze zur Realisierung subsymbolischer Repräsentationen nicht aus (cf. Abschnitt 4.9).
- Relevanz-Feedback führt häufig zu erheblichen Verbesserungen der Qualität im Information Retrieval, wie u.a. die Experimente im Rahmen von TREC gezeigt haben (cf. Abschnitt 2.1.4.2). Ein Information Retrieval System sollte die Urteile von Benutzern miteinbeziehen, dieses Wissen aber nicht mit den ursprünglichen Repräsentationen vermischen (cf. Abschnitt 6.1.2).

- Mehrere mögliche Information Retrieval Modelle auf der Basis des Backpropagation Netzwerks wie das Anfrage-Dokument-Profil-Modell und das Anfrage-Dokumenten-Vektor-Modell können den Information Retrieval Prozess nicht optimal abbilden (cf. Abschnitt 6.2.4).

Das COSIMIR-Modell reagiert auf diese Schwierigkeiten:

- Kognitiv adäquate Ähnlichkeitsfunktion:
  COSIMIR implementiert eine Ähnlichkeitsfunktion aufgrund von Trainingsbeispielen und macht dazu keine Annahmen über formale Eigenschaften der zu lernenden Funktion. COSIMIR kann nicht symmetrische und nicht transitive Funktionen realisieren, wenn die Trainingsdaten dies erfordern.
- Hohe Lernfähigkeit:
  COSIMIR nutzt den mächtigen und häufig eingesetzten Backpropagation-Algorithmus, der in den versteckten Schichten sub-symbolische Repräsentationen erstellt.
- Lernen anhand von Benutzerurteilen:
  Die Funktion zur Ähnlichkeitsberechnung wird vollständig und ausschließlich auf der Basis von Benutzerurteilen erlernt.
- Optimale Architektur:
  Durch die Einführung des Output-Neurons für die Relevanz entsteht das günstigste Backpropagation-Modell für Information Retrieval Prozesse.

Die Realisierung von COSIMIR führt zu einem Information Retrieval System das keine unrealistischen Annahmen zur Unabhängigkeit von Termen erfordert. Ein weiterer entscheidender Vorteil liegt in der Flexibilität und Erweiterbarkeit. Besonders die integrative Adaption zur Behandlung heterogener Daten zeigt die hohe Flexibilität des Ansatzes.

Neben dem COSIMIR-Modell ist das ausschließlich für die Heterogenitätsbehandlung einsetzbare Transformations-Netzwerk ein zweites vielversprechendes Backpropagation-Netzwerk im Bereich Information Retrieval.

7 Experimente mit dem COSIMIR-Modell und dem Transformations-Netzwerk

Die im vorhergehenden Kapitel diskutierten Netzwerke für Information Retrieval sind theoretisch gut fundiert. Aber erst die Evaluierung zeigt, ob diese Modelle für reale Daten geeignet sind und inwieweit sie die Qualität anderer Verfahren erreichen oder diese sogar übertreffen. Dieses Kapitel stellt Experimente mit dem COSIMIR-Modell, dem COSIMIR-Modell für Heterogenitätsbehandlung und dem Transformations-Netzwerk vor. Die Tests des Transformations-Netzwerk mit realen Kollektionen sind erforderlich, da die bisherigen Experimente nur auf sehr kleinen Datenmengen basieren (cf. Abschnitt 5.3.4.1). Die Qualität der COSIMIR-Modelle wird mit den für Information Retrieval Systeme üblichen Verfahren und Maßzahlen für den Erfolg des Retrievals gemessen (cf. Abschnitt 2.1.4.1). Beim Transformations-Netzwerk wird die Qualität der Transformation isoliert bewertet. Abschnitt 7.2 zeigt, dass dies auch bei anderen Verfahren zur Heterogenitätsbehandlung üblich ist und einen Vergleich mehrerer Algorithmen erlaubt. In einem IR-Prozess stellt die Transformation allerdings nur einen Aspekt dar und die Messung deren Qualität lässt keine Rückschlüsse auf die Qualität des gesamten Information Retrieval Prozesses zu.

Für das COSIMIR-Modell für Heterogenitätsbehandlung steht dagegen keine etablierte Evaluierungsmethode zur Verfügung. Abschnitt 7.4.3 präsentiert ein Verfahren, das eine erste Bewertung des Modells ohne den aufwendigen intellektuellen Aufbau einer Kollektion mit Expertenurteilen zu heterogenen Repräsentationsschemata oder heterogenen Objekten ermöglicht.

Ein Problem besteht in den unterschiedlichen Anforderungen an die Datenbasis. Während COSIMIR für die Evaluierung die gleichen Daten erfordert, wie jedes andere IR-Modell, ist die Adaption von COSIMIR für heterogene Datenbestände auf umfangreiche Relevanzurteile zu heterogenen Objekten angewiesen. Zur Zeit ist kein entsprechend aufbereiteter Datenbestand für die Heterogenitätsbehandlung bekannt. Zudem eignet sich auch nicht jeder heterogene Datenbestand für das entsprechende COSIMIR-Modell. Anders als für das Transformations-Netzwerk oder ein statistisches Transformations-Verfahren sind statt dem Doppelkorpus mit zweifach erschlossenen Objekten Relevanzbewertungen für die heterogenen Objekten nötig. Aus diesen Gründen ist es schwierig, einen realen Datenbestand für die Evaluierung aller Fragestellungen zu finden. Deshalb erfolgen die Experimente für die Modelle mit jeweils unterschiedliche Daten und somit sind die Ergebnisse untereinander schwer vergleichbar.

COSIMIR wird zunächst mit einer oft verwendeten, kleinen Textkollektion, der Cranfield II Kollektion evaluiert (cf. Abschnitt 7.1).

Das Transformations-Netzwerk wird anhand eines mittlerweile in TREC eingegangenen Datenbestandes getestet, den das IZ-Sozialwissenschaften intellektuell mit zwei unterschiedlichen Thesauri verschlagwortet hat (cf. Abschnitt 7.2).

Beide Datenbestände eignen sich nicht für das COSIMIR-Modell für Heterogenitätsbehandlung, in dem der Transformationsschritt nicht mehr explizit sichtbar ist. Auf Basis eines Bestandes von realen Faktendaten wird eine Evaluierungsmethode entwickelt, wobei das Standard-COSIMIR-Modell als Vergleichsmaßstab erneut bewertet wird. Dabei ergeben sich sehr ermutigende Resultate, die Abschnitt 7.3 detailliert vorstellt. Demnach eignet sich COSIMIR mit hoher Sicherheit für Fakten-Retrieval.

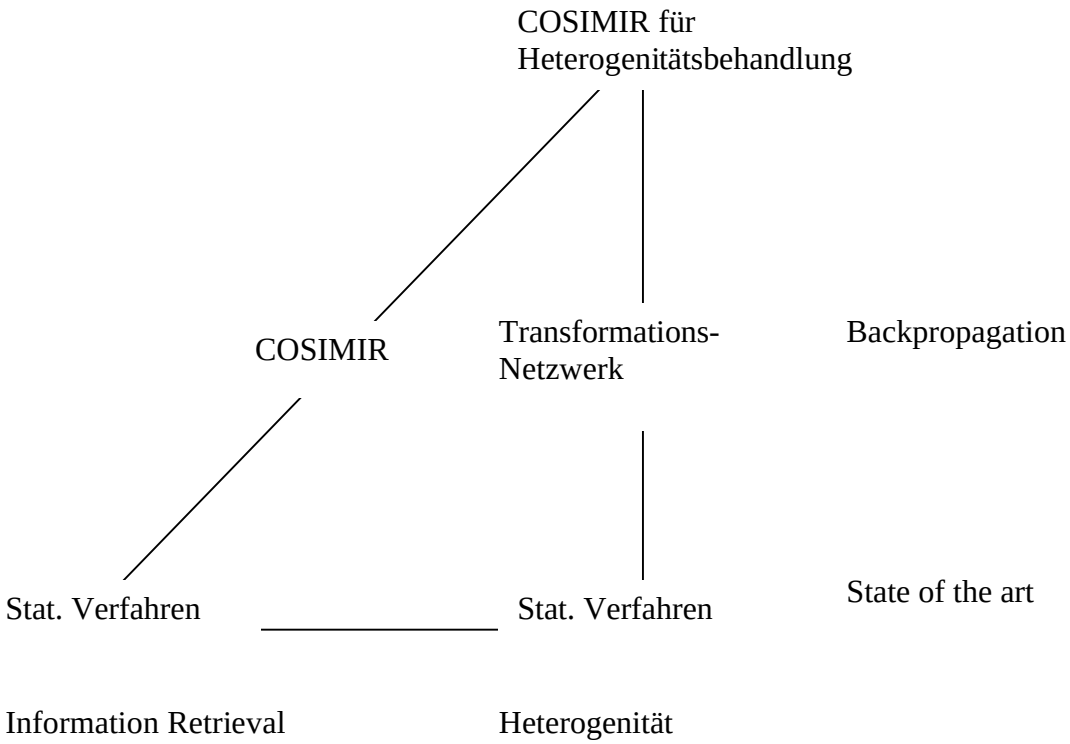


Abbildung 7-1: Status der Evaluierung von COSIMIR

Abbildung 7-1 gibt einen Überblick über die Fragestellungen und die jeweiligen Experimente ausgehend von den bestehenden IR-Verfahren. Die Komplexität der Experimente und die Anforderungen an die Methoden und die Datenbasis steigen ausgehend von den Standard-Verfahren unten nach oben hin an.

Folgende Bereiche werden also untersucht:

- COSIMIR
  - Kann das COSIMIR-Modell Ähnlichkeitsberechnungen lernen?
  - Eignet sich das COSIMIR-Modell für Ähnlichkeitsberechnungen im Information Retrieval und erreicht es eine vergleichbare Qualität wie andere Ansätze?
- Heterogenitätsbehandlung
  - Eignet sich das Transformations-Netzwerk für Transformationen und erreicht es eine mit statistischen Ansätzen vergleichbare Qualität?
- COSIMIR + Heterogenitätsbehandlung
  - Eignet sich das adaptierte COSIMIR-Modell für Heterogenitätsbehandlung ohne explizite Transformation?

Abschnitt 7.5 diskutiert die erzielten Ergebnisse.

7.1 Experimente mit der Cranfield-Text-Kollektion

Das COSIMIR-Modell wurde zunächst mit Textdaten evaluiert. Eine besondere Schwierigkeit stellen dabei die langen Vektoren dar, die zu sehr großen Netzen führen. So wurde zunächst eine kleine, häufig verwendete Testkollektion benutzt.

7.1.1 Cranfield Kollektion

Einen Überblick über die Cranfield Kollektion und die ursprünglich damit unternommenen Tests bietet Sparck Jones 1981.

Die Kollektion steht im Internet zur Verfügung[1]. Sie umfasst 1400 Dokumente und 225 Anfragen. Die Daten enthalten bereits die Vorkommenshäufigkeit der Terme in den Dokumenten und Anfragen und Relevanzurteile in Form von Listen von Dokumenten zu jeder Anfrage. Insgesamt kommen in den Dokumenten 3763 Deskriptoren vor, jedoch nur 585 kommen auch in den Anfragen vor. Für jedes Dokument sind durchschnittlich 84 Terme vergeben. Jeder Term ist durchschnittlich für 31 Dokumente vergeben.

[1] ftp://ftp.cs.cornell.edu/pub/smart/cran/

Der Nachteil der Cranfield Anfragen besteht darin, dass sie aufgrund der Dokumente formuliert wurden. Die Dokumente waren also der Ausgangspunkt für die Formulierung von Anfragen, um diese Dokumente zu finden. Es handelt sich also um eine künstlich geschaffene Beziehung und nicht um echte Benutzeranfragen. Die Beziehung zwischen Anfragen und Dokumenten ist dadurch sehr eng (cf. Sparck Jones 1981).

Der Hauptgrund für die Auswahl der Cranfield Kollektion liegt in der relativ großen Zahl von Anfragen im Verhältnis zu Dokumenten. Da bei COSIMIR Anfrage und Dokument parallel in das Netz eingehen, ist eine höhere Anzahl von Anfragen wünschenswert, um dem Netz auf der einen Hälfte des Input verschiedene Vektoren bieten zu können. Für die meisten Experimente wurden nur die 585 Deskriptoren ausgewählt, die in den Anfragen vorkommen. Dadurch werden die Netze kleiner und sind schneller trainierbar.

7.1.2 Cranfield Experimente mit COSIMIR

Als Vergleichsmaßstab dient ein Vektorraum-Modell mit inverser Dokument-Frequenz und dem Kosinus als Ähnlichkeitsmaß. Diese Baseline liefert relativ hohe Precision-Werte, die durch die oben erläuterte Künstlichkeit der Kollektion erklärbar sind.

In der ersten Phase erfolgten die Experimente mit der Simulationssoftware SNNS (Stuttgarter Neuronaler Netzwerk Simulator, cf. Abschnitt 3.6.1) unter SCO-UNIX auf einem Pentium-PC. Das Netz besteht aus einer Input-Schicht mit 1170 Units, je 585 für Dokument und Anfrage. Die erste versteckte Schicht umfasst 50 und die zweite zehn Neuronen. Die Output-Schicht enthält nur ein Neuron, das die Relevanz repräsentiert. Dies ergibt 292.510 Verbindungen. Aufgrund der langen Trainingszeiten konnte kaum experimentiert werden. In einer Woche waren nur wenige Epochen Training abgelaufen.

Für das Training wurden 150 Anfragen benutzt, die restlichen 75 für die Tests. Die 1400 Dokumente wurden für beides benutzt. Für erste Experimente wurden etwa 4000 Pattern gewählt. An das trainierte Netz wurden dann alle 75 Testanfragen mit je allen Dokumente als ein Input angelegt. So liefert das COSIMIR-Netz für alle Dokumente einen Ähnlichkeitswert zur Anfrage.

Daraus ergibt sich ein Ranking, mit dem die Precision bei Recall-Niveaus von 0,1 bis 0,9 berechnet wird. Die Ergebnisse sind weitaus schlechter als die des Vergleichssystems. Die Precision des Vergleichssystems ist jeweils mindestens zehnmal höher als in dem Experiment. Die Kurve fällt auch nicht ausgehend von einem hohen Precision-Level mit steigendem Recall, sondern verläuft nahe der X-Achse.

Wie lässt sich dieses schlechte Ergebnis erklären? Vermutlich stehen nicht genügend Lerndaten zur Verfügung. Laut einer Faustregel von Bigus 1996 sollten pro Verbindung zwei Beispiele zur Verfügung stehen. Bei 292.510 Verbindungen wären dies etwa 600.000 Beispiele statt den 4000.

Um dieses Verhältnis zu verbessern, wurde eine Trainingsmenge mit allen zur Verfügung stehenden Dokumenten erstellt. Damit kann die Generalisierung nicht mehr getetet werden, es wird dient lediglich geprüft, ob das Netz zumindest konvergiert und das Lernverfahren den Fehler minimiert. Bei 150 Anfragen für das Training und 1400 Dokumenten ergeben sich 210.000 Beispiele, so dass zumindest die Größenordnung der Verbindungsanzahl erreicht wird. Diese Trainingsmenge beanspruchte 0,5 Gigabyte Speicherplatz. Nach drei Wochen Trainingszeit war das Netz noch nicht konvergiert, so dass ein anderer experimenteller Aufbau gesucht werden musste. Auswege bieten die Verwendung leistungsfähigerer Hardware und die Komprimierung von Dokumentdaten.

7.1.3 Cranfield Experimente mit COSIMIR und LSI

Um ein aussagefähiges Experiment durchzuführen, erfolgte eine Kompression mit LSI (Latent Semantic Indexing, cf. Abschnitt 2.1.2.4.3). Die 585 originalen Dimensionen der Kollektion wurden per LSI auf zwischen 20 und 100 reduziert. Diese Reduktion leistet eine Testsoftware von Bellcore auf UNIX-Workstations.

Bei einer Reduktion auf 30 Dimensionen ergeben sich für das COSIMIR-Netz 60 Eingangs-Neuronen und bei einer versteckten Schicht mit fünfzehn Neuronen ca. 900 Verbindungen (cf. Abbildung 7-2). Nach der Faustregel von Bigus 1996 reichen dafür bereits ca. 1800 Trainingsbeispiele aus. Diese Experimente wurden mit der Simulations-Software DataEngine unter Windows NT und Windows 98 auf einem PC Pentium Processor durchgeführt (cf. Abschnitt 3.6.2).

Neuronale Netze werden in zahlreichen Trainingsläufe mit verschiedenen Parametern optimiert. Besonders die Anzahl der versteckten Neuronen wird heuristisch variiert. In den Experimenten mit COSIMIR wurden meist 50% und weniger der Neuronen der Eingangs-Schicht für die versteckte Schicht gewählt. Weiterhin sind die Transferfunktion, die Lernrate, Gewichts-Decay und verschiedene Backpropagation-Varianten und ihre Parameter wie Momentum wichtige heuristische Einstellmöglichkeiten. Die Bedeutung dieser Parameter erläutert Abschnitt 3.5.4. Als zusätzlicher Parameter kommt in diesem speziellen Fall die Anzahl der verwendeten LSI-Dimensionen dazu.

Als Entscheidungskriterium, ob eine Parameter-Änderung eine Verbesserung erbracht hat, dienen die Fehlermaße, wie Summe der Fehlerquadrate oder maximaler Fehler. Diese werden von DataEngine für die Trainings- als auch für die Testmenge berechnet. Sie geben an, wie gut das Netz die Funktion bereits gelernt hat. Für COSIMIR sind sie allerdings nicht sehr aussagekräftig. Ziel ist nicht die exakte Wiedergabe des Ähnlichkeitswerts, sondern Verbesserung des Retrievals. Diese Qualität zeigt sich erst bei der Berechnung der Recall-Precision Werte. Dazu muss eine umfangreiche Recall-Datei erstellt werden, die für alle Anfragen der Testmenge ein Muster mit der Anfrage und jedem Dokument enthält. Bei 50 Testanfragen sind dies 70.000 Muster.

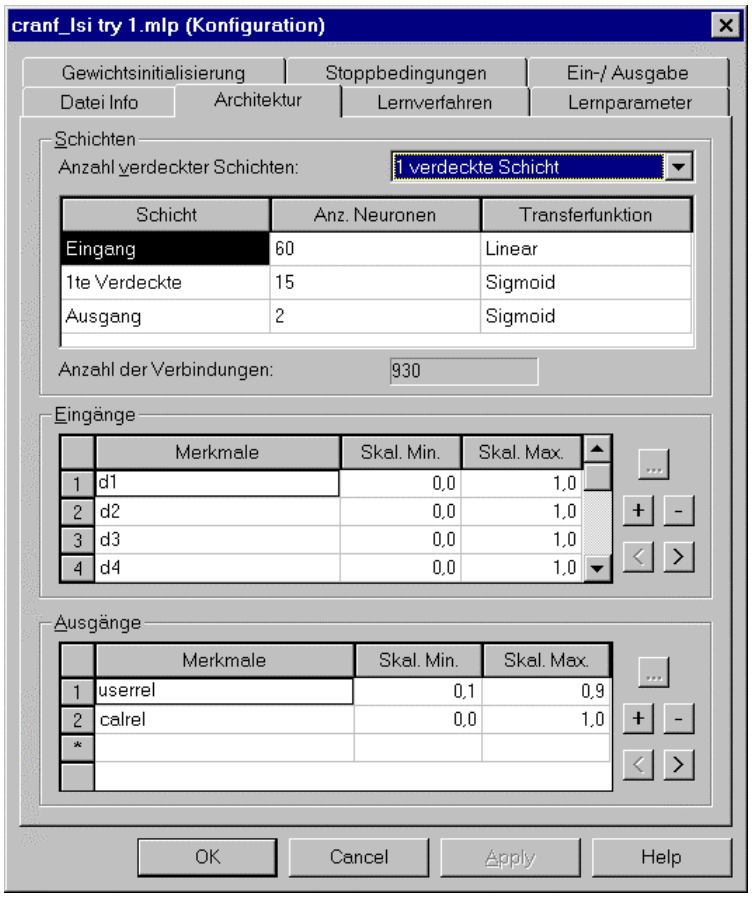


Abbildung 7-2: Architektur eines COSIMIR-Modells in der Software DataEngine

Für Anfragen in der Trainings- und Testmenge werden dagegen nicht alle Paare mit allen Dokumenten gebildet. Da die meisten Paare die Relevanz 0 aufweisen, würde das Netz trainiert, fast immer Null als Ergebnis zu liefern. Außerdem würde die Datei mit der Trainingsmenge wiederum sehr groß und bei dem Datenformat von DataEngine würden 180 LSI-Dimensionen und 200 Dokumenten in der Trainingsmenge bereits 0,5 Gigabyte umfassen.

Somit wurden die Trainingsanfragen nicht mit allen Dokumenten kombiniert. Um möglichst ausgewogene Ähnlichkeitswerte zu erhalten wurden zu jeder Anfrage alle relevanten Dokumenten und dann zufallsgesteuert noch eine feste Anzahl nicht relevanter Dokumente ausgewählt. Diese Zahl wurde bei den Experimenten variiert und lag zwischen 20 und 100. Auf diese Weise wurde auch der Input der Testmenge erstellt. Beim Output, also den zu errechnenden Ähnlichkeitswerten ergeben sich ebenfalls mehrere Möglichkeiten. Es ist zunächst naheliegend, die Relevanz aus Benutzer- oder Juroren-Sicht als Ähnlichkeit zu benutzen. Dadurch entstehen niedrige Durchschnittswerte für alle ins Netz eingehenden Relevanzwerte.

Um dies in den Trainingsdaten zu verbessern, kann bei den nicht relevanten Dokumenten eine berechnete Ähnlichkeit eingesetzt werden. Wird z.B. der Kosinus benutzt, liegen die Werte zwischen Null und Eins. Die vom Benutzer vorgegebene Relevanz ist also nach wie vor das Maximum. Beide Ansätze lassen sich zu einem Multi-Task-Netz (cf. Abschnitt 3.5.4.3) kombinieren. Dabei werden neben dem eigentlichen Output weitere Größen mit eigenen Neuronen hinzugefügt. Diese zusätzlichen Werte sollten einen inhaltlichen Zusammenhang mit dem gewünschten Output besitzen und helfen so dem Backpropagation-Netzwerk eine bessere interne Repräsentation aufzubauen. Alle diese Optionen wurden getestet, also sowohl Single-Task-Netze mit Juroren-Relevanz, Single-Task-Netze mit Juroren- und berechneter Relevanz als auch Multi-Task-Netze mit beidem.

In der Regel wird das Training beendet, sobald der Fehler in der Testmenge ein Minimum erreicht und durch Übergeneralisierung wieder zu steigen beginnt. Wie oben erwähnt spielen die absoluten Werte der erlernten Ähnlichkeitsfunktion bei COSIMIR und beim Information Retrieval allgemein nur eine sekundäre Rolle. Wichtig ist die Reihenfolge bei der Sortierung nach Ähnlichkeit. Dadurch kann der optimale Zeitpunkt für das Ende des Trainings nicht durch Fehlermaße des Netzes bestimmt werden, sondern nur durch die nachgeordnete Evaluierung. Dies erhöht den Aufwand für die Experimente zusätzlich. Trotz der Reduktion und der relativ kleinen Kollektion ist der Aufwand für das Training und die Evaluierung beträchtlich. Dieser Aufwand ist aber immer nur für das erneute Training des Netzes mit den Input-Daten erforderlich, wenn etwa größere Mengen neuer Daten hinzukommen. Im Einsatz dagegen und damit für den Benutzer wäre das COSIMIR-Modell schnell, da der Recall bereits trainierter neuronaler Netze sehr schnell abläuft und dafür auf dem Software-Markt optimierte Programme zur Verfügung stehen.

Die Ergebnisse der Experimente mit Cranfield sind in allen Fällen weit schlechter als die Baseline. Während die Recall-Precision-Kurve für die Baseline hoch liegt, verläuft sie für das COSIMIR-Modell nahe der X-Achse.

Das COSIMIR-Modell erreicht für dieses Korpus nicht die Qualität eines Standardmodells.

7.2 Transformations-Netzwerk: Thesaurus zu Klassifikation

Die Evaluierung der Heterogenitätsbehandlung mit dem Transformations-Netzwerk erfordert Daten, die mit zwei Indexierungsschemata erschlossen sind. Die Cranfield-Kollektion kam also dafür nicht in Frage. Statt dessen wurden Daten aus dem Bereich der Sozialwissenschaften benutzt.

7.2.1 Datenbanken des Informationszentrum Sozialwissenschaften

Das Informationszentrum Sozialwissenschaften in Bonn (IZ) betreibt die Datenbanken SOLIS und FORIS, die über verschiedene online-Hosts und als CD-ROM zugänglich sind (cf. Zimmer 1998, 1998a). SOLIS ist eine Literaturdatenbank für die Sozialwissenschaft, die Nachweise zu selbständigen und unselbständigen Publikationen enthält. Am IZ verschlagworten Indexierer die Dokumente intellektuell und erstellen Abstracts, soweit diese nicht vorhanden sind. Daneben pflegt das IZ die Projektdatenbank FORIS, die mit den Ergebnissen einer jährlichen Umfrage bei einschlägigen Forschungseinrichtungen in den Sozialwissenschaften und benachbarten Disziplinen gefüllt wird. Auch die Dokumente in FORIS werden intellektuell nach dem IZ-Thesaurus (cf. Informationszentrum Sozialwissenschaften 1997) indexiert.

Der Thesaurus umfasst ca. 10.500 Terme, wovon 6900 als Deskriptoren vergeben werden können. Die übrigen Terme werden im Sinne der Einheitlichkeit des Indexierungsvokabulars nicht benutzt, bei ihnen gibt der Thesuarus den Begriff an, den der Indexierer benutzen soll. Darüber hinaus enthält der Thesuarus für viele Einträge Über-, Unter und verwandte Begriffe. Daneben werden alle Dokumente zusätzlich in einer Klassifikation von Fachgebieten eingeordnet, die 159 Einträge enthält. Die Disziplinen sind dabei aus dem Blickwinkel der Sozialwissenschaften geordnet. Sowohl der Thesaurus als auch die Klassifikation liegen in elektronischer Form vor (cf. Riege 1998).

Im Rahmen der crosslingualen TREC-Experimente werden IZ-Daten bei TREC (cf. Abschnitt 2.1.4.2) und der neuen europäischen Initiative Cross-Language Evaluation Forum (CLEF)[1] als Experimentierdaten angeboten. Zuvor wurden diese Daten bereits im Rahmen der German Indexing and Retrieval Test (GIRT) Initiative zur Verfügung gestellt (cf. Kluck 1998,

[1] http://www.iei.pi.cnr.it/DELOS/CLEF

Knorz 1997). Dazu wurde ein Ausschnitt aus SOLIS und FORIS gebildet, Anfragen formuliert und relevante Dokumente gesucht. Bei den TREC-Experimenten bilden die IZ-Daten gemeinsam mit Texten der Neuen Züricher Zeitung aus der gleichen Zeit einen Pool, aus dem mit einer Anfrage sowohl sozialwissenschaftliche Fachtexte als auch Zeitungstexte recherchiert werden.

Für das hier besprochene Experiment wurden 12.965 Dokumente aus SOLIS und FORIS benutzt, die ursprünglich die GIRT-Kollektion bildeten und nun in dem größeren CLEF-Korpus aufgegangen sind. Von dieser Grundmenge bilden 12.000 Dokumente die Trainingsmenge und der Rest die Testmenge. Alle Ergebnisse in diesem Abschnitt gehen von dieser Datenmenge aus und beziehen sich auf die Testmenge.

Laut Aussagen der Indexierer am IZ sind Thesaurus und Klassifikation unabhängig voneinander und formalisierbare Beziehungen zwischen ihnen existieren nicht. Das Transformations-Netzwerk versucht in diesem Experiment, eine Abbildung zwischen diesen zwei unterschiedlichen intellektuellen Indexierungsschemata zu leisten.

Ein automatisiertes System, das diese Transformation vornimmt, könnte innerhalb des IZ und für bestimmte Nutzergruppen gewinnbringend eingesetzt werden. Auch beim Retrieval kann die Transformation für Benutzer sinnvoll sein, die immer mit der Klassifikation arbeiten und mit ihr vertraut sind. Würde nun aus Kostengründen die Indexierung nach der Klassifikation eingestellt, könnte ein automatisches System diesen Zugang durch die Transformation aufrecht erhalten. Ein weiterer Anwendungsfall ergibt sich aus dem Schalenmodell, in dem Objekte von einer Form der Inhaltserschließung in eine andere übertragen werden (cf. Abschnitt 5.1.2). Sollten externe Informationsanbieter ihre Dokumente nur mit dem IZ-Thesaurus erschließen, so werden diese Dokumente auch für Benutzergruppen zugänglich, die mit der Klassifikation vertraut sind.

Unabhängig vom Retrieval könnte eine solches System die Indexierer unterstützen und nach Vergabe der Terme aus dem Thesaurus Terme aus der Klassifikation vorschlagen. Einen solchen Vorschlagmodus realisieren auch andere Transformations-Systeme (cf. Abschnitt 5.3.2.1).

7.2.2 Transformations-Netzwerk und LSI

Für jedes Dokument sind nach dieser Kumulierung durchschnittlich dreizehn Terme und 2,3 Klassifikations-Einträgen vergeben. Besonders für die Thesaurus-Terme schwankt die Zahl jedoch stark, nämlich zwischen zwei und 39. Für die ca. 13.000 Dokumente liegen sowohl die Thesaurus-Einträge und die Klassifikation vor. Darin sind 5555 Thesaurus-Begriffe und 142 der 159

Klassifikations-Einträge belegt. Für den Test werden auf der Input Seite nur 3800 Terme ausgewählt, die mindestens in vier Dokumenten vorkommen Ansonsten kann kaum sichergestellt werden, dass jeweils alle Terme, die in der Testmenge enthalten sind, auch trainiert werden. Als gewünschter Ziel-Output gibt das System nicht alle 142 Klassen auf der untersten Hierarchieebene der Klassifikation vor, sondern 70 intellektuell kumulierte Klassen, was zu einer günstigeren Anzahl von Beispielen pro Klasse führt. Auch dadurch wird versucht, mit der Testmenge nichts zu prüfen, was vorher nicht mit Trainingsmenge gelernt wurde.

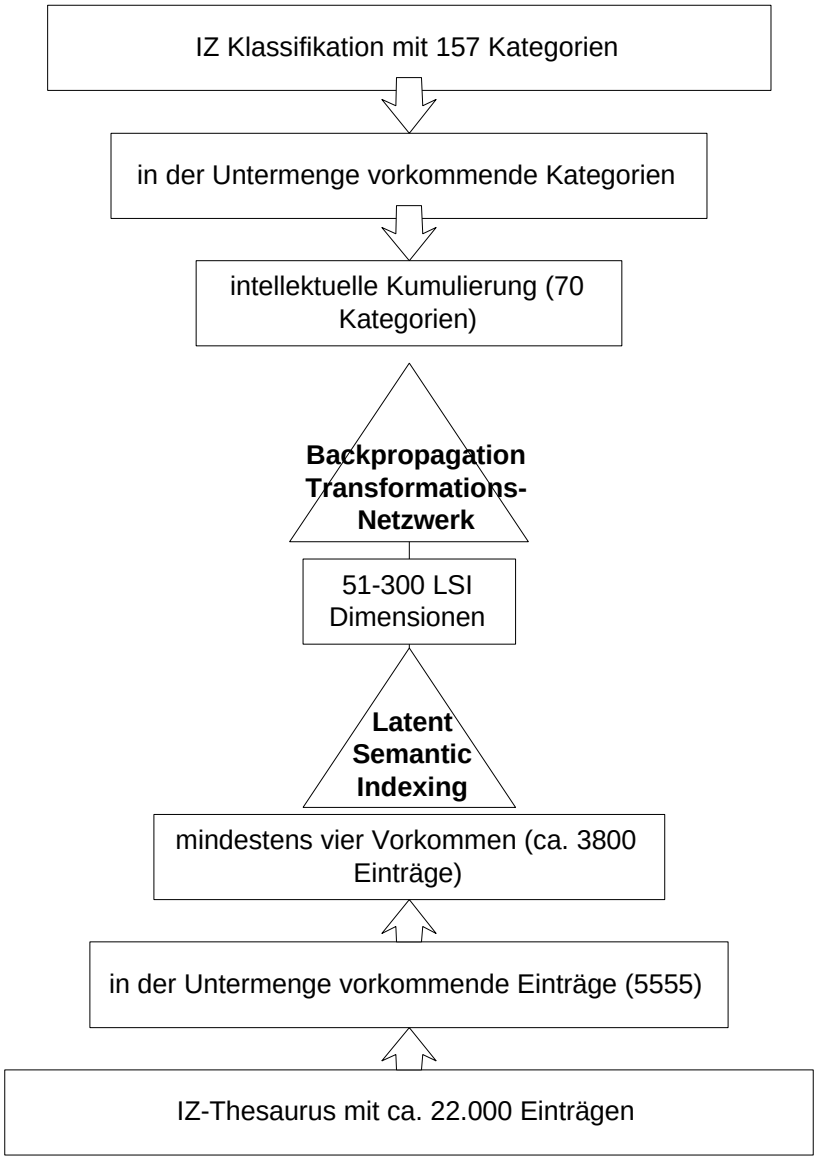


Abbildung 7-3: Schema der Transformation vom IZ-Thesaurus zur IZ-Klassifikation

Die Berücksichtigung von ca. 3800 Thesaurus-Termen, die mindestens viermal vorkommen, führt zu sehr langen Vektoren und damit zu sehr großen Netzen. Diese sind zum einen mit gängiger Hardware nicht trainierbar und die Anzahl der Trainingsbeispiele sind im Vergleich zu den Verbindungen zwischen den Neuronen zu gering. Die Faustregel von Bigus 1996 fordert zwei Muster pro Verbindung. Es stehen jedoch nur 12.000 Muster zur Verfügung und bei nur zehn versteckten Neuronen ergibt sich bereits ein Netz von 38.700 Verbindungen.

Wie bei den Experimenten mit der Cranfield-Kollektion (cf. Anschnitt 7.1.3) wurde diese Problematik auch hier durch eine Komprimierung mit Latent Semantic Indexing (LSI) gelöst. Die 3800 Dimensionen wurden nach Tests mit Werten zwischen 50 und 300 auf 107 LSI-Faktoren reduziert.

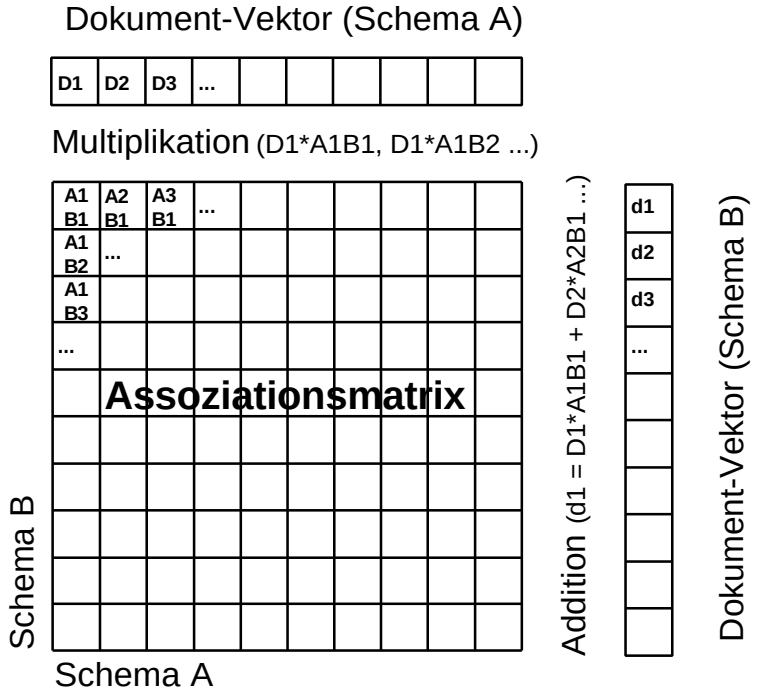


Abbildung 7-4: Statistische Transformation als Baseline

Durch diese Reduktion der Dimensionen mit LSI bleibt das Netz klein. Die Experimente erfolgten mit der Software DataEngine (cf. Abschnitt 3.6.2). Die Netze konvergieren dabei sehr schnell. Nach etwa 30 bis 40 Epochen änderten sich der Trainings- und Testfehler kaum mehr. Die Größe der versteckten Schicht wurde durch den Test mehrerer Konfigurationen optimiert. Das beste Netz besitzt zwanzig versteckte Neuronen.

Als Vergleichsmaßstab (Baseline) dient ein statistisches Verfahren, das aufgrund gemeinsamen Vorkommens von Termen in Dokumenten Assoziationen

(cf. Abschnitt 5.3.1) zwischen Thesaurus und Klassifikation berechnet. Dazu wurden die Kookkurrenzen berechnet und in einer Assoziationsmatrix festgehalten. Diese Matrix dient als Grundlage der Transformation wie in Abbildung 7-4 angedeutet.

7.2.3 Ergebnisse

Beim Training des neuronalen Netzes stellt sich ein ähnliches Problem wie beim COSIMIR-Modell (vgl. Abschnitt 7.1.3), der Testfehler des Netzes entspricht nicht dem endgültigen Qualitätsmaß für die Transformation. Das Netz misst für jedes Gewicht eines Terms die Übereinstimmung des Trainingswertes mit dem Ergebnis des Netzes. Die möglichst exakte Übereinstimmung der Gewichte gibt aber nicht den Ausschlag für eine erfolgreiche Transformation.

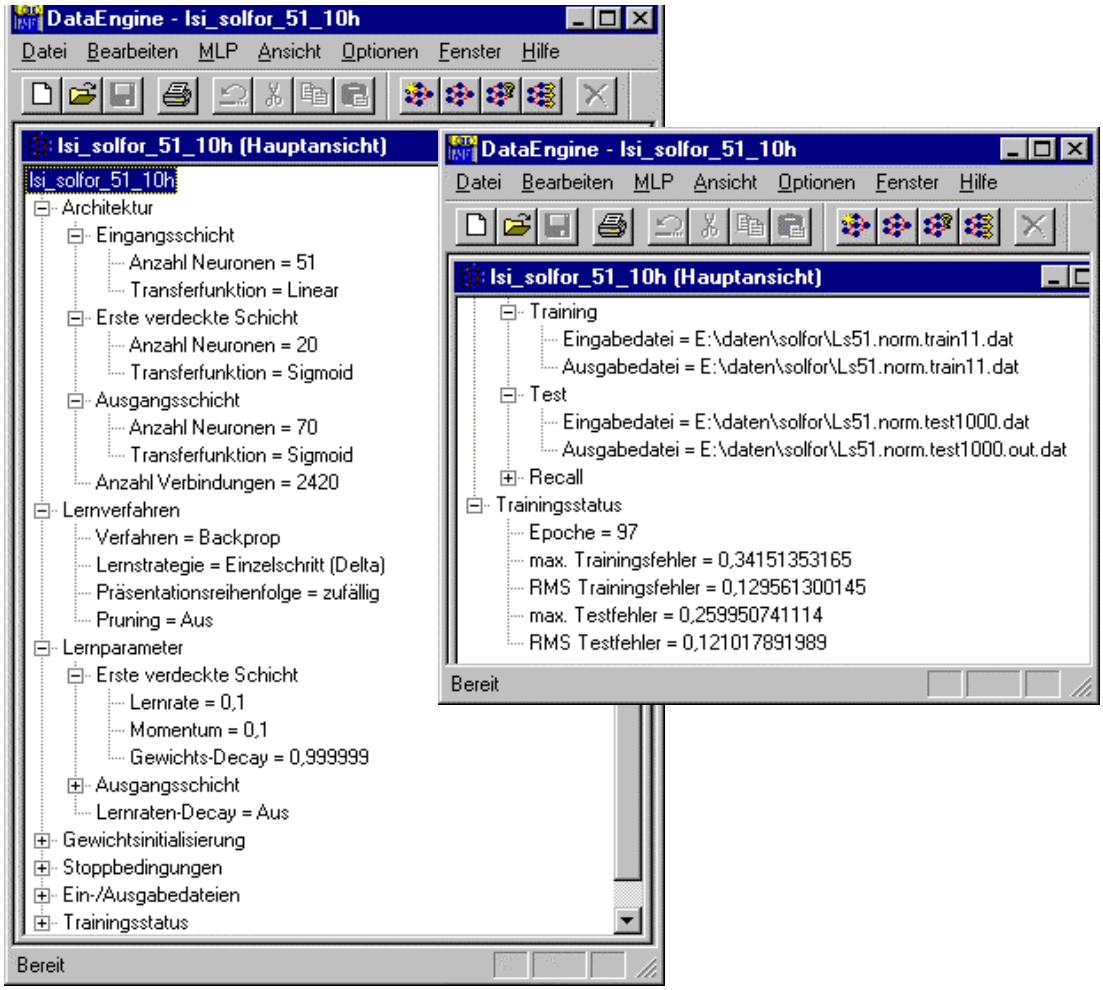
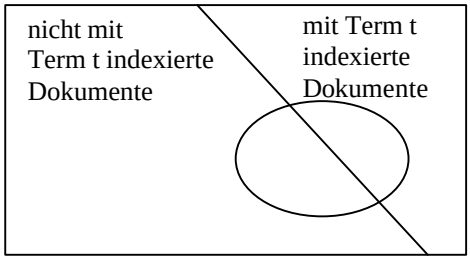


Abbildung 7-5: Netzwerk, Experimentparameter und Ergebnisse in DataEngine

So wie beim normalen Retrieval nicht der Absolutwert der Retrieval Status Value entscheidend ist, sondern die sich ergebende Reihenfolge der Dokumente, kommt es auch bei der Bewertung der Transformation in der Regel auf die Reihenfolge der Gewichte an. Das Training des neuronalen Netzes verringert primär den Fehler in der Testmenge. Im vorliegenden Fall ergeben sich durchschnittliche, quadrierte Fehler von ca. 0,1. Abbildung 7-5 zeigt beispielhaft die Parameter und Ergebnisse eines Netzes während des Trainings.

Die üblichen Maße für die Qualität von Transformationen sind Recall und Precision bezogen auf die Terme des Zielvokabulars (cf. z.B. Yang 1995, Apté et al. 1994, Lam/Ho 1998). Dabei ersetzen die Terme des Zielvokabulars die Anfragen, auf die sich Recall und Precision beim Retrieval beziehen. Ausgangspunkt ist beim Retrieval eine Anfrage und bei den Transformationen ein Term. Ergebnis ist in beiden Fällen eine Menge von Dokumenten. Eine Transformation bestimmt die Menge von Dokumenten, der sie den Term aus dem Zielvokabular zuordnet. Diese Menge wird dann mit den tatsächlichen Zuordnungen verglichen oder, falls diese nicht vorhanden ist, nachträglich von Experten beurteilt (cf. Abbildung 7-6). Im zweiten Fall entsteht das analoge Problem zur Recall beim Retrieval, dass die Zielmenge nicht genau bekannt ist und nur durch sehr hohen intellektuellen Aufwand bestimmt werden kann. Bei der Verwendung vager Verfahren zur Transformation entsteht in der Regel keine Menge von zugeordneten Dokumenten, sondern jedes Dokument erhält ein Gewicht für jeden Term. Damit muss auch bei Term-Recall und Term-Precision ein Recall-Precision-Graph erstellt werden. Abbildung 7-7 zeigt das Ergebnis der Transformation für das Transformations-Netzwerk und für das Vergleichsexperiment mit einem statistischen Verfahren.

Wie Abschnitt 2.1.4.1 demonstriert, sind diese Maße problematisch. Im Falle der Transformationen gilt dies um so mehr. Zwar ergibt sich formal eine völlig analoge Formel (cf. Abbildung 7-6), sie misst aber nicht die Retrieval-Qualität, wie die Begriffe Recall und Precision andeuten. Während beim Retrieval der Erfolg vom Urteil des Benutzers abhängt und daraus berechnet wird, spielt dieser Gesichtspunkt bei den Transformationen zunächst keine Rolle. Hier bestimmen Recall und Precision die Qualität einer Zuordnung von Termen, also einer automatischen Indexierung auf Basis einer bereits vorliegenden Indexierung. Erst der Einsatz dieser Indexierung im Retrieval zeigt ihren Wert für den Benutzer. Unabhängig davon kann eine Transformation als Unterstützungssystem für die Indexierung verwendet werden und z.B. einem Indexierer Vorschläge machen (vgl. Abschnitt 5.3.2.1). In diesem Fall ist die beschriebene Form der Evaluierung adäquater. Natürlich ist eine gute Indexierung die Basis für erfolgreiche Retrieval und somit hängen die beiden Formen des Recall zusammen.



	Von Transformation vorgeschlagen	Von Transformation nicht vorgeschlagen
Indexiert	R_E	R_N
Nicht indexiert	N_E	N_N

Der Term t aus dem Indexierungsschema B ist einigen der Dokumente zugeordnet (rechts der schrägen Linie). Das Ergebnis der Transformation ist eine Menge von Dokumenten (Oval), für die Term t vorgeschlagen wird.

$$\text{Tern-Recall} = \frac{R_E}{R_E + R_N}$$

$$\text{Term-Precision} = \frac{R_E}{R_E + N_E}$$

Abbildung 7-6: Term-Recall und Term-Precision

Um die angeführten Bedenken zu verdeutlichen und der Verwechslungsgefahr zu begegnen, werden im Folgenden im Gegensatz zur Literatur die Qualitätsmaße für Transformationen als Term-Recall und Term-Precision bezeichnet. Diese Benennung verdeutlicht, dass die Grundlage ihrer Berechnung in der Vergabe von Termen und nicht in einer Anfrage besteht. Weiterhin erfolgt eine zusätzliche Bewertung der Ergebnisse anhand einer anschaulicheren Methode.

Abbildung 7-7 stellt die Ergebnisse des Transformations-Netzwerks und der Baseline gegenüber. Die Kurven verlaufen fast gleichförmig, so dass die Qualität der beiden Verfahren für diesen Anwendungsfall praktisch identisch ist. Insgesamt liegen die Werte eher niedrig, die durchschnittliche Precision beträgt 0,19. In anderen Experimenten wird von höheren Werten berichtet. Yang 1995 z.B. erzielte durchschnittliche Precision-Werte zwischen 0,32 und 0,88 bei der Anwendung statistischer Verfahren für verschiedene Kollektionen. Dieses relativ niedrige Ergebnis lässt vermuten, dass in diesem Anwendungsfall die Transformationen sehr schwierig sind.

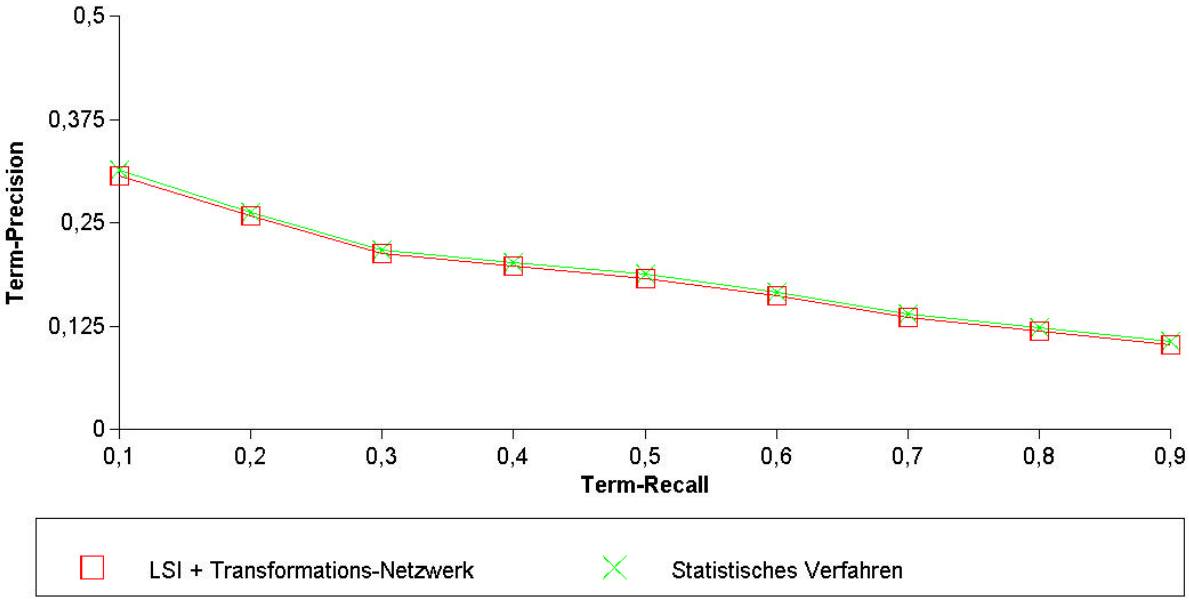


Abbildung 7-7: Ergebnis als Recall-Precision-Grafik

Ein erster Vergleich legt zunächst die Interpretation nahe, die Ergebnisse seien identisch. Eine nähere Betrachtung zeigt aber, dass nur die Qualität sehr ähnlich ist, die Ergebnisse dagegen weitgehend unterschiedlich. Dazu werden die Rangfolgen verglichen, die sich ergeben, wenn beide Verfahren ausgehend von einem Term den Dokumenten ein Gewicht zu diesem Term zuordnen. Um diese Rangfolgen vergleichen zu können, eignet sich der der häufig in der Literatur genannte Spearmansche Rangfolgenkoeffizient:

$$\text{Spearman:} \quad r = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

Hartung 1984:191, Clauß/Ebner 1979:126

Berechnet man den Spearman-Koeffizienten für alle 70 Zielklassen für die 1000 Dokumente in der Testmenge und den Durchschnitt daraus, so ergibt sich ein Wert von –0,05. Dies weist darauf hin, dass praktisch keine Korrelation zwischen den Rangfolgen besteht und die zwei Verfahren die Dokumente völlig unterschiedlich sortieren.

Eine weitere Analyse veranschaulicht die Unterschiedlichkeit der Ergebnisse noch besser. Sie untersucht die Schnittmengen der Ergebnisse der beiden Verfahren. Dazu wurden für jeden Term die ersten 20 Dokumente aus der Rangfolge gewählt und mit den 20 ersten Dokumenten des anderen Verfah-

rens verglichen. Im Durchschnitt ergeben sich für die 70 Terme in der Klassifikation nur 0,27 gemeinsame Dokumente von 20. Betrachtet man nur die relevanten Dokumente unter den ersten 20, also die mit einer korrekten Zuordnung, sinkt dieser Wert auf 0,14. Dies zeigt sehr deutlich, dass jedes Verfahren andere Treffer bringt. Die gleiche Analyse für die ersten 100 Dokumente führt zu einer durchschnittlichen Größe der Schnittmenge von 4,92 Dokumenten und zu 0,83 gemeinsamen relevanten Dokumenten. Tabelle 7-1 fasst diese Ergebnisse zusammen.

Tabelle 7-1: Schnittmengen aus den besten Dokumenten

	Größe der Schnittmenge	**in %**	**Größe der Schnittmenge der relevanten Dokumente**	**in %**
Ersten 20 Dokumente	0,27	5,43	0,14	0,71
Ersten 100 Dokumente	4,92	4,92	0,83	0,83

Diese Analyse erinnert an Ergebnisse der TREC-Konferenz (cf. Abschnitt 2.1.4.2). Dort hat sich ebenfalls gezeigt, dass die besten IR-Systeme sehr ähnliche Qualität erreichen, dass sich ihre Ergebnisse jedoch stark unterscheiden. Dies führte zur Entwicklung von Fusionsverfahren, die dies ausnutzen. Derartige Ansätze der Mehrfachindexierung (cf. Abschnitt 2.3.1.2) versuchen, mehrere Verfahren zu kombinieren, um so insgesamt mehr relevante Dokumente finden. Das Gesamtergebnis besteht aus einer Kombination mehrerer Ergebnismengen unterschiedlicher Verfahren (cf. Womser-Hacker 1997).

Dies bedeutet auch, dass die Optimierung nicht nur die Suche nach dem besten Information Retrieval Verfahren bedeutet. Vielmehr muss ein neuartiges Verfahren nicht notwendigerweise alle anderen übertreffen. Ein neues IR-Verfahren dessen Qualität mit bestehenden Verfahren vergleichbar ist, das aber eine weitgehend andere Ergebnismenge und damit auch andere Treffer bringt, kann bei der Fusion einen positiven Beitrag leisten. Diese Überlegungen gelten auch für die Heterogenitätsbehandlung.

Die Werte von Term-Recall und Term-Precision sind zwar die üblichen Bewertungsmaßstäbe für Transformationen, sie sind jedoch nicht sehr anschaulich. Die folgende zusätzliche Bewertung des Ergebnisses überträgt den Erfolg des Verfahrens auf einzelne Dokumente und zeigt so, inwieweit sich die Transformation als Vorschlagmodus eignet. In der Testmenge von 1000 Dokumenten sind 69% aller Zuordnungen und 97% aller Nicht-Zuordnungen richtig. Dies bedeutet, dass pro Dokument 1,4 von 2,0 Einträge und 32,4 von

35,9 Nicht-Zuordnungen richtig erkannt werden, bzw. dass pro Dokument 3,8 Einträge falsch hinzukommen. Dies scheint als erste Annäherung befriedigend. Bei einer Anwendung als Vorschlagsmodus müssen vom menschlichen Indexierer v.a. nicht passende Einträge gelöscht werden, was einfacher ist als neue passende zu finden. Bei dieser Interpretation der Ergebnisse wird deutlich, dass eine stark unterschiedliche Anzahl von Trainingsbeispielen für die einzelnen Klassen die Aufgabe für das Netz erschwert. Die Anzahl der Beispiele für eine Zielklasse schwankt in der Trainingsmenge zwischen vier und 3300, der Durchschnitt in der Gesamtmenge beträgt 206. Erwartungsgemäß ergibt sich eine starke Korrelation zwischen Anzahl von Trainingsbeispielen und Treffern (cf. Abbildung 7-8).

Abbildung 7-8: Überblick über die Ergebnisse der Transformation von IZ-Thesaurus zu IZ-Klassifikation

Eine ausgewogene Verteilung von Trainingsbeispielen auf Klassen könnte die Qualität wohl noch verbessern, jedoch ist sie in der Praxis schwierig zu erreichen. In realen Datenbeständen lässt sich die Verteilung der Häufigkeit der Terme nicht steuern. Zumindest sollte darauf geachtet werden, dass für einzelne Klassen nicht nur sehr wenige oder überhaupt keine Trainingsbeispiele vorliegen.

7.3 Transfomations-Netzwerk: Kölner Bibliotheks-Thesaurus zu IZ-Repräsentationen

Die Ergebnisse des Transformations-Netzwerks für den IZ-Thesaurus und die IZ-Klassifikation sind ermutigend. Häufig hängt die Qualität von Information Retrieval Verfahren stark vom jeweiligen Datenbestand ab. Weitere Experiment sollten zeigen, ob dies auch für Transformationen gilt, oder ob sich die Aussagen auf andere Datenbestände übertragen lassen. Aus dem Bereich der Sozialwissenschaften liegt ein weiteres Doppelkorpus vor, womit weitere Experimente mit Transformationen durchgeführt wurden. Der Aufbau der Experimente ist im Wesentlichen analog zu dem im vorigen Abschnitt beschriebenen.

Teile der Literatur, die das Informationszentrum Sozialwissenschaften (IZ) intellektuell indexiert, erschließt auch das Sondersammelgebiet (SSG) Sozialwissenschaft der Universitäts- und Stadtbibliothek (USB) Köln. Die Daten aus dem SSG enthalten ca. 10.400 Terme. Die Daten beschreibt eine Projektskizze von Kluck et al. 2000. Das darin geplante Projekt bearbeitet inhaltliche Probleme der Heterogenität, wie sie Kapitel 5 beschreibt. Als Datengrundlage dienen wieder die Datenbanken des IZ Sozialwissenschaften, das Sondersammelgebiet der USB Köln, sozialwissenschaftliche Dokumente der Friedrich-Ebert-Stiftung und die Daten des Darmstädter Virtuellen Gesamtkatalogs der TU Darmstadt. Ziel ist die Erstellung einer virtuellen digitalen Bibliothek, die dem Benutzer flexiblen Zugriff auf alle diese Daten ermöglicht, ohne dass die heterogenen Erschließungsverfahren zu semantischen Problemen führen.

Das in den beiden folgenden Abschnitten beschriebenen Experimente benutzen doppelt indexierte Text-Dokumente vom IZ und dem SSG der USB Köln. Die Repräsentationen wurden also im Gegensatz zum ersten Experiment von unterschiedlichen Indexierern erstellt.

7.3.1 USB-Thesaurus zu IZ-Klassifikation

In einem ersten Schritt konnten mit Hilfe des IZ und der USB 1979 Dokumente identifiziert werden, die sowohl mit den Termen der USB und mit der Klassifikation des IZ indexiert sind. Davon wurden 1779 für das Training und 200 für den Test benutzt. Für die Dokumente sind im Durchschnitt 2,2 Terme und maximal sieben Terme aus dem USB-Thesaurus vergeben. Die Aufgabe ist damit sehr schwierig, da zum einen wenig doppelt indexierte Dokumente zur Verfügung stehen und weiterhin wenig USB-Terme vergeben sind, so dass automatische Verfahren wenig Evidenz für die Transformation erhalten.

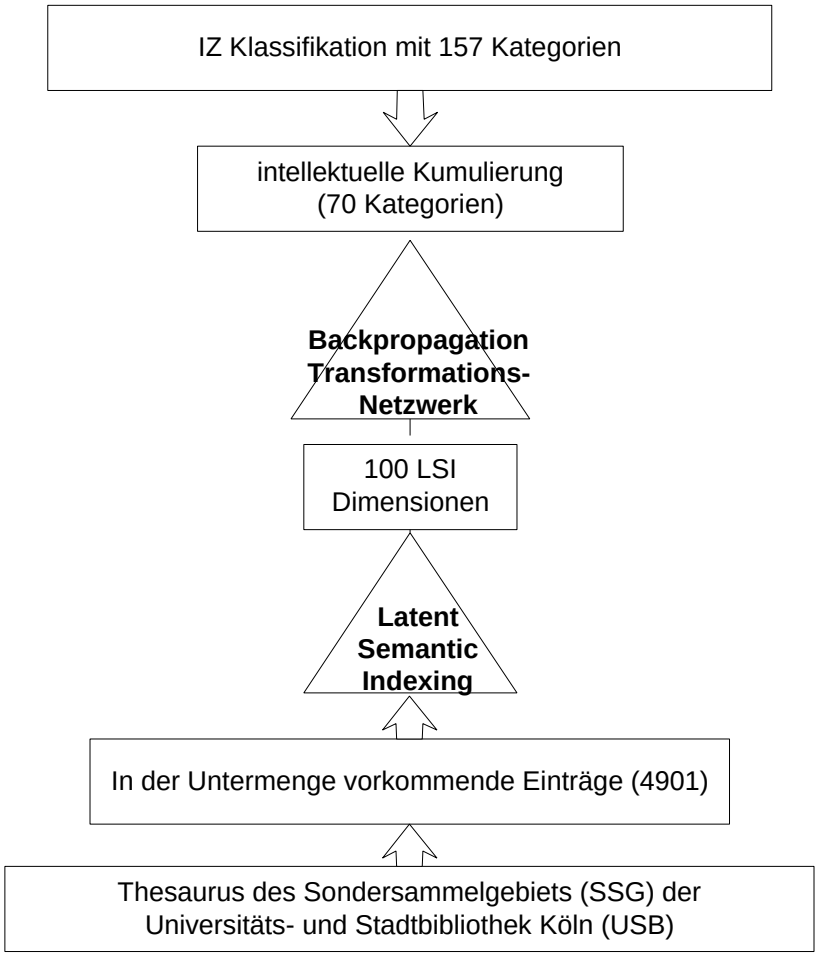


Abbildung 7-9: Schema der Transformation vom USB-Thesaurus zur IZ-Klassifikation

Für die IZ-Klassifikation werden wie im Experiment im vorigen Abschnitt einige inhaltlich sehr nahe verwandte Klassen zusammengelegt, so dass insgesamt 70 Klassen vorliegen. Die verwendeten Dokumente umfassen ca. 4900 Terme des USB-Thesaurus. Um diese mit einem Transformations-Netzwerk bearbeiten zu können erfolgt eine Reduktion mit Latent Semantic Indexing. Den Input für das Transformations-Netzwerk liefern 100 der errechneten LSI-Dimensionen. Das neuronale Netz besteht also aus 100 Input-Neuronen und 70 Output-Neuronen für die IZ-Klassifikation. Dazwischen liegt eine versteckte Schicht mit 20 Neuronen. Den Ablauf der Transformation skizziert Abbildung 7-9.

Als Vergleichsmaßstab dient eine Transformation der gleichen Daten mit einem statistischen Verfahren auf Basis einer Assoziationsberechnung. Mit der für das Training verwendeten Menge wird eine statistische Transformation erstellt, die für die Dokumente der Test-Menge durchgeführt und bewertet

wird. Die Bewertung anhand der Term-Precision und des Term-Recalls erfolgt analog zu dem in Abschnitt 7.2 beschriebenen Experiment.

Die Resultate zeigen eine höhere Term-Precision für das Transformations-Netzwerk als die auf LSI basierende Transformation durch eine statistische Assoziation. Die durchschnittliche Term-Precision liegt für das Transformations-Netzwerk bei 0,11 und für die statistische Transformation bei 0,064. Bei niedrigem Term-Recall ist das Verfahren sogar fast 100% besser, während die beiden Kurven sich bei hohen Recall-Werten stärker annähern wie Abbildung 7-10 zeigt.

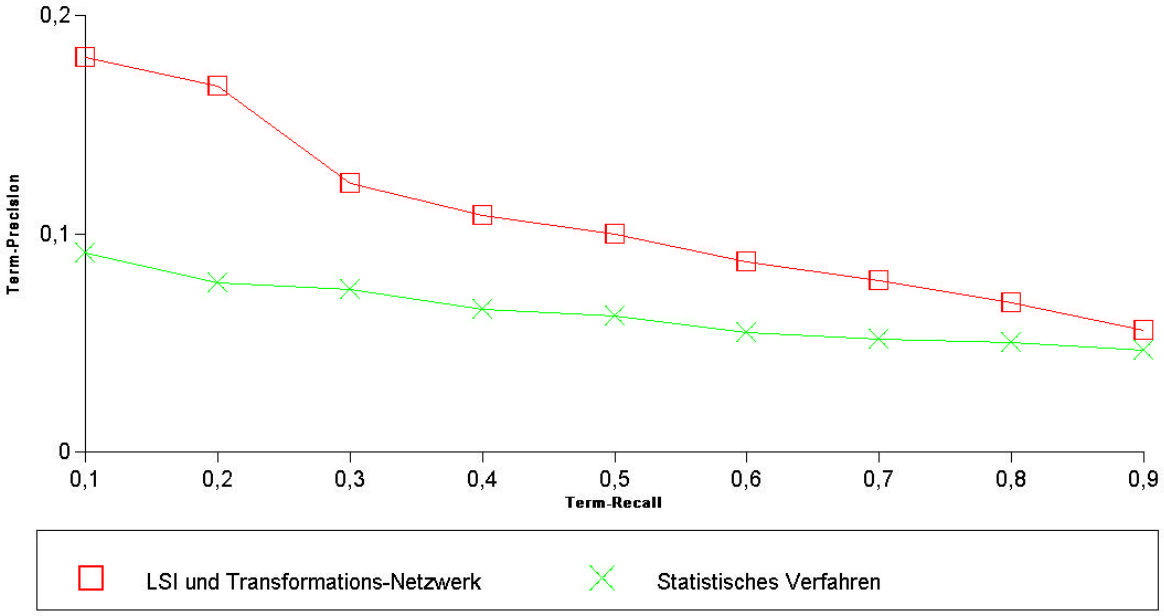


Abbildung 7-10: Ergebnis der Transformation vom USB-Thesaurus zur IZ-Klassifikation als Recall-Precision-Grafik

Allerdings darf dieses Ergebnis nicht überbewertet werden. Die Anzahl der benutzten Dokumente ist insgesamt niedrig und die Term-Precision Werte liegen insgesamt eher niedrig. Dieses insgesamt zufriedenstellende Resultat rechtfertigte die Erstellung eines größeren Korpus für ein weiteres Experiment mit größerer Aussagekraft.

7.3.2 USB-Thesaurus zu IZ-Thesaurus

Im zweiten Schritt wurden ca. 15.000 Dokumente identifiziert, die sowohl vom IZ als auch vom SSG inhaltlich erschlossen wurden. Damit ist eine Abbildung auch auf den detaillierten IZ-Thesaurus möglich. Die bisherigen Transformationen hatten als Ziel immer die IZ-Klassifikation mit relativ wenig Einträgen und Zuordnungen pro Dokument. Da die Anzahl der

Dokumente pro IZ-Thesaurus-Term sehr unterschiedlich ist, fokussiert das Experiment auf die 100 häufigsten Terme im Korpus.

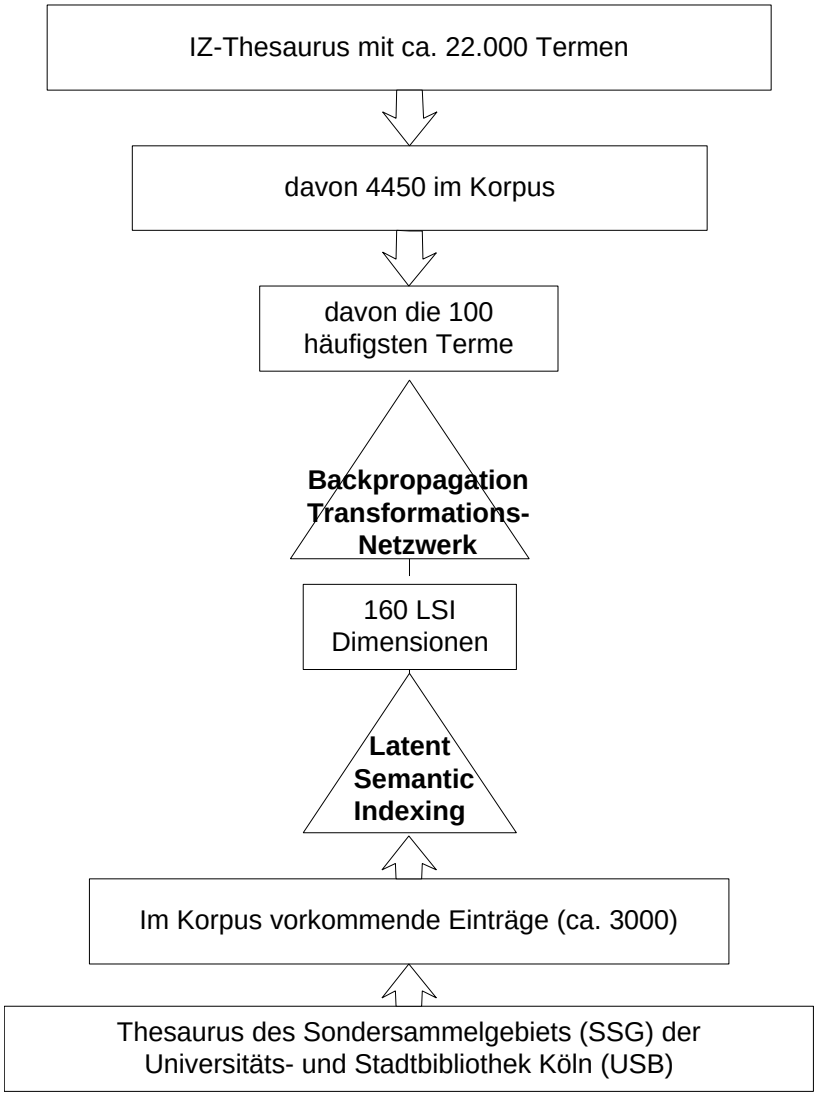


Abbildung 7-11: Schema der Transformation vom USB-Thesaurus zum IZ-Thesaurus

Das IZ hat für die doppelt indexierten Dokumente durchschnittlich 11,2 Terme vergeben, die USB 2,9. Die maximale Anzahl von Termen pro Dokument beträgt 56 beim IZ bzw. 19 bei der USB. Die USB hat 10% der Terme nur einmal in dem Korpus vergeben und weitere 21% nur zweimal. Beim IZ liegen diese Werte etwas höher und erreichen 26% bzw. 36%. Insgesamt wird in dieser Menge ein IZ-Term durchschnittlich 38 und ein USB-Term durchschnittlich 15 Dokumenten zugeteilt.

Wie in dem vorhergehenden Experiment komprimiert LSI die USB-Repräsentation der Objekte und zwar in diesem Fall auf 160 Dimensionen. Somit ergibt sich der in Abbildung 7-11 skizzierte Ablauf der Transformation. Die Trainingsmenge umfasst 13.000 Dokumente und die Testmenge 2238 Dokumente.

Die Qualität der Transformation für die Dokumente in der Test-Menge messen wieder die Größen Term-Precision und Term-Recall (cf. Abschnitt 7.2.3). Im Ergebnis zeigt sich eine wesentlich höhere Term-Precision für die Kombination aus LSI und neuronalem Transformations-Netzwerk als für die auf LSI basierende statistische Transformation. Die durchschnittliche Term-Precision liegt für das Transformations-Netzwerk bei 0,24 und für die statistische Transformation bei 0,044. Die Recall-Precision-Kurve verläuft für das neuronale Netz erheblich besser als für das statistische Verfahren, wie Abbildung 7-12 zeigt.

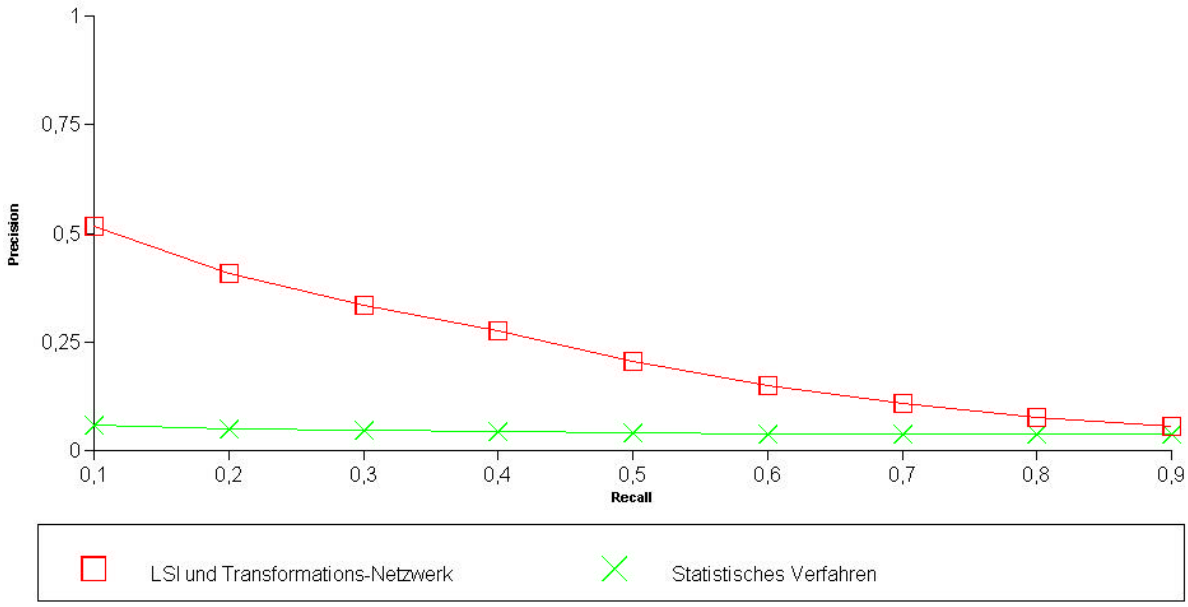


Abbildung 7-12: Ergebnis der Transformation USB-Thesaurus zu IZ- Thesaurus als Recall-Precision-Grafik

Das gute Abschneiden des neuronalen Netzes gegenüber dem statistischen Verfahren bei diesem Experiment weist darauf hin, dass die Ergebnisse sehr stark von den Daten abhängen. Welches das optimale Verfahren ist, muss für jeden Anwendungsfall erneut geprüft werden.

7.4 Experimente mit Faktendaten

Für weitere Experimente mit dem COSIMIR-Modell und erste Experimente mit dem COSIMIR-Modell für Heterogenitätsbehandlung dient ein Datenbestand aus dem Bereich Faktenretrieval, bei dem relativ kurze Vektoren die Retrieval-Objekte beschreiben. Für die Evaluierung von COSIMIR-Netzen ist grundsätzlich eine große Anzahl von Benutzerurteilen notwendig. Diese Urteile sind selten vorhanden und stehen für kein bekanntes Korpus zur Verfügung. Durch geeignete Auswahl von Trainingsdaten konnte für die vorliegenden Faktendaten ohne den Aufwand umfangreicher intellektueller Bewertungen experimentiert werden.

7.4.1 Datengrundlage: Werkstoffdaten

Der verwendete Datensatz besteht aus Werkstoffen, die aus der Datengrundlage des Projekts WING stammen (cf. Abschnitt 2.2.3.1). Ein Werkstoff wird durch zwei Vektoren beschrieben, seine Eigenschaften und sein Anwendungsprofil (cf. Tabelle 7-2 und Tabelle 7-3), wobei das Anwendungsprofil nur für eine Untermenge der Werkstoffe intellektuell erfasst wurde.

Tabelle 7-2: Beispielhafte Anwendungsvektoren von Werkstoffen

Werkstoff	**Ringe**	**Struktur-teile**	**Wel-len**	**Scheiben**	**Diffu-sor**	**Leit-kränze**	**...**
Legierung A1	0	0	1	1	0	1	...
Legierung A2	1	1	0	0	0	0	...

Tabelle 7-3: Beispielhafte Eigenschaftsvektoren von Werkstoffen

Werkstoff	**Anwen-dungs-temperatur**	**E-Modul**	**Zugfestig-keit**	**Bruch-Dehnung**	**Dehn-grenze**	**...**
Legierung A1	300	186	7,75	12,3	163	...
Legierung A2	400	140	6,78	9,7	129	...

Mandl 1994 und Ludwig/Mandl 1997 stellen das System NEURO-WING vor, das mit einem Backpropagation-Netz aus den Eigenschaften eines Werkstoffs dessen Anwendungsprofil ableitet. Wie Abbildung 7-13 zeigt, bestimmt das System nach Input eines Eigenschaftsvektors einen Anwendungsvektor für den gleichen Werkstoff. Das Netz lernt diese Abbildung anhand von Beispielen. Für diese Abbildung ist intuitives Expertenwissen notwendig, welches das Netz aus Beispielen lernt. Damit leistet NEURO-WING im Stile eines Transformations-Netzwerks (cf. Abschnitt 5.3.4) eine Abbildung zwischen heterogenen Repräsentationen.

Um im Anwendungsfall WING (cf. Abschnitt 2.2.3.1) Ähnlichkeitsretrieval zu ermöglichen, wurde auf Basis der Anwendungsvektoren die Ähnlichkeit zwischen den Werkstoffen durch eine mathematische Ähnlichkeitsfunktion bestimmt. Nach Expertenaussagen ergibt sich die Ähnlichkeit in den meisten Fällen aus der Anwendung und nicht aus den Eigenschaften. In der Praxis liegt nur der Eigenschaftsvektor für alle Werkstoffe vor und der Anwendungsvektor in der Regel nicht. NEURO-WING führte also die Transformation durch, um für alle Werkstoffe die Ausgangsdaten für eine Ähnlichkeitsberechnung zu erhalten. Laut Experten ist für diese Aufgabe menschliches Expertenwissen nötig (cf. Mandl 1994, Ludwig/Mandl 1997). Trotzdem war das System NEURO-WING erfolgreich. Die Qualität des Ähnlichkeitsretrieval stellte die Werkstoffexperten des industriellen Anwendungspartners insgesamt zufrieden (u.a. MTU, cf. Breitkopf et al. 1997).

7.4.2 COSIMIR mit Werkstoffdaten

Die Werkstoffdaten erlauben einen Test des COSIMIR-Modells ohne hohen intellektuellen Aufwand für die Erstellung von Expertenurteilen.

Für die Werkstoffe liegen mit Eigenschaftsvektor und Anwendungsvektor zwei heterogene Repräsentationen vor. Aus beiden lässt sich mit einer mathematischen Ähnlichkeitsfunktion jeweils eine Ähnlichkeitsmatrix berechnen. Die resultierenden Matrizen sind unterschiedlich (cf. Abschnitt 7.4.6). Die Methode zum Vergleich stellt Abschnitt 7.2.3 vor.

Für die Benutzer ist nach Auskunft der Werkstoffexperten die Ähnlichkeit nach der Anwendung ausschlaggebend (cf. Ludwig/Mandl 1997). Berechnet man aus den Anwendungsvektoren mit einer mathematischen Ähnlichkeitsfunktion wie dem Kosinus eine Ähnlichkeitsmatrix, so ist diese einer kognitiven Ähnlichkeitsfunktion sehr viel näher, als bei Verwendung der Eigenschaftsvektoren. Die Zufriedenheit der Experten mit solch einer berechneten Matrix stützt diese Annahme (cf. Mandl 1994). Deshalb nutzen die folgenden Tests die Ähnlichkeit anhand des Kosinus auf Basis der Werkstoff-Anwen-

dungen als gute Annäherung einer kognitiven Ähnlichkeitseinschätzung (cf. Abbildung 7-14). Diese berechnete Ähnlichkeit bildet den Ziel-Output der getesteten Modelle. In einigen Fällen wurde der Kosinus durch andere Ähnlichkeitsmaße ersetzt, um die Ergebnisse abzusichern

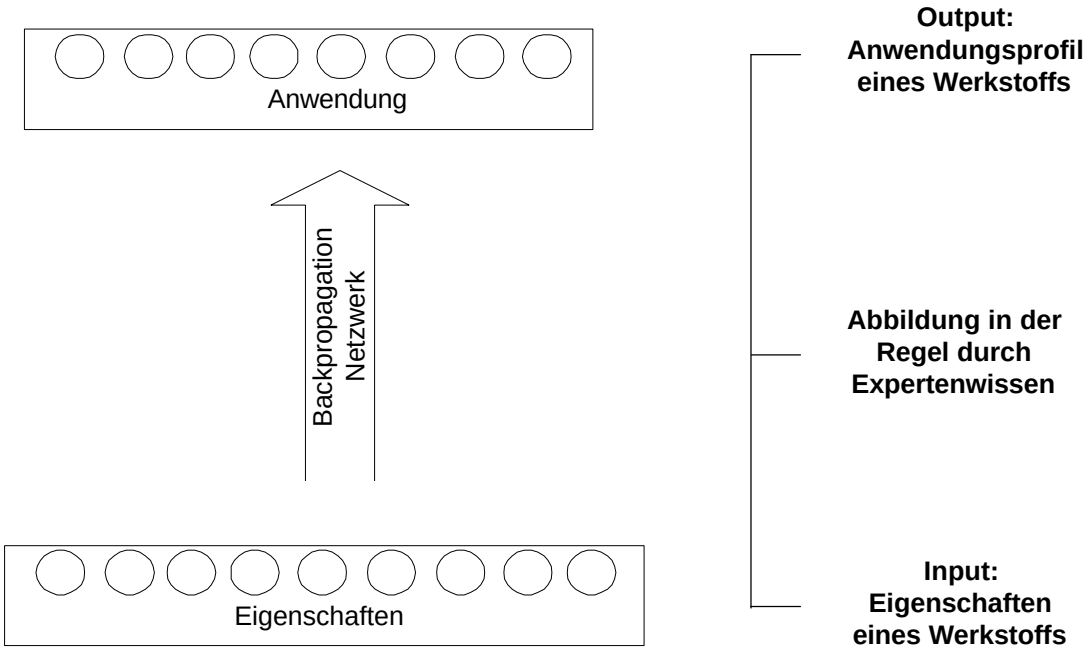


Abbildung 7-13: Architektur von NEURO-WING

Ein Pretest sollte zeigen, wie gut COSIMIR eine mathematische Ähnlichkeitsfunktion annähert. Dazu erhält das Netz als Input die Anwendungsvektoren zweier Werkstoffe und bestimmt daraus den Kosinus. In diesem Pretest ersetzt das COSIMIR-Netzwerk gewissermaßen die Kosinus-Formel. Das für die Praxis relevante Experiment skizziert Abbildung 7-15. Die Anwendungsvektoren werden durch die Eigenschaftsvektoren ersetzt. Input sind somit jeweils zwei Eigenschaftsvektoren und Output nach wie vor die Anwendungsähnlichkeit.

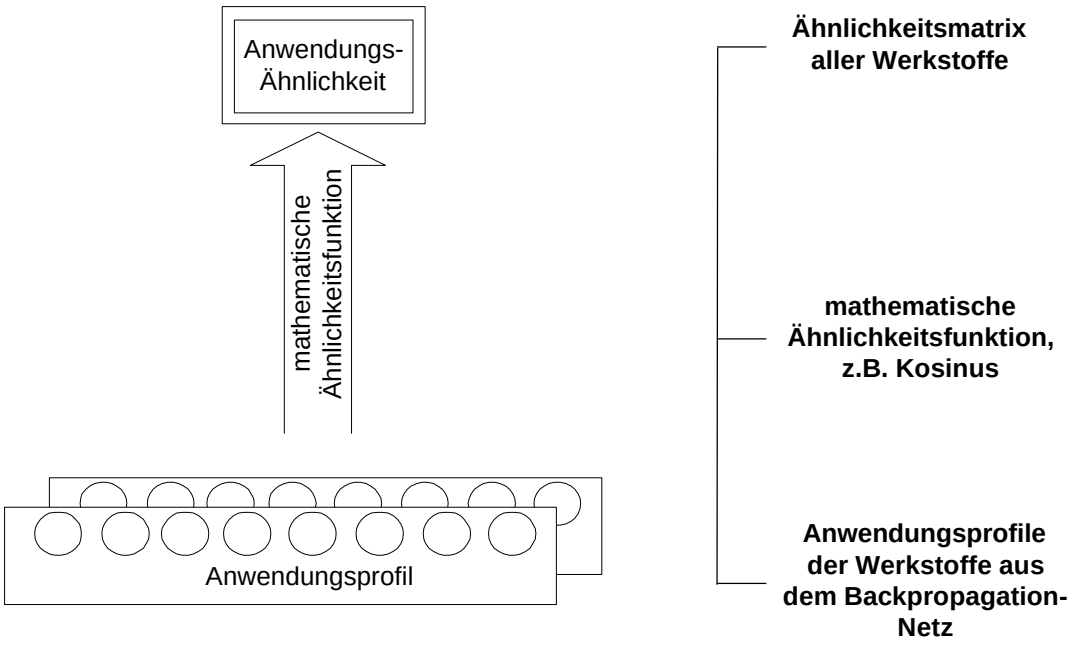


Abbildung 7-14: Erstellung der Trainingsdaten

Damit erhält das COSIMIR-Modell einen Input, aus dem der Output nicht mit statistischen Standard-Verfahren bestimmt werden kann. Wie bereits erwähnt, führen die Eigenschaften zu anderen Ähnlichkeiten unter den Werkstoffen als die Anwendungen. Demnach können Standardverfahren den Output aus den Input-Daten nicht bestimmen. Die Aufgabe besteht in der Abbildung von Basiseigenschaften auf eine kognitiv definierte Ähnlichkeit. Das Experiment testet, inwieweit COSIMIR diese in Abbildung 7-15 skizzierte Aufgabe lösen kann.

Die Werkstoffe wurden in Trainings- und Testmenge unterteilt. Das Netz wird mit Paaren von Werkstoffen aus der Trainingsmenge trainiert und lernt die gewünschte Abbildung. Die Paare unbekannter Werkstoffe in der Testmenge prüfen die Generalisierungsfähigkeit.

Gegenüber den Experimenten mit der Cranfield-Kollektion (cf. Abschnitt 7.1) ist diese Aufgabe prinzipiell schwieriger. Dort lernt das Netz wie bei fast allen Experimenten im Information Retrieval anhand einer Menge von Anfragen und Dokumente. Beim Test erhält das Modell eine neue Menge von Anfragen, während die Dokumente gleich bleiben. Information Retrieval Systeme werden in der Regel während des Tests bei jeder Ähnlichkeitsberechnung zwischen Anfrage und Dokument nur mit einem neuen Objekt konfrontiert. Im Falle der Werkstoffe ist die Rolle von Anfrage und Dokument nicht klar und die Entscheidung fiel für die schwierigere Variante. COSIMIR erhält damit beim Test zwei unbekannte Objekte.

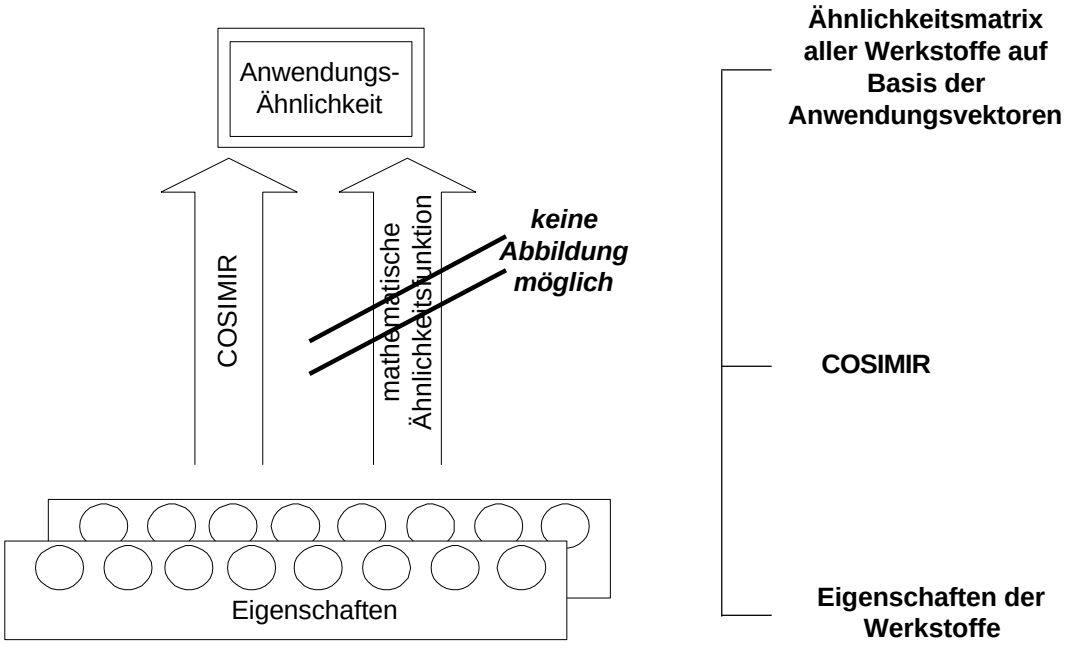


Abbildung 7-15: Aufgabe für COSIMIR

7.4.3 COSIMIR für heterogene Werkstoffdaten

COSIMIR kann auch für Heterogenitätsbehandlung im IR ohne expliziten Transformationsschritt benutzt werden (cf. Abschnitt 6.4.4). Dabei wird die Flexibilität des Modells ausgenutzt, das keine Annahmen über die Gleichförmigkeit des Eingangsdaten erfordert. Anstatt der Repräsentation zweier Objekte in einem Repräsentationsschema gehen als Input die Repräsentation eines Objektes in einem Schema und die Repräsentation eines anderen Objektes in zweiten Schema als Input in das Netz ein. Die Struktur der vorliegenden Werkstoffdaten erlaubt auch einen derartigen Test ohne die aufwendige Erstellung intellektueller Trainingsdaten.

Dabei versucht COSIMIR, die kognitiv motivierte Ähnlichkeit zwischen zwei Werkstoffen aus dem Anwendungsvektor eines Werkstoffs und aus dem Eigenschaftsvektor des anderen zu bestimmen. Das Experiment hat den Nachteil, dass ein Teil des Inputs, der Anwendungsvektor des einen Werkstoffs, bei der Berechnung des Outputs bereits verwendet wurde. Die Aussagekraft ist daher nicht so hoch wie die des vorherigen Experiments. Allerdings ist zur Zeit kein Datenbestand bekannt, der diese Anforderungen erfüllt. Die Erstellung der Daten für ein derartiges Experiment erfordert einen prohibitiv hohen intellektuellen Aufwand. Deshalb erscheint es sinnvoll, die Plausibilität des Modells mit dem beschriebenen Aufbau zu testen.

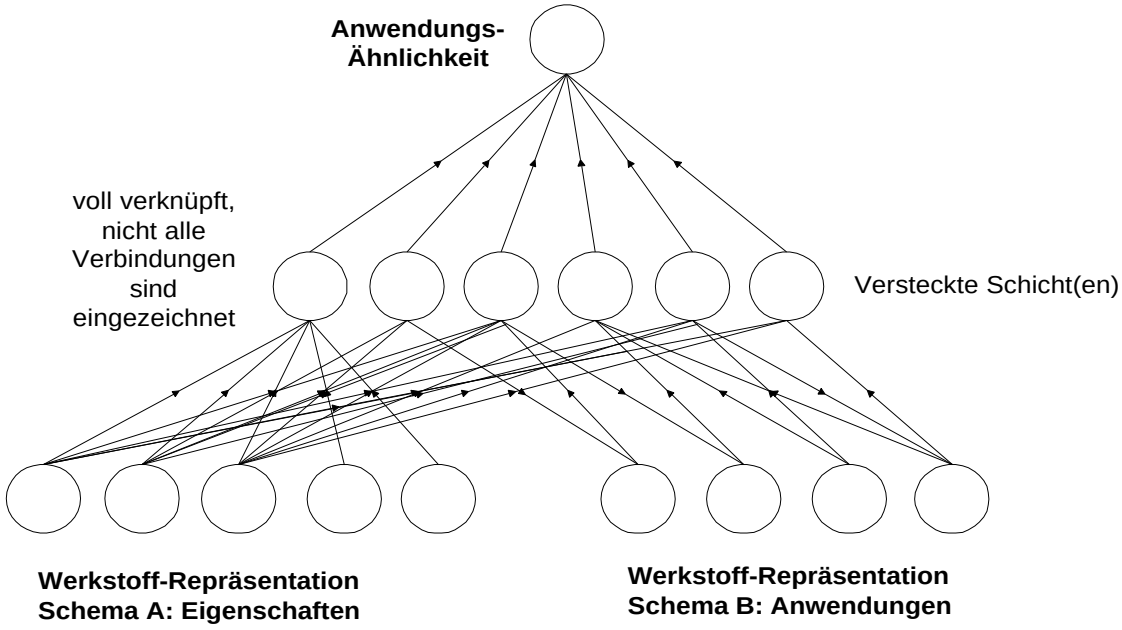


Abbildung 7-16: Ein COSIMIR-Netzwerk mit heterogenen Repräsentationen von Werkstoffen

7.4.4 Multi-Task-Learning

Zusätzliche Experimente untersuchten Multi-Task-Learning (cf. Abschnitt 3.5.4.3), wobei neben dem gewünschten Output ein zusätzlicher Output integriert wird. Das Netzwerk lernt zusätzlich diese Größe, bei der Bewertung der Qualität der Abbildung spielt sie aber keine Rolle (cf. Abbildung 7-17). Als zusätzliche Größen bieten sich für COSIMIR weitere Arten der Ähnlichkeit an. Dies können andere Definitionen von Ähnlichkeit sein, die sich aus dem Anwendungsfall ergeben (für den Werkstoffbereich cf. z.B. Mandl 1994) oder mit mathematischen Funktionen berechnete Ähnlichkeiten wie etwa der Dice-Koeffizient zusätzlich zum Kosinus. Im vorliegenden Fall dient der Kosinus als Annäherung der kognitiven Ähnlichkeit die Ähnlichkeit auf Basis der Anwendungen. Die Ähnlichkeit auf Basis der Eigenschaften schied als zusätzlicher Output aus. Da im Input zwei Eigenschafts-Vektoren anliegen, ergibt sich dann die Ähnlichkeit durch eine einfache Berechnung wie beim Pretest (cf. Abschnitt 7.4.2). Somit wurden andere mathematische Ähnlichkeitsmaße auf Basis der Anwendungen eingesetzt.

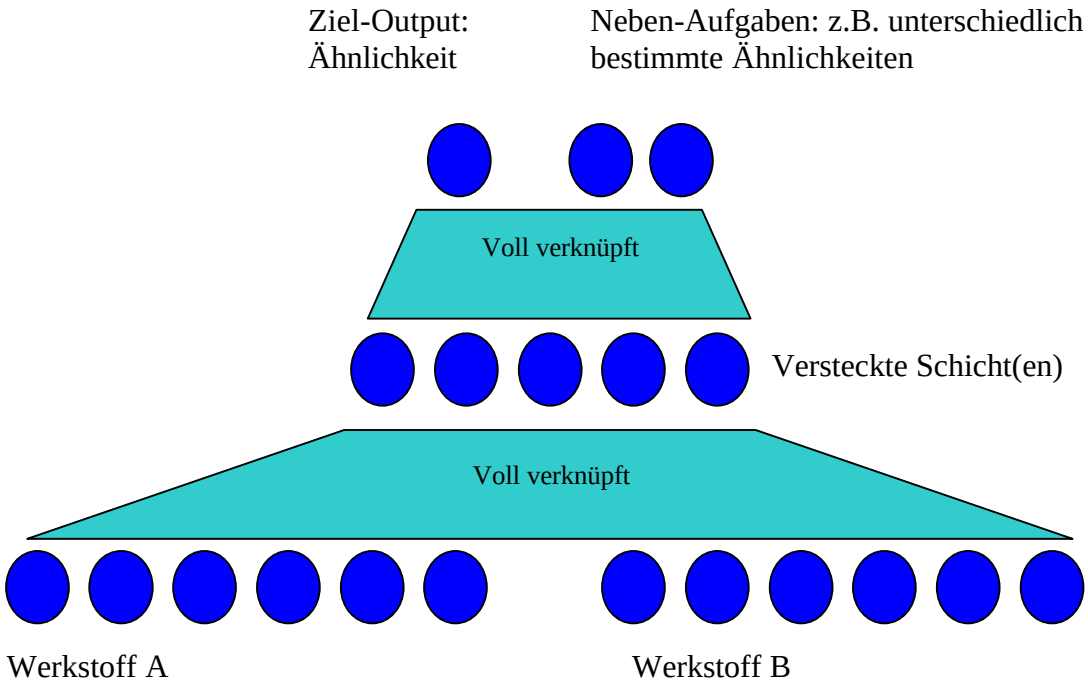


Abbildung 7-17: COSIMIR für Multi-Task-Learning

7.4.5 Vergleich von Rangfolgen

Für die Experimente mit den Werkstoff-Daten schieden die üblichen Qualitäts-Maße des Information Retrieval wie Recall und Precision aus, da nicht für jeden Werkstoff eine Menge dazu relevanter Werkstoffe vorliegt. Die Experimente führen vielmehr jeweils zu verschiedenen Ähnlichkeitsmatrizen, die verglichen werden müssen. Da COSIMIR für den Benutzer ein Ähnlichkeitswerkzeug darstellt, ist trotzdem die Sichtweise von einem Objekt (dem Ausgangswerkstoff oder der Anfrage) aus interessant. Deshalb bilden nicht die vollständigen Matrizen den Ausgangspunkt für den Vergleich, sondern die einzelnen Spalten. Diese beinhalten die Rangfolgen aller Werkstoffe ausgehend von einem Werkstoff, so wie eine Anfrage in einem Information Retrieval System zu einer Sortierung des Dokumentenbestandes führt. Für den Vergleich von Rangfolgen eignen sich Korrelationsmaße wie der bereits in Abschnitt 7.2.3 vorgestellten Spearmansche Rangkorrelationskoeffizient. Um die Resultate zu überprüfen, berücksichtigt dieses Experiment zusätzlich den Kendallschen Rangkorrelationskoeffizient:

Kendall: $$t = 1 - \frac{4\sum_{i=1}^{n} q_i}{n(n-1)}$$

(Hartung 1984:199, Grimmer/Mucha 1998:121)

Aus der Sicht des Information Retrieval sind beide Maße problematisch, da sie jeden Rang gleich werten. Für den Benutzer sind aber die ersten Ränge besonders wichtig. Da die Testmenge nur 23 Werkstoffe enthält, ist dieses Problem im vorliegenden Fall nicht schwerwiegend.

Aus den Rangfolgen aller Spalten wird ein Mittelwert gebildet, der das Gesamtergebnis für die Matrix darstellt.

7.4.6 Ergebnisse

Die Experimente mit Faktendaten wurden mit dem Simulationsprogramm DataEngine (cf. Abschnitt 3.6.2) unter Windows98 auf einem PC mit Pentium-I-Prozessor durchgeführt. Die Trainingszeiten für ein Netz bei ca. 200 Trainingsepochen liegen bei unter fünfzehn Minuten.

Die Ergebnisse basieren auf dem Spearmanschen Koeffizienten. Der Kendallsche Koeffizient liegt meist niedriger, verhält sich sonst aber ähnlich. Alle Werte beziehen sich auf die Testmenge (23 Werkstoffe), die Werte in der Trainingsmenge (46 Werkstoffe) liegen natürlich höher.

Beim Pretest, der Reimplementierung von Ähnlichkeitsmaßen, ist das COSIMIR-Modell durchaus erfolgreich. Der Kosinus wurde mit 79% erreicht, das Dice-Maß mit 82%. Untereinander korrelieren Kosinus und Dice mit 95%, Kosinus und Pearson-Maß 89% und Kosinus und Euklidisches Maß nur 65%. Damit liegen die Unterschiede zwischen dem Kosinus und dem von COSIMIR berechneten Kosinus im Bereich der Unterschiede zwischen anderen Ähnlichkeitsmaßen. Im eigentlichen Test lernt COSIMIR die kognitive Ähnlichkeit aufgrund der Eigenschaftsvektoren und ist dabei ähnlich erfolgreich. Sowohl für Kosinus als auch für Dice auf Basis der Anwendungsvektoren erreicht COSIMIR 70% Korrelation zwischen den Matrizen. Mit mathematischen Funktionen auf Basis der Eigenschaften berechneten Matrizen wiesen mit den auf Basis der Anwendungsprofile berechneten dagegen nur sehr geringe Korrelation auf (Kosinus: 37%, Dice 6%). Dies zeigt, dass die Anwendungen tatsächlich zu anderen Ähnlichkeiten führen als die Eigenschaften und damit die Grundannahme des Experiments richtig ist. Obwohl Standard-Verfahren die Anwendungsähnlichkeit also nicht aus den Eigenschaftsvektoren bestimmen können, lernt COSIMIR diese Abbildung anhand von Trainingdaten.

Im Test mit heterogenen Repräsentationen fielen die Ergebnisse erwartungsgemäß niedriger aus, da diese Aufgabe schwieriger ist. Der Kosinus wird 50% und der Dice-Koeffizient mit 64% erreicht. Diese Korrelation ist jedoch noch

deutlich höher als die zwischen der berechneten Anwendungs- und der berechneten Eigenschaftsähnlichkeit. Damit können auch die Werte für heterogene Repräsentationen als Erfolg gelten. Tabelle 7-4 fasst alle Ergebnisse zusammen.

Tabelle 7-4: Ergebnisse der Tests mit Werkstoffdaten

Experiment	**Erster Input**	**Zweiter Input**	**Output**	**Korrelation**
Mathematische Ähnlichkeitsfunktion implementieren	Anwedungs-Profile	Anwedungs-Profile	Kosinus aus Anwendungs-Profilen	79%
COSIMIR-Modell	Eigenschaften	Eigenschaften	Kosinus aus Anwendungs-Profilen	70%
COSIMIR-Modell für heterogene Repräsentationen	Anwedungs-Profile	Eigenschaften	Kosinus aus Anwendungs-Profilen	50%

Die Ergebnisse der Multi-Task-Experimente (cf. Tabelle 7-5) erlauben keine eindeutige Interpretation. Interessanterweise ergibt sich eine Verbesserung nur für die schwierigste Aufgabe, der Ähnlichkeitsberechnung aus heterogenen Vektoren. Dabei führt aber das Integrieren des Kosinus zu einer Verbesserung. Bei den anderen Aufgaben führte der Kosinus als zusätzlicher Lernfaktor zu einer geringfügigen Verschlechterung. Der dritte Output führt in allen Fällen zu einer Verschlechterung, wobei diese im schwierigsten Fall der heterogenen Repräsentationen am geringsten ausfiel.

Tabelle 7-5: Ergebnisse der Multi-Task-Experimente

Experiment	**Single-Task** (wie oben): Kosinus aus Anwendungs-Profilen	**Mit zweitem Output:** Dice aus Anwendungs-Profilen	**Mit drittem Output:** Pearson aus Anwendungs-Profilen
Mathematische Ähnlichkeitsfunktion implementieren	79%	78%	71%
COSIMIR-Modell	70%	66%	62%
COSIMIR-Modell für heterogene Repräsentationen	50%	62%	61%

7.5 Fazit: Experimente

Die Experimente sollten zeigen, ob Backpropagation-Netzwerke im Information Retrieval gute Ergebnisse erzielen und eventuell sogar zu besserer Qualität führen als andere Modelle. Das COSIMIR-Modell und das Transformations-Netzwerk basieren auf dem Backpropagation-Ansatz und bilden den Gegenstand der Experimente. Für beide Modelle reduziert Latent Semantic Indexing die Input-Daten.

Die Evaluierung des COSIMIR-Modells mit Textdaten erwies sich als schwierig, da nicht in ausreichendem Umfang Trainingsdaten zur Verfügung stehen. Die Vorteile von COSIMIR kommen erst zur Wirkung, wenn sehr viele Benutzerurteile für eine Kollektion bereitstehen. Die Experimente mit der Cranfield-Kollektion führten dementsprechend nicht zu befriedigenden Ergebnissen. Der intellektuelle Aufwand zur Erstellung solcher Daten ist hoch, so dass die Evaluierung von COSIMIR ein Versuchsaufbau mit Werkstoffdaten entwickelt, für die zwei Repräsentationen vorliegen. Aufgrund von Wissen über den Anwendungsbereich konnten die Experimente notwendige Benutzerurteile simulieren. Für diese Art der Experimente musste eine Bewertungsmethode entwickelt werden. Die positiven Ergebnisse zeigen, dass COSIMIR für Anwendungsbereiche mit kurzen Repräsentationsvektoren berücksichtigt werden sollte. Da die Performanz eines Information Retrieval

Verfahrens bei mehreren Korpora unterschiedlich sein kann, dürfen die Resultate nicht auf den Bereich Text-Retrieval übertragen werden, sie ermutigen jedoch zu weiteren Untersuchungen.

Die Evaluierung des Transformations-Netzwerks in Verbindung mit LSI konnte mit Standard-Methoden bewertet werden. Als Grundlage dienten reale, zweifach intellektuell indexierte Korpora. Dabei erreichte das Transformations-Netzwerk in den meisten Experimenten mindestens annähernd die Qualität eines statistischen Vergleichsverfahrens.

Bei einem Experiment mit sehr ähnlicher Qualität zeigten weitere Untersuchungen, dass die Ergebnisse zwar gleich gut, aber unterschiedlich waren. Zwei Analysen demonstrierten, dass die Schnittmengen zwischen Ergebnissen mit verschiedenen Verfahren sehr klein sind. Ähnliche Phänomene treten beim Standard Information Retrieval auf. Fusionsverfahren nutzen dies aus und erzeugen durch die Kombination der Ergebnisse verschiedener Verfahren ein besseres Gesamtergebnis. Die Experimente mit dem Transformations-Netzwerk und alternativen statistischen Ansätzen zeigen, dass bei Transformationen eine ähnliche Situation besteht. Die Ausgangssituation scheint also günstig für Fusionsverfahren, die auch für Transformationen getestet werden sollten.

Ansonsten zeigen die unterschiedlichen Ergebnisse des Transformations-Netzwerks im Vergleich zu einem statistischen Verfahren, dass stark von den Daten abhängt, welches Verfahren die besten Ergebnisse erbringt.

8 Fazit

Die vorliegende Arbeit befasst sich mit Information Retrieval Modellen auf der Basis neuronaler Netze und insbesondere des Backpropagation-Algorithmus. Ein Überblick über Grundlagen des Information Retrieval identifiziert die inhärente Vagheit von Informationsprozessen als besondere Herausforderung. Folgende Schwachpunkte bestehender Information Retrieval Modelle in den Bereichen Ähnlichkeitsberechnung, Adaptivität und Heterogenität lassen sich darauf zurückführen:

- Die Ähnlichkeit zwischen Anfrage und Dokument wird mit mathematischen Ähnlichkeitsfunktionen berechnet, welche die Eigenschaften der menschlichen Ähnlichkeitsbewertung nicht berücksichtigen.
- Information Retrieval Systemen passen sich nur in sehr geringem Umfang an die individuellen Eigenschaften und Bedürfnissen von Benutzern an.
- Umfassende Informationsangebote, in denen eine Anfrage auf verschiedene Datenquellen mit unterschiedlichen Objekten zugreift, führen zu starker semantischer Heterogenität, die mit der zunehmenden weltweiten Vernetzung weiter an Bedeutung gewinnt und meist unterschätzt wird.

In diesen Bereichen besteht Verbesserungspotenzial, um Informationssysteme toleranter für Benutzer und ihre Eigenschaften und Anforderungen zu gestalten. Die große Rolle der Vagheit im Information Retrieval führt zunehmend zur Einführung vager Methoden der Modellierung wie neuronalen Netzen.

Eine Einführung in die Grundlagen neuronaler Netze zeigt, dass die Stärke von neuronalen Netzen in ihrer Lernfähigkeit liegt. Ein umfassender state-of-the-art Bericht zu neuronalen Netzen im Information Retrieval behandelt vier Klassen von Systemen:

- Assoziative Speicher
- Kohonen-Netzwerke
- Spreading-Activation-Netzwerke
- Backpropagation-Modelle

Die beiden ersten Klassen werden zwar in Information Retrieval Systemen für große Datenmengen eingesetzt, für den Vergleich zwischen Anfrage und Dokument eignen sie sich aber aufgrund ihrer spezifischen Architektur nicht unmittelbar. Die meisten und erfolgreichsten Systeme beruhen auf dem einfa-

chen neuronalen Spreading-Activation-Netzwerk, das nur in geringem Umfang lernfähig ist. Das mächtige Backpropagation-Netzwerk ist nur im Transformations-Netzwerk realisiert, das bisher kaum empirisch untersucht wurde. Auch erfolgversprechende Ansätze zur Dimensionalitätsreduktion wie Latent Semantic Indexing wurden im Information Retrieval bisher nicht mit dem Backpropagation-Netzwerk kombiniert.

Das in dieser Arbeit vorgestellte COSIMIR-Modell ergibt sich aus der Analyse des state-of-the-art und bindet den Backpropagation-Algorithmus in den Kern eines Information Retrieval Systems ein. COSIMIR implementiert den Vergleich zwischen Anfrage und Dokument in einem Backpropagation-Netzwerk, das Repräsentationen beider Objekte als Input nutzt und das als Output die Ähnlichkeit oder System-Relevanz liefert. Die Trainingsdaten sind Benutzerurteile über Relevanz. COSIMIR lernt somit die Ähnlichkeitsfunktion unabhängig von einer mathematischen Modellierung nur aufgrund menschlicher Entscheidungen. Dadurch implementiert es die Adaptivität im Kern des Systems. Eine Erweiterung von COSIMIR bietet zudem die Möglichkeit zur direkten Behandlung von Heterogenität ohne einen Transformations-Schritt.

Im letzten Kapitel werden das COSIMIR-Modell und das bisher wenig untersuchte Transformations-Netzwerk mit realen Daten evaluiert. Für COSIMIR wurde eine Evaluierungsmethode entwickelt, um die aufwendige intellektuelle Erstellung eines Korpus zu vermeiden. Bei einem Anwendungsfall aus dem Bereich Fakten-Retrieval führt COSIMIR zu guten Ergebnissen.

Die Experimente mit dem Transformations-Netzwerk, das in Kombination mit Latent Semantic Indexing evaluiert wurde, führten zu interessanten Ergebnissen. Die Qualität der Transformation wurde mit einem statistischen Verfahren verglichen. Je nach Datengrundlage waren die Ergebnisse sehr unterschiedlich. In einem Fall war das Transformations-Netzwerk wesentlich besser, in einem anderen Experiment ergab sich annähernd gleiche Qualität. Welches Verfahren zum besten Ergebnis führt, hängt also stark von den Daten ab.

Damit zeigt sich eine Analogie zwischen Transformationen und Retrieval. Die TREC-Studien haben für das Retrieval gezeigt, dass sich die Qualität der besten aktuellen IR-Systeme nur unwesentlich unterscheidet, während sich ihre Ergebnisse teilweise stark unterscheiden. Das heißt, jedes System findet ungefähr gleich viele relevante Dokumente, jedes findet aber andere. Dies führte zur Entwicklung von Fusions-Ansätzen, die versuchen, die Ergebnisse einzelner Verfahren so zu kombinieren, dass ein besseres Gesamtergebnis entsteht. Das bedeutet, dass ein neuartiges Information Retrieval Verfahren wie COSIMIR nicht notwendigerweise alle anderen Verfahren übertreffen muss, sondern dass eine solche Entwicklung bereits gerechtfertigt ist, wenn

der neue Ansatz im Rahmen einer Fusion zu einem besseren Gesamtergebnis beiträgt.

Die Experimente mit dem Transformations-Netzwerk haben gezeigt, dass bei Transformationen ähnliche Phänomene auftreten können. Verschiedene Verfahren wie das Transformations-Netzwerk und ein statistischer Ansatz führen teilweise zu gleicher Qualität, die Schnittmenge der Ergebnisse ist aber relativ klein. Jedes Verfahren findet also andere Treffer, in diesem Fall Terme aus dem Ziel-Term-Raum. Als Konsequenz sollte auch die Heterogenitätsbehandlung verstärkt auf Fusions-Verfahren setzen. Ansonsten haben die Experimente gezeigt, dass die Qualität eines Transformations-Verfahrens stark mit von den Daten abhängt. Das *beste* Transformations-Verfahren muss also von Fall zu Fall gefunden werden.

Anhang

Literaturverzeichnis

Alle hier und in Fußnoten aufgeführten Internetquellen (URLs) wurden am 7.5.2001 verifiziert.

Aigrain, Philippe; Zhang, Hongjiang; Petkovic, Dragutin (1996): Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art-Review. In: Furht, Borko (Hrsg.): Multimedia Tools and Applications. An International Journal 3(3) 1996. S. 179-202.

Alonso-Betanzos, Amporo; Fontenla-Romero, Oscar; Guijarro-Berdiñas, Bertha; Principe, Juan Carlos (1999): A Multi-Resolution Principal Component Analysis Neural Network for the Detection of Foetal Heart Rate Patterns. In: Zimmermann 1999.

Amir, Amihood; Feldman, Ronen; Kashi, Reuven (1997): A New and Versatile Method for Association Generation. In: Information Systems 22(6/7). S. 333-347.

Amstutz, Hans; Holländer-Thönssen, Barbara (1991): Elektronische Ablage und Archivierung auf der Basis eines Database Management Information Retrieval Systems: Die Bedürfnisse - Das Angebot - Die Realität. In: Fuhr, Norbert (Hrsg.): Information Retrieval. GI/GMD-Workshop, Darmstadt, 23.-24.6.1991. S. 78-93.

Anahory, Sam; Murray, Dennis (1997): Data Warehouse: Planung, Implementierung und Administration. Bonn et al.

Apté, Chidanand; Damerau, Fred; Weiss, Sholom (1994): Towards Language Independent Automated Learning of Text Categorization. In: Bartell et al. 1994. S. 23-30

Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier (Hrsg.)(1999): Modern Information Retrieval.

Banvilhon, F.; Richard, P. (1984): Managing Texts and Facts in a Mixed Database Environment. In: Gardarin, Gelenbe (Hrsg.): New Applications of Databases. New York: Academic Press.

Barnden, John (1994): On the Connectionist Implementation of Analogy and Working Memory Matching. In: Barnden, John; Holyoak, Keith (Hrsg.): Advances in Connectionist and Neural Computation Theory. vol 3: Analogy, Metaphor, and Reminding. Norwood, NJ. S. 327-374.

Bartell, Brian; Cottrell, Garrison; Belew, Richard (1994): Automatic Combination of Multiple Retrieval Systems. In: Croft, Bruce (Hrsg.): Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ´94). Dublin, 3.-6.7.1994. London et al.

Bartlmae, Kai (1998): A Countryrisk Assessment Framework Using Neural Multitask Learning. In: Zimmermann 1998. S. 1019-1023.

Bayraktar, Osman; Womser-Hacker Christa (1998) Qualitätsbewertung von Information-Retrieval-Systemen: Sind Synergieeffekte durch die Koordination verschiedener Bewertungsmethoden möglich? In: Zimmermann/Schramm 1998. S. 427-437.

Beasley, David; Bull, David; Martin, Ralph (1993a): An Overview of Genetic Algorithms: Part 1, Fundamentals. In: University Computing 15 (2). S. 58-69.

Beasley, David; Bull, David; Martin, Ralph (1993b): An Overview of Genetic Algorithms: Part 2, Research Topics. In: University Computing 15 (4). S.170-181.

Bekavac, Bernard (1999): Suche und Orientierung im WWW. Verbesserung bisheriger Verfahren durch Einbindung hypertextspezifischer Informationen. Konstanz.

Belew, Richard (1986): Adaptive Information Retrieval: Machine Learning in Associative Networks. PhD Dissertation. University of Michigan, Ann Harbor.

Belew, Richard (1989): Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents. In: Belkin/van Rijsbergen 1989. S. 11-20.

Belkin, Nicholas (1993): Interaction with Texts: Information Retrieval as Information Seeking Behaviour. In: Knorz et al. 1993. S. 55-66.

Belkin, Nicholas; Ingwersen, Peter; Pejtersen, Annelise (Hrsg.)(1992): Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92). Kopenhagen, Dänemark. 21-24.6.1992. London et al.

Belkin, Nicholas; Rijsbergen, C.J. van (Hrsg.)(1989): Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '89) Cambridge, MA, USA 25.-28.6.89. London et al.

Bentz, Hans-Joachim; Hagström, Michael; Palm, Guenther (1998): Information Storage and Effective Data Retrieval in Sparse Matrices. In: Neural Networks 2 (4). S. 289-293.

Berry, Michael (1992): Large Scale Sparse Singular Value Computations. In: International Journal of Supercomputer Applications. S.13-49.

Berry, Michael; Do, Theresa; O'Brien, Gavin; Krishna, Vijay; Varadhan, Sowmini (1993): SVDPACKC (Version 1.0) User's Guide. Arbeitspapier. Computer Science Department. University of Tennessee in Knoxville, USA.
http://www.netlib.org/svdpack/svdpackc.tgz

Berry, Michael; Dumais, Susan; Letsche, Todd (1995): Computational Methods for Intelligent Information Access. In: Proceedings of ACM Supercomputing '95. San Diego, CA. S. 1-38.

Biebricher, B.; Fuhr, Norbert; Lustig, G.; Schwantner, M.; Knorz, Gerhard (1988): The Automatic Indexing System AIR/PHYS - from Research to Application. In: Chiaramella, Yves (Hrsg.): Proceedings of the 11th International Conference on Research & Development in Information Retrieval. ACM. New York. S. 333-342.

Bigus, Joseph (1996): Data Mining with Neural Networks. Solving Buisness Problems from Application Development to Decision Support. New York et al.

Bimbo, Alberto del (1999): Visual Information Retrieval. San Francisco.

Bollmann, Peter; Konrad, Erhard (1979): Mathematische Modelle von Information Retrieval Systemen. In: Kuhlen, Rainer (1979): Datenbasen, Datenbanken, Netzwerke. München et al. Bd. II.

Bookstein, Abraham; Chiaramella, Yves; Salton, Gerard; Raghavan, Vijay V. (Hrsg.)(1991): Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '91). Chicago, IL, USA 13.-16.10.91. New York.

Bordogan, Gloria; Pasi, Gabriella; Petrosino, Alfredo (1996): Relevance Feedback Based on a Neural Network. In: Zimmermann 1996. S. 846-850.

Bosc, Patrick; Pivert, Olivier (1991): About Equivalence in SQLf, a Relational Language Supporting Imprecise Querying. In: Terano, Toshiro (Hrsg.): Fuzzy engineering toward human friendly systems: Proceedings of the International Fuzzy Engineering Symposium '91, IFES '91, Yokohama, Japan, 13.-15.11.91. Tokio.

Boughanem, M.; Chrisment, C.; Soulé-Dupuy, C. (1999): Query Modification based on Relevance Back-Propagation in an ad hoc Environment. In: Information Processing and Management 35. S. 121-139.

Boughanem, M.; Soulé-Dupuy, C. (1994): Query Expansion and Neural Network. In: Intelligent Multimedia Information Retrieval Systems and Management. Proceedings of the RIAO 94 (Recherche d'Information assistée par Ordinateur). Rockfeller University. New York. S. 519-532.

Boughanem, M.; Soulé-Dupuy, C. (1997): MercureO2: adhoc and routing tasks. In: Voorhees/Harman 1997.

Boughanem, M.; Soulé-Dupuy, C. (1998): Mercure at trec6. In: Voorhees/Harman 1998

Boughanem, M.; Dkaki, T.; Mothe, J; Soulé-Dupuy, C. (1999): Mercure at trec7. In: Voorhees/Harman 1999

Boyd, Richard; Driscoll, James; Syu, Inien (1994): Inorporating Semantics within a Connectionist Model and a Vector Processing Model. In: Harman 1994. S. 291-302.

Braschler, M.; Krause, Jürgen; Peters, Carol; Schäuble, Peter (1999): Cross-Language Information Retrieval (CLIR) Track Overview. In: Voorhees/Harman 1999. S. 25-32.

Breitkopf, Günter; Over, Helmut; Rösner, Helmut (1997): Werkstoffproblematik und Anwendungsdomänen. In: Krause/Womser-Hacker 1997. S. 19-42.

Buckland, Michael; Gey, Fredic; et al. (1999): Mapping Entry Vocabulary to Unfamiliar Metadata Vocabularies. In: D-Lib Magazine 5(1). http://www.dlib.org/dlib/january99/buckland/01buckland.html

Buckley, Chris (1998): TREC 6 High-Precision Track. In: Harman/Voorhees 1998. S. 69-71.

Buder, Marianne; Rehfeld, Werner; Seeger, Thomas; Strauch, Dietmar (Hrsg.)(1997): Grundlagen der praktischen Information und Dokumentation: ein Handbuch zur Einführung in die fachliche Informationsarbeit

Bullinger, Hans-Jörg; Ziegler, Jürgen (Hrsg.)(1999): Human-Computer Interaction: Communication, Cooperation and Application Design. Proceedings of the HCI International´99 (8th International Conference on Human-Computer Interaction), München, 22-27.8.1999.

Burkart, Margarete (1997): Thesaurus. In: Buder et al. 1997. S. 160-179.

Caid, William R.; Dumais, Susan; Gallant, Stephen (1995): Learned Vector-Space Models for Document Retrieval. In: Information Processing & Management 31(3). S. 419-429.

Caruana, Rich (1994): Learning Many Related Tasks at the Same Time With Backpropagation. In: Advances in Neural Information Processing Systems 7 (Proc. of NIPS ´94) S. 657-664. http://www.cs.cmu.edu/afs/cs.cmu.edu/user/caruana/pub/papers/nips94.ps

Caruana, Rich (1997): Multitask Learning. In: Machine Learning 28. S. 41-75. http://www.cs.cmu.edu/afs/cs.cmu.edu/user/caruana/pub/papers/mlj97.ps

Caudell, Thomas; Smith, Scott; Escobedo, Richard; Anderson, Michael (1994): NIRS: Large Scale ART-1 Neural Architectures for Engineering Design Retrieval. In: Neural Networks 7(9). S. 1339-1350.

Chen, Chaomei (1999): Information Visualization and Virtual Environments. London et al.

Chen, Hsinchun (1994): A Machine Learning Approach to Document Retrieval: An Overview and an Experiment. Technical Report. MIS Department. University of Arizona, Tucson.

Chen, Hsinchun (1995): Machine Learning for Information Rerieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. In: Journal of the American Society for Information Science. JASIS 46(3). S. 194-216.

Chen, Hsinchun (1998): Introduction: Trailblazing Path to Semantic Interoperability. In: Journal of the American Society for Information Science. JASIS 49(7). S. 579-581.

Chen, Hsinchun; Martinez, Joanne; Kirchhoff, Amy; Ng, Tobun; Schatz, Bruce (1998): Alleviating Search Uncertainty through Concept Associations: Automatic Indexing, Co-Occurrence Analysis, and Parallel Computing. In: Journal of the American Society for Information Science. JASIS 49(3). S. 206-216.

Chen, Hsinchun; Martinez, Joanne; Ng, Tobun; Schatz, Bruce (1996): A Concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: An Experiment on the Worm Community System. In: Journal of the American Society for Information Science. JASIS 48(1). S. 17-31.

Chen, Hsinchun; Schuffels, Chris; Orwig, Richard (1996): Internet Categorization and Search: A Self-Organizing Approach. In: Journal of Visual Communication and Image Representation. 7(1). S. 88-101.

Chen, Qiyang; Norico, Anthony (1992): Modelling Users with Neural Architectures. In: International Joint Conference on Neural Networks (IJCNN). Baltimore, 7.-11.06.1992. New York, NY. 1992. S. 547-52. vol. 1 of 4.

Chung, Yi-Ming; Pottenger, Wiliam; Schatz, Bruce (1998): Automatic Subject Indexing Using an Associative Neural Network. In: Proceedings of the Third ACM Conference on Digital Libraries. 23.–26.6.1998, Pittsburgh, PA, USA. S. 59-68

Cichocki, Andrzej; Unbehauen, Rolf (1993): Neural Networks for Optimization and Signal Processing. Stuttgart et al.

Clauß, G.; Ebner, H. (1979): Grundlagen der Statistik. Frankfurt a.M.

Cochet, Yves; Paget, Gerard (1988): ZZENN: ZIG ZAG Epigenatic Neural Networks and Their Use in Information Systems. In: Personnaz, L.; Dreyfus, G. (Hrsg.): Neural Networks: From Models To Applications. Proceedings of Neuro`88. S. 663-672.

Cooper, William (1988): Getting Beyond Boole. In: Information Processing and Management 24 (3). S. 243-248.

Cooper, William (1991): Some Inconsistencies and Misnomers in Probabilistic Information Retrieval. In: Bookstein et al 1991. S. 57-61.

Cortez, Edwin; Park, Sang; Kim, Seonghee (1995): The Hybrid Application of an Inductive Learning Method and a Neural Network for Intelligent Information Retrieval. In: Information Processing and Management 31(6). S. 789-813.

Creput, Jean-Charles; Caron, Armand (1997): An Information Retrieval System Using a New Neural Network Model. In: Cybernetica XL(2). S. 127-139.

Crestani Fabio (1993): Learning Strategies for an Adaptive Information Retrieval System using Neural Networks. In: Proceedings of the IEEE International Conference on Neural Networks. San Francisco, California, USA. S. 244-249. http://www.cs.strath.ac.uk/~fabioc/papers/93-icnn.pdf

Crestani, Fabio (1993a): An adaptive information retrieval system based on neural networks. In: Mira, José; Cabestany, Joan; Prieto, Alberto (Hrsg.): New trends in Neural Computation. International Workshop on Artificial Neural Networks. IWANN '93 Proceedings. Berlin. http://www.cs.strath.ac.uk/~fabioc/papers/93-iwann.pdf

Crestani, Fabio (1994): Comparing Neural and Probabilistic Relevance Feedback in an Interactive Information Retrieval System. In: Proceedings of the IEEE International Conference on Neural Networks. Orlando, Florida, USA. Jun 1994. http://www.cs.strath.ac.uk/~fabioc/papers/94-icnn.pdf

Crestani, Fabio (1994a): Domain Knowledge Acquisition for Information Retrieval Using Neural Networks. In: The new review of applied expert systems. S. 101-115. http://www.cs.strath.ac.uk/~fabioc/papers/94-ijaes.pdf

Crestani, Fabio (1995): Implementation and Evalution of a Relevance Feedback Device Based on Neural Networks. In: Mira, José; Cabestany, Joan (Hrsg.): From Natural to Artificial neural Computation: International Workshop on Artificial Neural Networks. Spain. S. 1-8.

Crestani, Fabio (1997): Application of Spreading Activation Techniques in Information Retrieval. In: Artificial Intelligence Review 11 (6). S. 453-482. http://www.cs.strath.ac.uk/~fabioc/papers/97-air.pdf

Crestiani, Fabio (1997a): Retrieving Documents by Constrained Spreading Activation on Automatically Constructed Hypertexts. In: Zimmermann 1997. S. 1210-1214.

Crestani, Fabio; Rijsbergen, Keith van (1997): A Model for Adaptive Information Retrieval. In: Journal of Intelligent Information Systems 8 (1). S. 29-56. http://www.cs.strath.ac.uk/~fabioc/papers/97-joiis.pdf

Croft, Bruce; Moffat, Alistair; Rijsbergen, Keith van; Wilkinson, Ross; Zobel, Justin (1998): Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ´98). Melbourne 24.-28.8.1998. New York.

Deerwester, Scott; Dumais, Susan T.; Harshman, Richard (1990): Indexing by Latent Semantic Analysis. In: Journal of The American Society For Information Science (JASIS) 41(6). S. 391-407.

Delgado, Miguel; Sánchez, Daniel; Vila, Maria-Amparo (1999): Fuzzy Quantified Dependencies in Relational Databases. In: Zimmermann 1999.

Dorffner, Georg (1991). Konnektionismus. Von neuronalen Netzwerken zu einer natürlichen KI. Stuttgart.

Doszkocs, T.E.; Reggia, J.; Lin, Xia (1990): Connectionist Models and Information Retrieval. In: Annual Review of Information Science and Technology (ARIST) 25. S. 209-260.

Düsterhöft, Antje (1999): GI-FG-Treffen "Datenbanksysteme, Information Retrieval und Verteilte Künstliche Intelligenz": Workshop ADI'99 (Agenten, Datenbanken und Information Retrieval). http://wwwdb.informatik.uni-rostock.de/adi99/

Dumais, Susan (1994): Latent Semantic Idexing (LSI) and TREC-2. In: Harman 1994. S. 105-115.

Dumais, Susan; Letsche, Todd; Littman, Michael; Landauer, Thomas (1997): Automatic Cross-Language Retrieval Using Latent Semantic Indexing. In: Hull/Oard 1997. S. 15-21.

Eibl, Maximilian (2000): Visualisierung im Dokument Retrieval: Theoretische und praktische Zusammenführung von Softwareergonomie und Graphik Design. Dissertation. Univerität Koblenz-Landau.

Escobedo, Richard; Smith, Scott; Caudell, Thomas (1993): A Neural Information Retrieval System. In: International Journal of Advanced Manufacturing Technology 8(4). S. 269-274.

Fachgruppe IR (1996): Fachgruppe Information Retrieval. http://ls6-www.informatik.uni-dortmund.de/ir/fgir/mitgliedschaft/brochure2.html

Fayyad, Usama (1997): Editorial. In: Data Mining and Knowledge Discovery 1(1). S. 5-10.

Ferber, Reginald (1997): Automated Indexing with Thesaurus Descriptors: A Coocurrence Base Approach to Multilingual Retrieval. In: Peters, Carol; Thanos, Constantino (Hrsg.): Research and Advanced Technology for Digital Libraries. 1st European Conf. ECDL'97. Pisa, 1.-3.9.97. Berlin et al. S. 233-252.

Fox, Edward (Hrsg.)(1995): Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 95). Seattle, USA 9.-13.7.95. New York.

Frei, Hans-Peter; Harman, Donna; Schäuble, Peter; Wilkinson, Ross (Hrsg.)(1996): Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96). Zürich, 18.-22.8.96. New York.

Fuhr, Norbert (1992): Integration of Probabilistic Fact and Text Retrieval. In: Belkin et al. 1992. S. 211-222.

Fuhr, Norbert (1995): Modelling Hypermedia Retrieval in Datalog. In: Kuhlen/Rittberger 1995. S. 163-174.

Fuhr, Norbert (1999): Towards Data Abstraction in Networked Information Retrieval Systems. In: Information Processing and Management 35. S. 101-119.

Fuhr, Norbert (1999a): Information Retrieval in Digitalen Bibliotheken. In: Schmidt 1999. S. 93-102.

Fuhr, Norbert; Rittberger, Marc; Womser-Hacker, Christa (1998): Information Retrieval. Materialien zur Herbstschule. Bonn: Gesellschaft für Informatik.

Gabler, Siegfried (1997): Datenfusion. In: ZUMA-Nachrichten 40. Zentrum für Umfragen, Methoden und Analysen. S. 81-92.

Gallant, Stephen; Caid, William; Carleton, Joel; Hecht-Nielsen, Robert; Qing, Kent; Sudback, David (1993): HNC's MatchPlus System. In: Harman 1993. S. 107-111.

Gallant, Stephen; Caid, William; Carleton, Joel; Gutschow, Todd; Hecht-Nielsen, Robert; Qing, Kent; Sudback, David (1994): Feedback and Mixing Experiments with Match Plus. In: Harman 1994. S. 101-104.

Gauch, Susan; Wang, Jianying (1997): Corpus Analysis for TREC 5 Query Expansion. In: Voorhees/Harman 1997.

Gloor, Peter (1997): Elements of Hypermedia Design: Techniques for Navigation & Visualization in Cyberspace. Birkhäuser: Boston.

Gövert, Norbert (1996): Information Retrieval in vernetzten heterogenen Datenbanken. In: Krause Jürgen; Herfurth, Matthias; Marx, Jutta (Hrsg.): Herausforderungen an die Informationswirtschaft: Informationsverdichtung, Informationsbewertung und Datenvisualisierung. Proceedings des 5. Internationalen Symposiums für Informationswissenschaft (ISI 96). Berlin 17.-19.10.96. Konstanz.

Gövert, Norbert (1997): Evaluierung eines entscheidungstheoretischen Modells zur Datenbankselektion. In: Fuhr, Norbert; Dittrich, Gisbert; Tochtermann, Klaus (Hrsg.): Hypertext - Information Retrieval - Multimedia (HIM). Theorien, Modelle und Implementierungen integrierter elektronischer Informationssysteme. Konstanz. S. 135-146.

Gordon, Michael D.; Dumais, Susan (1998): Using Latent Semantic Indexing for Literature Based Discovery. In: Journal of the American Society for Information Science. JASIS. 49(8). S. 674-685.

Graupe, D.; Kordylewski, H. (1998): A Large Memory Storage and Retrieval Neural Network for Adaptive Retrieval and Diagnosis. In: International Journal of Software Engineering and Knowledge Engineering 8(1). S. 115-138.

Grauel, A.; Ludwig, L.; Renners, I. (1999): Comparison of Neuro/Fuzzy Methods for Toxity Evaluation. In: Zimmermann 1999.

Grimmer, Udo; Mucha, Hans-Joachim (1998): Datensegmentierung mittels Clusteranalyse. In: Nakhaeizadeh 1998. S. 109-141.

Gu, Junzhong; Thiel, Ulrich; Zhao, Jian (1993): Efficient Retrieval of Complex Objects: Query Processing in a Hybrid DB and IR System. In: Knorz et al. 1993. S. 67-81.

Gupta, Amarnath; Jain, Ramesh (1997): Visual Information Retrieval. In: Communications of the ACM. 40(5). S. 71-79.

Haenelt, Karin (1996): Das KONTEXT-Modell und die Konzeption der textmodellbasierten Verarbeitung natürlichsprachiger Texte. Arbeitspapiere der GMD 1009. Juli 1996. Darmstadt.

Hagström, Michael (1996): Textrecherche in großen Datenmengen auf der Basis spärlich codierter Assoziativmatrizen. Dissertation. Universität Hildesheim.

Harman, Donna (1992): Relevance feedback revisited. In: Belkin et al. 1992. S. 1-10.

Harman, Donna (Hrsg.)(1993): The First Text Retrieval Conference (TREC-1). NIST Special Publication 500-207. National Institute of Standards and Technology. Gaithersburg, Maryland, 4.-6.11.1992. http://trec.nist.gov/pubs/trec1/t1_proceedings.html

Harman, Donna (Hrsg.)(1994): The Second Text REtrieval Conference (TREC-2). Publication 500-215. National Institute of Standards and Technology. Gaithersburg, Maryland,31.8.-2.9.1993. http://trec.nist.gov/pubs/trec2/t2_proceedings.html

Harman, Donna (Hrsg.)(1995): The Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-225. National Institute of Standards and Technology. Gaithersburg, Maryland, 2.-4.11.1994. http://trec.nist.gov/pubs/trec3/t3_proceedings.html

Harman, Donna (Hrsg.)(1996): The Fourth Text Retrieval Conference (TREC-4). NIST Special Publication 500-236. National Institute of Standards and Technology. Gaithersburg, Maryland, 1.-3.11.1995. http://trec.nist.gov/pubs/trec4/t4_proceedings.html

Hartung, Joachim (1984): Lehr- und Handbuch der angewandten Statistik. München, Wien.

Hawking, Daniel; Craswell, Nick; Thistlewaite, Paul; Harman, Donna (1999): Result and Challenges in Web Search Evaluation. In: Proceedings of the 8th WWW Conference. Toronto. S. 243-252. http://www8.org/w8-papers/2c-search-discover/results/results.html

Heitland, Michael (1994): Einsatz der SpaCAM-Technik für ausgewählte Grundaufgaben der Informatik. Dissertation. Universität Hildesheim.

Heuer, Andreas; Saake, Gunter (1997): Datenbanken : Konzepte und Sprachen. Bonn et al.

Honkela, Timo; Kaski, Samuel; Lagus, Krista; Kohonen, Teuvo (1997): WEBSOM-Self-Organizing Maps of Document Collections. In Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6, Helsinki University of Technology, Neural Research Centre, Espoo, Finland. S. 310-315.

Hoogeveen, Martijn; van der Meer, Kees; Sol, Henk (1992): The Integration of Information Retrieval and Database Management Facilities in Support of Multimedia Information Work. In: Zmmermann, Harald; Luckhardt, Heinz-Dirk; Schulz, Angelika (Hrsg.): Mensch und Maschine – Informationelle Schnittstellen der Kommunikation. Proceedings 3. Int. Symposium für Informationswissenschaft. (ISI ´92). Konstanz. S. 260-274.

Hui, Siu; Goh, Angela (1996): Incorporating Fuzzy Logic with Neural Networks for Document Retrieval. In: Engineering Applications of Artificial Intelligence 9(5). S. 551-560.

Hull, David; Grefenstette, Gregory (1996): Querying Across Languages: A Dictionary-Based Approach to Multilingual Information Retrieval. In: Frei et al. 1996. S. 49-57.

Hull, David; Oard, Doug (1997): Cross-Language Text and Speech Retrieval Papers from the 1997 AAAI Spring Symposium. Technical Report SS-97-05.

Informationzentrum Sozialwissenschaften (1997): Thesaurus Sozialwissenschaften. Deutsch-Englisch. Bearbeitet von Hannelore Schott. Bonn.

Ingwersen, Peter (1992): Information Retrieval Interaction. London.

Jennings, Andrew; Higuchi, Hideyuki (1992): A Personal News Service Based on a User Model Neural Network. In: IEICE-Transactions on Information and Systems (March 1992) no. 2. S. 198-209.

Jones, William; Furnas, George (1987): Pictures of Relevance: A Geometric Analysis of Similarity Measures. In: Journal of the American Society for Information Science. JASIS 38 (6). S. 420-442.

Jong, E. de; Keuken, H.; Pol, E. van der; Dekker, E. den; Kerckhoffs, E. J. (1996): Exergy Analysis of Industrial Processes Using AI Techniques. In: Computers and Chemical Engineering 20. S. S1631-S1636.

Kaski, Samuel (1998): Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. In Proceedings of IJCNN'98, International Joint Conference on Neural Networks, Piscataway, NJ. vol. 1. S. 413-418.

Kaski, Samuel; Lagus, Krista; Honkela, Timo; Kohonen, Teuvo (1998): Statistical Aspects of the WEBSOM System in Organizing Document Collections. In: Computing Science and Statistics 29. S. 281-290.

Kinnebrock, W. (1992): Neuronale Netze. Grundlagen, Anwendungen, Beispiele. München; Wien.

Kluck, Michael (1998): German indexing and retrieval test database: some results of the pre-test. In: Discovering new worlds of IR. Proceedings of the IRSG ´98. Grenoble, France. 25-27.3.1998.

Kluck, Michael; Krause, Jürgen; Müller, Matthias; in Kooperation mit Schmiede, R.; Wenzel, H.; Winkler, S.; Meier, W. (2000): Virtuelle Fachbibliothek Sozialwissenschaften: IZ-Arbeitsbericht 19, Informationszentrum Sozialwissenschaften, Bonn. http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Virtuelle

Knorz, Gerhard (1997): Testverfahren für intelligente Indexierungs- und Retrievalsysteme anhand deutschsprachiger sozialwissenschaftlicher Fachinformation (GIRT), Bericht über einen Workshop am IZ Sozialwissenschaften, Bonn, 12.9.97. In: LDV-Forum 14(2). S. 43-56.

Knorz, Gerhard; Krause, Jürgen, Womser-Hacker, Christa (1993)(Hrsg.): Information Retrieval '93. Von der Modellierung zur Anwendung. Proceedings der 1. Tagung Information Retrieval '93, Regensburg, Oktober 1993. Konstanz.

Kohonen, Teuvo (1984). Self-Organization and Associative Memory. Berlin et al.

Kohonen, Teuvo (1997): Exploration of very large databases by self-organizing maps. In Proceedings of ICNN'97, International Conference on Neural Networks. IEEE Service Center, Piscataway, NJ. S. PL1-PL6.

Kohonen, Teuvo (1997[2]a): Self-Organizing Maps. Springer: Berlin et al.

Kohonen, Teuvo (1998): Self-organization of Very Large Document Collections: State of the art. In Niklasson, L.; Bodén, M.; Ziemke, T. (Hrsg.): Proceedings of ICANN ´98, 8th Intl Conference on Artificial Neural Networks, Springer: London. vol. 1, S. 65-74.

Kramer, Ralf; Nikolai, Ralf; Habeck, Corinna (1997): Thesaurus Federations: Loosely Integrated Thesauri for Document Retrieval in Networks Based on Internet Technologies. In: International Journal on Digital Libraries. S. 122-131.

Krause, Jürgen (1996): Principles of Content Analysis for Information Retrieval Systems: An Overview. In: Zuell, C.; Harkness, J.; Hoffmeyer-Zlotnik, J. (Hrsg.): Text Analysis and Computer. ZUMA-Nachrichten Spezial. Mai. Mannheim. S.77-100. ftp://ftp.zuma-mannheim.de/pub/zuma/zuma-nachrichten_spezial/znspezial1.pdf

Krause, Jürgen (1996a): Informationserschließung und -bereitstellung zwischen Deregulation, Kommerzialisierung und weltweiter Vernetzung. (‚Schalenmodell'). IZ-Arbeitsbericht 6, Informationszentrum Sozialwissenschaften, Bonn. http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Informationserschließung

Krause, Jürgen (1997): Das WOB-Modell. In: Krause/Womser-Hacker 1997. S. 59-88.

Krause, Jürgen (1998): Innovative Current Research Information Systems in the Information Society. In: CRIS ´98 Current Research Information Systems. Luxemburg, 12-14.3.1998. ftp://ftp.cordis.lu/pub/cybercafe/docs/krause.zip

Krause, Jürgen; Mandl, Thomas; Stempfhuber, Maximilian (1997): Text-Fakten-Integration in ELVIRA. IZ-Arbeitsbericht 12, IZ Sozialwissenschaften, Bonn. http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Text

Krause, Jürgen; Mandl, Thomas; Stempfhuber, Maximilian (1998): Design der Benutzungsoberfläche des ZVEI-Verbandsinformationssystems ELVIRA. In: Scheinost et al. 1998. S. 39-65.

Krause, Jürgen; Mutschke, Peter (1999): Indexierung und Fulcrum-Evaluierung. IZ-Arbeitsbericht 17, IZ Sozialwissenschaften, Bonn. http://www.bonn.iz-soz.de/publications/series/working-papers/index.htm#Relationale

Krause, Jürgen; Schaefer, André (1998): Textrecherche-Oberfläche für ELVIRA II. ELVIRA-Arbeitsbericht 16, IZ Sozialwissenschaften, Bonn.

Krause, Jürgen; Womser-Hacker, Christa (1997)(Hrsg.): Vages Information Retrieval und graphische Benutzungsoberflächen - Beispiel Werkstoffinformation. Konstanz.

Kuhlen, Rainer (1991): Hypertext: ein nicht-lineares Medium zwischen Buch und Wissensbank. Berlin.

Kuhlen, Rainer (1999): Die Konsequenzen von Informationsassistenten: Was bedeutet informationelle Autonomie oder wie kann Vertrauen in elektronische Dienste in offenen Informationsmärkten gesichert werden? Frankfurt a.M.

Kuhlen, Rainer; Rittberger, Marc (Hrsg.)(1995): HIM'95. Hypertext, Information Retrieval, Multimedia. Synergieeffekte elektronischer Informationssysteme. Konstanz. 5.-7.April 1995. Konstanz.

Kwok, K. L. (1989): A Neural Network for Probabilistic Information Retrieval. In: Belkin/Rijsbergen 1989. S. 21-30.

Kwok, K. L. (1991a): Query Modification and Expansion in a Network with Adaptive Architecture. In: Bookstein et al. 1991. S. 192-201.

Kwok, K. L. (1991b): Query Learning Using an ANN with Adaptive Architecture. In: Machine Learning ‘91. S. 260-264.

Kwok, K. L. (1996): A New Method of Weighting Query Terms for Ad-Hoc Retrieval. In: Frei et al. 1996 S. 187-195.

Kwok, K. L.; Chan, M. (1998): Improving Two-Stage Ad-Hoc Retrieval for Short Queries. In: Croft et al. 1998. S. 250-256.

Kwok, K.L.; Grunfeld, L. (1994): TREC2 Document Retrieval Experiments using PIRCS. In: Harman 1994. S. 233-242.

Kwok, K.L.; Grunfeld, L. (1996): TREC-4 Ad-Hoc, Routing Retrieval and Filtering Experiments Using PIRCS. In: Harman 1996.

Kwok, K.L.; Grunfeld, L.; Chan, M; Dinstl, N.; Cool, C. (1999): TREC-7 Ad-Hoc, Routing Retrieval and Filtering Experiments Using PIRCS. In: Voorhees/Harman 1999. S. 343-352.

Kwok, K.L.; Papadoupolos, L.; Kwan, Y.Y. (1993): Retrieval Experiments with a Large Collection Using PIRCS. In: Harman 1993. S. 153-172.

Lagus, Krista (1998): Generalizibility of the WEBSOM Method to Document Collections of Various Types. In: Zimmermann 1998. S. 210-214.

Lam, Wai; Ho, Chao Yang (1998): Using a Generalized Instance Set for Automatic Text Categorization. In: Croft, Bruce; Moffat, Alistair; Rijsbergen, K. van; Wilkinson, Ross; Zobel, Justin (Hrsg.): Proceedings of the 21th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR ’98). Melbourne 24.-28.8.98. New York. S. 81-89.

Lamirel, Jean-Charles; Crehange, Marion (1994): Application of a Symbolico-Connectionist Approach for the Design of a Highly Interactive Documentary Database Interrogation System with On-Line Learning Capabilities. In: ACM Conference on Information and Knowledge Management 11/1994. S.155-163.

Layaida, Redouane; Caron, Armand (1994): Applications of the Backpropagation Algorithm to an Information Retrieval System. In: Intelligent Multimedia Information

Retrieval Systems and Management. Proceedings of the RIAO 94 (Recherche d' Information assistée par Ordinateur). Rockfeller University. New York. S. 161-171.

Lee, Joon Ho (1995): Combining Multiple Evidence from Different Properties of Weighting Schemes. In: Fox 1995. S. 180-188.

Lee, Jonghoon; Dubin, David (1999): Context-Sensitive Mapping with a Spreading Activation Network. In: Hearst, Marti; Frederic, Gey; Richard, Tong (Hrsg.): Proceedings of the 22nd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '99). Berkeley, CA 15-19.8.99. New York. S. 198-205.

Lelu, Alain; Francois, Claire (1992): Hypertext Paradigm in the Field of Information Retrieval: a Neural Approach. In: Lucarella, Dario; Nanard, Jocelyne (Hrsg.): Proceedings of the ACM Conference on Hypertext (ECHT '92). Mailand, 30.11.-4.12.92. New York. S. 112-121.

Letsche, Todd (1996): Toward Large-Scale Information Retrieval Using Latent Semantic Indexing. Thesis. Master of Science Degree. University of Tennessee, Knoxville, USA.

Letsche, Todd; Berry, Michael (1997): Large-Scale Information Retrieval with Latent Semantic Indexing. In: Information Science - Applications 100. S. 205-237.

Lesteven, S.; Poincot, P.; Murtagh, F. (1996): Neural Networks and Information Extraction in Astronomical Information Retrieval. In: Vistas in Astronomy 40(3). S. 395-400.

Lewis, David; Schapire, Robert; Callan, James; Papka (1996): Training algorithms for linear text classifiers In: Frei et al. 1996. S. 298-306.

Lin, C.; Chen, Hsinchun (1994): An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents. Technical Report. MIS Department, University of Arizona, Tucson, USA. Juli 1994.

Lin, Xia (1995). Searching and Browsing on Map Displays. In: Kinney, Thomas (Hrsg.): Proceedings of the 58th Annual Meeting of the American Association for Information Science (Vol. 32) ASIS'95. Chicago, Oktober 1995. S. 13-18.

Lin, Xia; Soergel, Dagobert; Marchionini, Gary (1991): A Self-Organizing Semantic Map for Information Retrieval. In: Bookstein et al. 1991. S. 262-269.

Ludwig, Michaela; Mandl, Thomas (1997): Ähnlichkeit von Werkstoffen: Die Anwendung unterschiedlicher Wissensmodellierungstechniken für eine intelligente Komponente von WING. In: Krause/Womser-Hacker 1997. S. 169 - 184.

Lynch, Patrick ; Horton, Sarah (1999): Web Style Guide: Basic Design Principles for Creating Web Sites

MacLeod, Kevin J.; Robertson, W. (1991): A Neural Algorithm for Document Clustering. In: Information Processing & Management 27 (4). S. 337-346.

Mandl, Thomas (1994): Entwicklung eines Ähnlichkeitswerkzeugs auf der Basis neuronaler Netze am Beispiel der Werkstoffinformation. Universität Regensburg, Linguistische Informationswissenschaft, Projekt WING-IIR, Arbeitsbericht 61, Oktober 1994.

Mandl, Thomas (1998): Der Einsatz vager Verfahren für Transformationen. ELVIRA-Arbeitsbericht Nr.13, IZ Sozialwissenschaften, Bonn.

Mandl, Thomas; Schaefer, André; Stempfhuber, Maximilian (1998): Exemplarische Transformationen für die Text-Fakten-Integration. ELVIRA-Arbeitsbericht 15, Informationszentrum Sozialwissenschaften, Bonn.

Mandl, Thomas; Stempfhuber, Maximilian (1998): Softwareergonomische Gestaltung von Wirtschaftsinformationssystemen am Beispiel von ELVIRA. In: Ockenfeld/Mantwill 1998. S. 145-157.

Mandl, Thomas; Womser-Hacker, Christa (1995): „Softcomputing"-Verfahren zur Behandlung von Ähnlichkeit und Vagheit in objektorientierten Informationssystemen. In: Kuhlen/Rittberger 1995. S. 277-292.

Mántaras, Ramon López de; Plaza, Enric (1997): Case-Based Reasoning: an Overview. In: Al Communications 10. S. 21-29.

Marx, Jutta; Schudnagis, Monika (1997): Überblick über die WING-IIR-Benutzertests und methodisches Vorgehen. In: Krause/Womser-Hacker 1997. S 43-58.

Mattox, Dave; Maybury, Mark; Morey, Daryl (1999): Enterprise Expert and Knowledge Discovery. In: Bullinger/Ziegler 1999. Bd. 2, S. 303-307.

Mayer, A.; Mechler, B.; Schlindwein, A.; Wolke, R. (1993): Fuzzy Logic. Einführung und Leitfaden zur praktischen Anwendung. Bonn et al.

McClelland, James; David Rumelhart (1988): Explorations in Parallel Distributed Processing. A Handbook of Models, Programs, and Exercises. Cambridge (MA.) et al.

Medina, J.; Cubero, J.; Pons, O.; Vila, M. (1994): Fuzzy Knowledge Representation in Relational Databases. Arbeitsbericht Universität Granada, DECSAI-94112. November 1994. ftp://decsai.ugr.es/pub/arai/tech_rep/medina/gefred.ps.Z

Medina, J.M.; Pons, O;. Vila, M.A. (1994): GEFRED. A Generalized Model of Fuzzy Relational Databases. In: Information Sciences 76(1-2) S. 87-109.

Merkl, Dieter (1995): Content-Based Document Classification with Highly Compressed Input Data. In: Proceedings of the International Conference on Artificial Neural Networks ICANN '95. Paris. Oktober 9-13 1995. Bd. 2, S. 239-244.

Merkl, Dieter; Tjoa, A Min; Kappel, Gerti (1994): Learning the Semantic Similarity of Reusable Software Components. In: Frakes, William B. (Hrsg.): Proceedings of the Third International Conference on Software Reuse: Advances in Software Reusability. Rio de Janeiro, 1.-4.11. 1994. Washington et al. S. 33-41.

Mizzaro, Stefano (1997): Relevance: The Whole History. In: Journal of the American Society for Information Science. JASIS 48(9). S. 810-832.

Moody, John (1992): The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. In: Moody, John E.; Hanson, S.J.; Lippmann, R.P. (Hrsg.): Advances in Neural Information Processing Systems (NIPS) 4. San Mateo, CA.

Mothe, Josiane (1992): SYRENE: An Information Retrieval System Based on Neural Approaches: Experimental Results. In: Fifth International Conference. Neural Networks and their Applications. NEURO NIMES ´92. Nimes, France, Nov. EC 2, S. 81-91.

Mothe, Josiane (1994): Search Mechanisms Using a Neural Network Model. In: Intelligent Multimedia Information Retrieval Systems and Management. Proceedings of the RIAO ´94 (Recherche d'Information assistée par Ordinateur). Rockfeller University. New York, 11.-13.10.94. S. 275-294.

Mothe, Josiane; Dkaki, Toufiq (1998): Interactive Multidimensional Document Visualization. In: Croft et al. 1998. S. 363-364.

Mothe, Josiane; Soule-Dupuy, C (1992): Integration of a Connectionist Model in Information Retrieval Systems. In: Aleksander, I.; Taylor, J. (Hrsg.): Artificial Neural

Networks, Proceedings of the 1992 International Conference (ICANN-92). Brighton, Sept. 1992. S. 1611-14, vol 2 of 2.

Mönnich, Michael (1999): Kriterien zur Bewertung und Auswahl von Internetsuchmaschinen. In: Schmidt 1999. S. 141-151.

Mori, Hirohiko; Chung, Cheng Long; Kinoe, Yousuke; Hayashi, Yoshio (1990): An Adaptive Document Retrieval System Using a Neural Network. In: International Journal of Human-Computer Interaction 2 (3). S. 267-280.

Müller, A.; Thiel, Ulrich (1994): Query Expansion in a Abductive Information Retrieval System. In: Intelligent Multimedia Information Retrieval Systems and Management. Proceedings of the RIAO ´94 (Recherche d'Information assistée par Ordinateur). Rockfeller University. New York, 11.-13.10.94. S. 461-480.

Nakhaeizadeh, Gholamreza (Hrsg.)(1998): Data Mining: Theoretische Aspekte und Anwendungen. Heidelberg.

Narasimahalu, A. Desai; Leong, Mun-Kew (1995): Experiences with Content Based Retrieval of Multimedia Information. In: Ruthven 1995. S. 1-15.

Nauck, Detlef; Klawon, Frank; Kruse, Rudolf (1994): Neuronale Netze und Fuzzy-Systeme. Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kopplung mit wissensbasierten Methoden. Braunschweig/Wiesbaden.

Newell, Allen; Simon, H.A. (1976): Computer Science as Empirical Inquiry: Symbols and Search. In: Communications of the ACM 19(3). S. 113-126.

Nikolai, Ralf; Traupe, Andreas; Kramer, Ralf (1998): Thesaurus Federations: A Framework for the Flexible Integration of Heterogeneous, Autonomous Thesauri. In: Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries. ADL '98. Santa Barbara, USA, 22.-24.4. Los Alamitos et al. S. 46-55.

Notess, Greg (2000): Fast 300 Million Special Supplement Report. Januar 2000. http://www.notess.com/search/stats/fast300.shtml

Notess, Greg (2000a): Search Engine Statistics: Dead Links Report. Februar 2000. http://www.notess.com/search/stats/dead.shtml

Notess, Greg (2000b): Search Engines Statistics: Database Overlap. Februar 2000. http://www.notess.com/search/stats/overlap.shtml

Oard, Douglas (1997): Alternative Approaches for Cross-Language Text Retrieval. University of Maryland. http://www.clis.umd.edu/dlrg/filter/sss/papers/oard/paper.html

Ockenfeld, Marlies; Mantwill, Gerhard (Hrsg.)(1998): Information und Märkte. Kongreß der Deutschen Gesellschaft für Dokumentation (DGD). Bonn, 22.-24.9.98. Frankfurt.

Orwig, Richard; Chen, Hsinchun; Nunamaker, Jay (1997): A Graphical, Self-Organizing Approach to Classifying Electronic Meeting Output. In: Journal of the American Society for Information Science. JASIS 48(2). S. 157-170.

Papka, Ron; Callan, James; Barto, Andrew (1996): Text-Based Information Retrieval Using Exponentiated Gradient Descent. In: Touretzky, David (Hrsg.): Advances in Neural Information Processing. San Mateo, California. S. 3-9.

Pelletier, Sophie-Julie; Arcand, Jean-Francois; Velissarios, John (1996): STEALTH: A Personal Digital Assistant for Information Filtering. In: Crabtree, Barry (Hrsg.): Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 96). London, 22.-24.4.96. S. 455-474.

Perfetti, Renzo; Massarelli, Emanuele (1997): Training Spatially Homogeneous Fully Recurrent Neural Networks in Eigenvalue Space. In: Neural Networks 10(1). S. 125-137.

Personnaz, L; Guyon, I.; Dreyfus, G. (1986): Neural Network Design for Efficient Information Retrieval. In: Bienenstock, E.; Fogelman Soulie, F.; Weisbuch, G. (Hrsg.): Disordered Systems and Biological Organization. Berlin: Springer. S. 227-231.

Petry, Frederick (1996): Fuzzy Databases: Principles and Applications. Kluwer: Boston et al.

Plaunt, Christian; Norgard, Barbara (1998): An Association-Based Method for Automatic Indexing with a Controlled Vocabulary. In: Journal of the American Society of Information Science JASIS 49(10). S. 888-902.

Rabitti, F.; Savino, P. (1990): Retrieval of Multimedia Documents by Imprecise Query Specification. In: Bancilhon, F.; Thanos, C.; Tsichritzis, D. (Hrsg.): Advances in Database Technology - EDBT'90. Berlin et al. S. 203-218.

Refenes, A.N.; Azema-Barac, M. (1994): Neural Network Applications in Financial Asset Management. In: Neural Computing & Applications 2(1). S.13-39.

Rich, Elaine; Knight, Kevin (1991): Artificial Intelligence. McGraw Hill: New York et al.

Riege, Udo (1998): Thesaurus und Klassifikation Sozialwissenschaften: Entwicklung der elektronsichen Version. In: Ockenfeld/Mantwill 1998. S. 225-254.

Rijsbergen, Keith van (1979): Information Retrieval. London et al.

Rittberger, Marc (1995): Auswahl von Online-Datenbanken - Eine Rechercheschnittstelle für das Online-Retrieval integriert in das Konstanzer Hypertext System. Dissertation, Sozialwissenschaftliche Fakultät der Universität Konstanz.

Rodeghier, Mark (1997): Marktforschung mit SPSS: Analyse, Datenerhebung und Auswertung. Bonn et al.

Rojas, Raúl (1993): Theorie der neuronalen Netze. Eine systematische Einführung. Berlin et al.

Roppel, Stephan (1998): Visualisierung und Adaption: Techniken zur Verbesserung der Interaktion mit hierarchisch strukturierter Information. Konstanz.

Rose, D.; Belew, Richard (1991): A Connectionist and Symbolic Hybrid for Improving Legal Research. In: International Journal of Man-Machine-Studies 31(1). S. 1-33.

Rozmus, J. Michael (1995): Information Retrieval by Self-Organizing Maps. In: Wiliams, Martha (Hrsg.): Proceedings of the 16th National Online Meeting. New York, 2.-4.5.1998. Medford, NJ, 1995. S. 349-354.

Rumelhart, David; McClelland, James; PDP Research Group (1986): Parallel Distributed Processing. Explorations in the Microstructure of Cognition. vol.1: Foundations. vol.2: Psychological and Biological models. Cambridge, MA et al.

Rumelhart, David; Hinton, Geoffrey; McClelland, James (1986): A General Framework for Parallel Distributed Processing. In: Rumelhart/McClelland (1986) vol. 1. S. 45-76.

Salton, Gerard; Buckley, Chris (1988): On the Use of Spreading Activation Methods in Automatic Information Retrieval. In: Chiaramella, Yves (Hrsg.): Proceedings of the 11th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '88). Grenoble 13-15.7.88. New York. S. 147-160.

Salton, Gerard; McGill, Michael (1983): Introduction to Modern Information Retrieval. New York et al.

Schäuble, Peter (1997): Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases. Kluwer.

Schatz, Bruce (1998): High-Performance Distributed Digital Libraries: Building the Interspace on the Grid. In: 7th IEEE Int Symp High-Performance Distributed Computing (July). S. 224-234. http://www.canis.uiuc.edu/archive/papers/hpdc.pdf

Schatz, Bruce; Johnson, Eric; Cochrane, Pauline (1996): Interactive Term Suggestion for Users of Digital Libraries: Using Subject Thesauri and Co-occurrence Lists for Information Retrieval. In: Fox, Edward; Marchionini, Gary (Hrsg.): Proceedings of the 1st ACM International Conference on Digital Libraries. Bethseda, Maryland 20.-23.3.96. New York. S. 126-133.

Schatz, Bruce; Mischo, William; Cole, Timothy; Hardin, Joseph; Bishop, Ann; Chen, Hsinchun (1996): Federating Diverse Collections of Scientific Literature. In: IEEE Computer 29(5). S. 28-36. http://computer.org/computer/dli/r50028/r50028.htm

Scheinost, Ulrich; Haas, Hansjörg; Krause, Jürgen; Lindlbauer, Jürg (Hrsg.)(1998): Marktanalyse und Marktprognose. Das ZVEI Verbandsinformationssystem ELVIRA. Bonn: Informationszentrum Sozialwissenschaften [= Forschungsberichte 2].

Scherer, Andreas (1997): Neuronale Netze: Grundlagen und Anwendungen. Braunschweig, Wiesbaden.

Scheuermann, Peter; Wen-Syan, Li; Clifton, Chris (1998): Multidatabase Query Processing with Uncertainty in Global Keys and Attribute Values. In: Journal of the American Society of Information Science JASIS 49(3). S. 283-301.

Schirmer, Kai; Müller, Axel (1999): Nachrichtenfilterdienste in Deutschland. In: Schmidt 1999. S. 153-156.

Schmidt, Ralph (Hrsg.)(1999): Proceedings der 21. Online Tagung der DGI. Aufbruch ins Wissensmanagement. Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis. Frankfurt, 18.-20.5.99. Frankfurt.

Sciore, Edward; Siegel, Michael; Rosenthal, Arnon (1994): Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems. In: ACM Transactions on Database Systems 19(2). S. 254-290.

Scholtes, J.C. (1992): Neural Nets in Information Retrieval: A Case Study of the 1987 Pravda. In: Ruck, Dennis (Hrsg.): Science of Artificial Neural Networks: Proceedings of the International Society for Optical Engineering, 1710 (SPIE Conf., Orlando, Fl.) S. 631-641.

Schütze, Hinrich; Pedersen, Jan (1997): A Cooccurrence-Based Thesaurus and two Applications to Information Retrieval. In: Information Processing & Management 33(3). S. 307-318.

Schwenker, F.; Sommer, F.; Palm, G. (1996): Iterative Retrieval of Sparsely Coded Associative Memory Patterns. In: Neural Networks 9(3). S. 445-455.

Searle, Warren (2000): Neural Networks: FAQ (Frequently Asked Questions and Answers), Newsgroup comp.ai.neural-nets. ftp://ftp.sas.com/pub/neural/FAQ.html

Sheridan, Páraic, Ballerini, Jean Paul (1996): Experiments in Multilingual Information Retrieval using the SPIDER System. In: Frei et al. 1996. S. 58-65.

Sigel, Alexander (1998): Long-Term Value Adding in an Open Category Network: An Informal Social Approach Towards Relating Conceptual Order Systems on the Internet. In: Zimmermann/Schramm 1998. S. 296-305.

Smeaton, Alan; Wilkinson, Ross (1997): Spanish and Chinese Document Retrieval in TREC-5. In: Voorhees/Harman 1997. S. 57-64.

Smith, Scott; Escobedo, Richard; Anderson, Michael; Caudell, Thomas (1997): A Deployed Engineering Design Retrieval System Using Neural Networks. In: IEEE Transactions on Neural Networks 8(4). S. 847-851.

Smolensky, Paul (1988): On the Proper Treatment of Connectionism. In: Behavioral and Brain Sciences 11. S. 1-74.

Sommer, G. (1978): On fuzzy information retrieval (outline of a fuzzy IR-system for real estate agency services). In: Rose, J. (Hrsg.): Current Topics in Cybernetics and Systems. Amsterdam, 21-25.8.1978. Berlin. S. 380-382.

D'Souza, Daryl; Thom, James (1996): How good are similarity measures across distributed document collections? Technical Report. Department for Computer Science. Royal Melbourne Institute of Technology RMIT. TR 96-27.

Sparck Jones, Karen (1981): The Cranfield Tests. In: Sparck Jones, Karen (Hrsg.): Information Retrieval Experiment. London, Boston, et al. S. 256-284.

Spies, Marcus (1993): Unsicheres Wissen. Wahrscheinlichkeit, Fuzzy-Logik, neuronale Netze und menschliches Denken. Heidelberg et al.

Stafylopatis, Andreas; Likas, Aristidis (1992): Pictorial Information Retrieval Using the Random Neural Network. In: IEEE Transactions on Software Engeneering 18 (7). 1992. S. 590-600.

Statistisches Bundesamt (1994): Gegenüberstellung Güterklassifikation: Sytematisches Güterverzeichnis für Produktionsstatistiken.

Syu, Inien; Lang, S. D. (1994): Heuristic Information Retrieval: A Competition-Based Connectionist Model. In: Intelligent Multimedia Information Retrieval Systems and Management. Proceedings of the RIAO 94 (Recherche d'Information assistée par Ordinateur). Rockfeller University. New York. S. 248-265.

Syu, Inien; Lang, S.; Deo, Narsingh (1996): Incorporating Latent Semantic Indexing into a Neural Network Model for Information Retrieval. In: Barker, Ken; Oezsu, Tamer (Hrsg.): ACM Conference on Information and Knowledge Management (CIKM.'96). Rockville MD. 12.-16-11.96. S. 145-153.

Tahani, Valiollah (1977): A Conceptual Framework for Fuzzy Query Processing - a Step Toward Very Intelligent Database Systems. In: Information Processing & Management 13. S. 289-303.

Tversky, Amor (1977): Features of Similarity. In: Psychological Review 84(4) Juli 1977. S. 327-351.

Voorhees, Ellen; Gupta, Narendra, Johnon-Laird, Ben (1995): Learning Collection Fusion Strategies. In: Fox 1995. S. 172-179.

Voorhees, Ellen; Harman, Donna (Hrsg.)(1997): The Fifth Text Retrieval Conference (TREC-5). NIST Special Publication 500-238. National Institute of Standards and Technology. Gaithersburg, Maryland, 20.-22.11.1996.
http://trec.nist.gov/pubs/trec5/t5_proceedings.html

Voorhees, Ellen; Harman, Donna (1997a): Overview of the Fifth Text REtrieval Conference (TREC-5). In: Voorhees/Harman 1997. S. 1-28.

Voorhees, Ellen; Harman, Donna (Hrsg.)(1998): The Sixth Text Retrieval Conference (TREC-6). NIST Special Publication 500-240. National Institute of Standards and

Technology. Gaithersburg, Maryland, 19.-21.11.1996.
http://trec.nist.gov/pubs/trec6/t6_proceedings.html

Voorhees, Ellen; Harman, Donna (1998a): Overview of the Sixth Text REtrieval Conference (TREC-6). In: Voorhees/Harman 1998. S. 1-24.

Voorhees, Ellen; Harman, Donna (Hrsg.)(1999): The Seventh Text Retrieval Conference (TREC-7). NIST Special Publication 500-242. National Institute of Standards and Technology. Gaithersburg, Maryland, 9.-11.11.1999.
http://trec.nist.gov/pubs/trec7/t7_proceedings.html

Voorhees, Ellen; Harman, Donna (1999a): Overview of the Seventh Text REtrieval Conference (TREC-7). In: Voorhees/Harman 1999. S. 1-24.

Wakimoto, K.; Tanaka, S.; Maeda, A.; Shima, M. (1995): A similarity retrieval method of drawings based on graph representation. In: Systems and Computers in Japan 26 (11). S.100-109.

Waltz, David; Pollack, Jordan (1985): Massively Parallel Parsing: A Strongly Interactive Model of Natural Langauge Interpretation. In: Cognitive Science 9. S. 51-74.

Waard, W. P. de (1994): Neural techniques and postal code detection. In: Pattern Recognition Letters 15 (2). S.199-205.

Wilkinson, Ross; Hingston, P. (1992): Incorporating the vector space model in a neural network used for document retrieval. In: Library HiTech News 10 (1-2), S. 69-75.

Wilkinson, Ross; Zobel, Justin; Sacks-Davis, Ron (1996): Similarity Measures for Short Queries. In: Voorhees/Harman 1996.

Wilkinson, Ross; Hingston, Philip (1991): Using the Cosine Measure in a Neural Network for Document Retrieval. In: Bookstein et al. 1991. S. 202-210.

Wilkinson, Ross (1998): Chinese Document Retrieval at TREC-6. In: Harman/Voorhees 1998. S. 25-29.

Womser-Hacker, Christa (1989): Der PADOK-Retrievaltest. Zur Methode und Verwendung statistischer Verfahren bei der Bewertung von Information-Retrieval-Systemen. Hildesheim et al.

Womser-Hacker, Christa (1997): Das MIMOR-Modell. Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval. Habilitationsschrift. Universität Regensburg, Informationswissenschaft.

Womser-Hacker, Christa (1997a): FUZZY-WING: Ein Werkzeug für Faktenabfragen mit vagen Kriterien. In: Krause/Womser-Hacker 1997. S. 185-203.

Womser-Hacker, Christa; Mandl, Thomas (1999): Adapting Meta Information Retrieval to User Preferences and Document Features. In: Bullinger/Ziegler 1999. Bd. 2 S. 604-608.

Wong, S. K. M.; Cai, Y. J.; Yao, Y. Y. (1993): Computation of Term Associations by a Neural Network. In: Korfhage, Robert; Rasmussen, Edie; Willett, Peter (Hrsg.): Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (SIGIR '93). Pittsburgh 27.6.-1.7.93. New York. S. 107-115.

Wu, Jian-Kang; Lam, Chian-Prong; Methre, Babu; Gao, Yong Jian; Narasimhalu, Arcot, Desai (1996): Content-Based Retrieval for Trademark Registration. In: Furht, Borko (Hrsg.): Multimedia Tools and Applications. An International Journal 3 (3) 1996. S. 245-267.

Yager, Ronald; Larsen, Henrik (1993): Retrieving Information by Fuzzification of Queries. In: Journal of Intelligent Information Systems, Vol. 2. S. 421-441.

Yang, Yiming (1995): Noise Reduction in a Statistical Approach to Text Categorization. In: Fox 1995. S. 256 - 263.

Yasdi, Ramin (1999): A Learning Personal Agent for Information Filtering. In: Bullinger, Hans-Jörg; Vossen, Paul (Hrsg.): HCI International´99 (8th International Conference on Human-Computer Interaction), Adjunct Proceedings. München, 22-27.8.1999. S. 209f.

Young, Paul (1994): Cross-Language Information Retrieval Using Latent Semantic Indexing. Technical Report. University of Tennessee at Knoxville. http://www.cs.utk.edu/~library/TechReports/1994/ut-cs-94-259.ps.Z

Zadeh, Lofti (1965): Fuzzy Sets. In: Information and Control 8. S.338-353.

Zadeh, Lofti (1994): What is BISC? http://http.cs.berkeley.edu/projects/Bisc/bisc.memo.html#what_is_sc

Zavrel, Jakub (1996): Neural Navigation Interfaces for Information Retrieval: Are They More than an Appealing Idea? In: Artificial Intelligence Review 10. S. 477-504.

Zell, Andreas (1994): Simulation neuronaler Netze. Bonn et al.

Zell, Andreas; Mamier, Günter; Vogt, Michael; Mache, Niels; Hübner, Ralf; Döring, Sven; et al. (1995): SNNS. Stuttgart Neural Network Simulator. User Manual, Version 4.1. Universität Stuttgart. IPVR. Report Nr. 6/95. http://www-ra.informatik.uni-tuebingen.de/SNNS/UserManual/UserManual.html

Zimmer, Monika (1998): SOLIS – Sozialwissenschaftliches Literaturinformationssystem. http://www.bonn.iz-soz.de/information/databases/solis/index.htm

Zimmer, Monika (1998a): FORIS - Forschungsinformationssystem Sozialwissenschaften. http://www.bonn.iz-soz.de/information/databases/foris/index.htm

Zimmermann, Hans-Jürgen (Hrsg.)(1995): Datenanalyse: Anwendung von DataEngine mit Fuzzy Technologien und Neuronalen Netzen. Düsseldorf.

Zimmermann, Hans-Jürgen (Hrsg.)(1996): EUFIT ‘96. 4th European Congress on Intelligent Techniques and Soft Computing. Aachen, 1996.

Zimmermann, Hans-Jürgen (Hrsg.)(1997): EUFIT ‘97. 5th European Congress on Intelligent Techniques and Soft Computing. Aachen, 8.-11.9.1997.

Zimmermann, Hans-Jürgen (Hrsg.)(1998): EUFIT ‘98. 6th European Congress on Intelligent Techniques and Soft Computing. Aachen, 8.-10.9.1998.

Zimmermann, Hans-Jürgen (Hrsg.)(1999): EUFIT ‘99. 7th European Congress on Intelligent Techniques and Soft Computing. Aachen, 13.-16.9.1999.

Zimmermann, Harald; Schramm, Volker (Hrsg.)(1998): Knowledge Management und Kommunikationssysteme: Workflow Management, Multimedia, Knowledge Transfer. Proceedings 6. Int. Symposium für Informationswissenschaft. (ISI ´98). 3.-7.11.98, Karlsuniversität Prag. Konstanz.

Abkürzungsverzeichnis

AIR	Adaptive Information Retrieval
ART	Adaptive Resonance Theory
BFAI	Bundesstelle für Außenhandelsinformationen, Köln
COSIMIR	Cognitive Similarity Learning in Information Retrieval
DBMS	Database Management System
DIW	Deutsches Institut für Wirtschaftsforschung, Berlin
ELVIRA	Elektronisches Verbandsinformations-, Recherche- und Analysesystem
GP 95	Güterverzeichnis für Produktionsstatistiken, Stand 1995
HVB	Hauptverband der Deutschen Bauindustrie, Wiesbaden
IR	Information Retrieval
IZ	Informationszentrum Sozialwissenschaften, Bonn
KI	Künstlichen Intelligenz
LSI	Latent Semantic Indexing
OECD	Organisation for Economic Cooperation and Development
PIRCS	Probabilistic Indexing and Retrieval-Component-System
SNNS	Stuttgarter Neuronaler Netzerk Simulator
SOM	Self-Organizing Maps
SQL	Structured Query Language
USB	Universitäts- und Stadtbibliothek Köln
TREC	Text Retrieval Conference
SVD	Singular Value Decomposition
VDMA	Verband Deutscher Maschinen- und Anlagenbau, Frankfurt
WING	Werkstoffinformationssystem mit natürlichsprachlicher/graphischer Benutzungsoberfläche
WA	Warenverzeichnis für die Außenhandelsstatistik
WOB	Auf der Werkzeugmetapher beruhende, strikt objektorientierte Benutzungsoberflächen
ZVEI	Zentralverband der Elektrotechnik- und Elektronikindustrie, Frankfurt

Abbildungsverzeichnis