Automatisches Klassifizieren bibliographischer Beschreibungsdaten

Vorgehensweise und Ergebnisse

Diplomarbeit
Studiengang Bibliothekswesen
Fakultät für Informations- und Kommunikationswissenschaften
Fachhochschule Köln

vorgelegt von:

Jens Wille
Hahnenstr. 17 / Et-Zi 03-61.1
50354 Hürth
Matrikel-Nr.: 110 360 27

am 27. Juni 2006.

Erstgutachter: Prof. Dipl.-Math. Winfried Gödert
Zweitgutachter: Prof. Dr. Klaus Lepsky

Zusammenfassung

Diese Arbeit befasst sich mit den praktischen Aspekten des Automatischen Klassifizierens bibliographischer Referenzdaten. Im Vordergrund steht die konkrete Vorgehensweise anhand des eigens zu diesem Zweck entwickelten Open Source-Programms „COBRA – Classification Of Bibliographic Records, Automatic“. Es werden die Rahmenbedingungen und Parameter für einen Einsatz im bibliothekarischen Umfeld geklärt. Schließlich erfolgt eine Auswertung von Klassifizierungsergebnissen am Beispiel sozialwissenschaftlicher Daten aus der Datenbank SOLIS.

Schlagworte: Automatisches Klassifizieren, Automatisches Indexieren, Bibliographische Referenzdaten, Open Source-Software, Evaluation

Abstract

This work deals with the practical aspects of automated categorization of bibliographic records. Its main concern regards the course of action within the ad hoc developed open source program „COBRA – Classification Of Bibliographic Records, Automatic“. Preconditions and parameters for application in the library field are clarified. Finally, categorization results of socio-scientific records from the database SOLIS are evaluated.

Keywords: Automated text categorization, Automatic indexing, Bibliographic records, Open source software, Evaluation

Inhaltsverzeichnis

Abkürzungsverzeichnis

COBRA	Classification Of Bibliographic Records, Automatic
CPAN	Comprehensive Perl Archive Network
E-LIS	E-prints in Library and Information Science
KNN	k-Nearest Neighbour (Klassifizierungsalgorithmus)
MAB	Maschinelles Austauschformat für Bibliotheken
MARC	MAchine Readable Cataloging
OPAC	Online Public Access Catalogue
SOLIS	SOzialwissenschaftliches LiteraturInformationsSystem
STN	The Scientific & Technical Information Network
SVM	Support Vector Machine (Klassifizierungsalgorithmus)

1 Einleitung

Die vorliegende Arbeit befasst sich mit den praktischen Aspekten des Automatischen Klassifizierens bibliographischer Daten und ging aus einem studentischen Schwerpunktprojekt zum Thema „Automatische Klassifizierung von bibliographischen Referenzdaten“ hervor, welches im Wintersemester 2005/06 an der Fachhochschule Köln stattfand.

1.1 Thematische Einordnung

Für die verbale Sacherschließung gibt es mit der Automatischen Indexierung bereits vermehrt zur Anwendung kommende Verfahren, die hier unterstützend eingreifen. Für die klassifikatorische Sacherschließung stehen Verfahren zur Automatischen Klassifizierung zur Verfügung, deren praktischer Einsatz im Folgenden exemplarisch dargestellt werden soll.

Unter Automatischer Klassifizierung ist die maschinelle Zuweisung von Klassen eines Klassifikationssystems zu Dokumenten zu verstehen,[1] wobei heutzutage zumeist maschinelle Lernverfahren zum Einsatz kommen.[2] Als „Dokumente“ treten im bibliothekarischen Bereich hauptsächlich bibliographische Beschreibungen stellvertretend für Bücher,

[1] Oberhauser 2005, S. 12
[2] Oberhauser 2005, S. 22

Aufsätze u.dgl. auf, was besondere Anforderungen an die maschinellen Lernverfahren auf der einen und die bibliographischen Beschreibungen auf der anderen Seite stellt.

Für einen theoretischen Einstieg in die Automatische Klassifizierung sei z.B. auf Oberhauser 2005 oder Sebastiani 1999 verwiesen.

1.2 Aufgabenstellung

Nachdem mit Oberhauser 2005 eine umfassende theoretische Behandlung dieses Themas in deutscher Sprache vorliegt, stellt sich die Frage, wie die Verfahren und Konzepte umzusetzen und in den bibliothekarischen Geschäftsgang zu integrieren sind. Zwar existiert eine Reihe von kommerziellen wie auch freien Produkten,[3] aber diese sind zumeist auf elektronische Volltexte spezialisiert und werden somit den bibliothekarischen Anforderungen nicht ausreichend gerecht.

Daher wurde, beginnend mit dem bereits erwähnten Projekt, auf der Basis einer existierenden Lösung (dem Perl-Modul AI::Categorizer[4] von Ken Williams) vom Verfasser ein Programm entwickelt, welches einerseits auf die Bedürfnisse bibliothekarischer Anwendungen zugeschnitten und andererseits auch einfach zu handhaben ist. Zum Einsatz kam hier die Programmiersprache Perl,[5] so dass – entsprechende Kenntnisse vorausgesetzt – die Möglichkeit, Anpassungen und Erweiterungen vorzunehmen, gewährleistet ist. Das Perl-Modul AI::Categorizer unterstützt eine Reihe von Klassifizierungsalgorithmen und ist somit gut geeignet, verschiedene Ansätze zu testen und zu vergleichen.

Des Weiteren spielt eine wichtige Rolle, inwiefern die Güte solcher automatisch erzielter Klassifizierungen akzeptabel ist und im bibliothekarischen Alltag Nutzen bringen kann.

[3] Vgl. Oberhauser 2005, Kap. 2.7
[4] `http://search.cpan.org/dist/AI-Categorizer/lib/AI/Categorizer.pm` [26. 06. 2006]
[5] `http://www.perl.org/` [26. 06. 2006]

Hier kommen verschiedenste Parameter zum Tragen, die es auf die jeweilige Umgebung abzustimmen gilt.

1.3 Zielsetzung

Diese Arbeit verfolgt das Ziel, unter Einsatz der eigens zu diesem Zweck entwickelten Programmumgebung „COBRA – Classification Of Bibliographic Records, Automatic“[6] die Durchführung einer Automatischen Klassifizierung, einschließlich der Übertragung und Anpassung der Vorgehensweise auf die eigenen Anforderungen, zu ermöglichen.

Dazu werden die notwendigen Voraussetzungen für eine Automatische Klassifizierung bibliographischer Daten geklärt und der Vorgang am Beispiel sozialwissenschaftlicher Datensätze aus der Datenbank SOLIS[7] dargestellt. Eine besondere Rolle kommt hier dem Herausarbeiten von möglichen Parametern zu, die sich auf die Klassifizierung auswirken.

Darüber hinaus werden die erzeugten Klassifizierungsergebnisse auf ihre Güte hin untersucht und Anhaltspunkte dafür geliefert, was mit Automatischer Klassifizierung im Vergleich zu intellektueller Klassifizierung zu erreichen ist und was nicht. Allerdings kann auch dies nur als Veranschaulichung der Vorgehensweise gesehen werden, da eine aussagekräftige Beurteilung auf einer größeren Datenbasis beruhen und vor allem auch die jeweiligen Anforderungen und Umstände berücksichtigen muss.

[6] Die Benennung erfolgte in Anlehnung an Sebastiani 2006, „Classification of text, automatic“.
[7] `http://www.gesis.org/Information/SOLIS/` [26. 06. 2006]

1.4 Aufbau der Arbeit

Zunächst werden die **Voraussetzungen und Rahmenbedingungen** für die Durchführung einer Automatischen Klassifizierung erörtert (Kap. 2). Dabei geht es insbesondere um die Beschaffenheit der Ausgangsdaten und des zu Grunde liegenden Klassifikationssystems, sowie um benötigte Tools.

Daraufhin wird auf die **Aufbereitung der Daten** eingegangen, was die Überführung in ein geeignetes Format und eine eventuelle Automatische Indexierung umfasst (Kap. 3).

Anschließend wird dann der **Klassifizierungsvorgang** anhand der Programmumgebung COBRA im Einzelnen dargestellt (Kap. 4). Dies beinhaltet sowohl die Anbindung an das Bibliothekssystem bzw. die Datenbank (Import und Export) als auch das Trainieren eines Klassifikators und schließlich die eigentliche Klassifizierung.

Darauf folgt eine **Diskussion der Klassifizierungsergebnisse**, die insbesondere auch aufzuzeigen versucht, an welchen Stellen möglicherweise noch Optimierungen vorgenommen werden können (Kap. 5).

Die Arbeit schließt mit einer **Zusammenfassung** der Ergebnisse und einem **Ausblick** auf mögliche Weiterentwicklungen (Kap. 6).

Im Anhang findet sich eine Beschreibung der Zugriffsmöglichkeiten auf das Programm COBRA, einschließlich dessen Quellcode, sowie auf die verwendeten Daten und die erzeugten Ergebnislisten (Anhang A).

1.5 Danksagung

Mein besonderer Dank gilt Herrn Prof. Gödert und Herrn Prof. Lepsky für die Betreuung dieser Arbeit und die allseitige Unterstützung während des Studiums. Des Weiteren möchte ich Herrn Dr. Oberhauser für die theoretische Aufarbeitung dieses Themenkomplexes danken, was es mir erlaubte, mich in dieser Arbeit den praktischen Aspekten zuzuwenden. Schließlich danke ich Herrn Prof. Jüngling und meinen Kommilitoninnen für die Zusammenarbeit im Projekt „Automatische Klassifizierung von bibliographischen Referenzdaten“.

2 Rahmenbedingungen

Im Folgenden sollen die wichtigsten Voraussetzungen und Parameter für die Automatische Klassifizierung bibliographischer Daten benannt werden. Dabei steht weniger eine Analyse ihrer konkreten Auswirkungen im Vordergrund als vielmehr eine möglichst vollständige Aufzählung aller beeinflussenden und beeinflussbaren Rahmenbedingungen.

Von der Kenntnis dieser Faktoren hängt in besonderer Weise ab, wie eine einzusetzende Software auf die eigenen Bedürfnisse abzustimmen ist, aber auch welche Aspekte bei einer Evaluation der Ergebnisse zu berücksichtigen sind, vor allem im Hinblick auf die Vergleichbarkeit derartiger Untersuchungen. Somit soll diese Aufzählung ein Rahmengerüst bieten, anhand dessen ein automatischer Klassifizierungsvorgang erfasst und beurteilt werden kann.

2.1 Ausgangsdaten

Die Ausgangsdaten sind es, von denen bei einer Automatischen Klassifizierung unter Einsatz maschineller Lernverfahren alles abhängt: Ausgehend von einer bereits erschlossenen Datenmenge erstellt das Klassifikationsprogramm während der Trainingsphase ein Modell jeder Klasse. Daraus folgt, dass zum einen für jede Klasse des Klassifikationssystems eine ausreichende Zahl an Beispieldokumenten vorhanden sein muss, und dass zum

anderen thematische und sprachliche Homogenität eine erhebliche Rolle spielen.

Letzteres beinhaltet vor allem die Dokumentsprache(n) und den Anteil an fachsprachlicher Terminologie. Hier können Verfahren zur Automatischen Indexierung ansetzen, um einerseits die sprachliche Vielfalt zu reduzieren und andererseits die Dokumente beispielsweise mit Synonymen anzureichern.

Im Kontext dieser Arbeit soll ferner auf die spezifischen Eigenschaften bibliographischer Referenzdaten eingegangen werden. Es sind also auch das Datenschema und die Art der vorhandenen Kategorien zu berücksichtigen; eine besondere Rolle kommt dabei inhaltserschließenden Elementen zu. Schlagworte, Abstracts u.dgl. liefern wichtige Informationen über den bloßen Titel hinaus, und können somit einen erheblichen Einfluss auf den Klassifizierungsvorgang und dessen Güte haben.

Bezug zu COBRA und den durchgeführten Experimenten

Für die exemplarische Durchführung im Rahmen dieser Arbeit, wie auch schon in dem erwähnten Projekt „Automatische Klassifizierung von bibliographischen Referenzdaten“, kamen knapp 5.000 sozialwissenschaftliche Datensätze aus der Datenbank SOLIS zusammen mit der „Klassifikation Sozialwissenschaften“ (s. Abschnitt 2.2) zum Einsatz. Hierbei wurde eine Beschränkung auf deutschsprachige Titel vorgenommen und es wurden sowohl Tests mit ebendiesen Daten als auch mit automatisch erstellten Indexaten[1] durchgeführt. Die Daten waren neben Titel (`TI`) und Verfasserangabe (unberücksichtigt) mit Deskriptoren des „Thesaurus Sozialwissenschaften“[2] (`CT`), einer Aufzählung der verwendeten Methoden[3] (`ME`), einem deutschsprachigen Abstract (`AB`) und einer Sprachangabe (`LA`) versehen.[4] Listing 2.1 zeigt einen Beispieldatensatz, wobei der Zeilenumbruch allerdings nicht dem Original entspricht, sondern darstellungsbedingt ist.

[1] Unter Verwendung der Software Lingo: `http://lex-lingo.de/` [26. 06. 2006]
[2] `http://www.gesis.org/Information/Rechercheunterst/#Thesaurus` [26. 06. 2006]
[3] `http://www.gesis.org/Information/Rechercheunterst/Methodenliste/` [26. 06. 2006]
[4] Vgl. auch das Datenblatt bei STN: `http://info.cas.org/ONLINE/DBSS/solisss.html` [26. 06. 2006]

Listing 2.1: SOLIS-Datensatz im STN-Format

```
DN 20050100886
TI Versuchungen der Unfreiheit: Erasmus-Intellektuelle im Zeitalter des
   Totalitarismus.
LA Deutsch
AB "Kommunismus und Faschismus waren im letzten Jahrhundert die grossen
   Versuchungen auch fuer Intellektuelle. Aber wer von ihnen war gegen
   diese Totalitarismen immun? Anhang von drei Helden - Karl Popper,
   Raymond Aron, Isaiah Berlin - zeigt der Autor auf, welche vier Tugen-
   den ihnen die Kraft gaben, den Versuchungen der Zeit zu widerstehen.
   Die Summe dieser Grundhaltungen laesst sich in einer Person - einem
   oeffentlichen Intellektuellen - auf den Begriff bringen: Erasmus von
   Rotterdam." (Autorenreferat)
ME deskriptive Studie; normativ
CT Intellektueller; Totalitarismus; Widerstand; politische Kultur; Hu-
   manismus; Renaissance; Tugend; Popper, K.; Kritik; Kritikfaehigkeit;
   Kritischer Rationalismus; Persoenlichkeit
CC *10501; 10101; 10220
```

2.2 Klassifikationssystem

Das zu Grunde liegende Klassifikationssystem steht zunächst einmal außen vor; es findet nur insofern Berücksichtigung als es implizit in den Trainingsdaten enthalten ist. Als Rahmenwerte sind hier also die Anzahl der Klassen und die Tiefe der Hierarchisierung zu nennen. Soll das Klassifikationssystem aber explizit für das Training oder im Zuge von Tests auf Klassenintegrität verwendet werden, gelten ähnliche Bedingungen wie für die Ausgangsdaten.

Bezug zu COBRA und den durchgeführten Experimenten

Die „Klassifikation Sozialwissenschaften“[5] enthält 159 Klassen auf bis zu 4 Hierarchieebe-

[5] `http://www.gesis.org/Information/Rechercheunterst/Klassifikation/` [26. 06. 2006]

nen. Sie ist auf Deutsch verfasst und beinhaltet neben der eigentlichen Klassenbenennung auch zahlreiche Beispiele und Scope Notes sowie Einschließungs- und Ausschließungshinweise für einige Klassen. Listing 2.2 zeigt einen Ausschnitt der Klassifikation unterhalb der Klasse „Kommunikationswissenschaften“.

Listing 2.2: „Klassifikation Sozialwissenschaften“ (Ausschnitt)

```
10800   Kommunikationswissenschaften (einschließlich Publizistik, Infor-
        mationswissenschaft, Bibliothekswissenschaft)
10801   Allgemeines, spezielle Theorien und "Schulen", Methoden, Ent-
        wicklung und Geschichte der Kommunikationswissenschaften

        Beispiele: Symbolismus, Zeichentheorie, Kommunikationsgeschichte
10802   Lehre und Studium, Professionalisierung und Ethik, Organisatio-
        nen und Verbände der Kommunikationswissenschaften
10803   interpersonelle Kommunikation

        Beispiele: dialogische Kommunikation, Kommunikation in Gruppen
        und Organisationen, Gesprächsanalyse, Kommunikationsstruktu-
        ren, einseitige interpersonale Kommunikation (Reden, Vorträge,
        Predigten u.a.)
1080400 Massenkommunikation (medienübergreifende Darstellungen)
1080401 Rundfunk, Telekommunikation

        Beispiele: Hörfunk, Fernsehen
1080402 Druckmedien

        Beispiele: Presse, Bücher
1080403 andere Medien

        Beispiele: Film, Musik, Theater

        Beinhaltet hier nicht:
        Druckmedien (ist bei 1080402 - Druckmedien)
        elektronische Medien (ist bei 1080404 - interaktive, elektroni-
        sche Medien)
        Rundfunk (ist bei 1080401 - Rundfunk, Telekommunikation)
        Telekommunikation (ist bei 1080401 - Rundfunk, Telekommunikati-
        on)
```

Forts. auf der nächsten Seite

Forts. von der vorherigen Seite

```
1080404 interaktive, elektronische Medien

        Beispiele: Multimedia, Internet, Computerspiel
1080405 Medieninhalte, Aussagenforschung
1080406 Kommunikatorforschung, Journalismus
1080407 Wirkungsforschung, Rezipientenforschung
1080408 Meinungsforschung
1080409 Werbung, Public Relations, Öffentlichkeitsarbeit
1080410 Medienpädagogik
1080411 Medienpolitik, Informationspolitik, Medienrecht
1080412 Medienökonomie, Medientechnik
1080500 Informationswissenschaft
1080501 Information und Dokumentation, Bibliotheken, Archive
1080502 Informationsmanagement, informationelle Prozesse, Informations-
        ökonomie
1080503 Szientometrie, Bibliometrie, Informetrie
10899   Sonstiges zu Kommunikationswissenschaften
```

2.3 Tools

Die benötigten Tools sind zu heterogen, um hier näher auf deren Rahmenbedingungen und Systemanforderungen einzugehen. Hervorzuheben sind darunter aber die Fähigkeiten zum Im- und Export der Daten, da hierdurch die Integration in den Geschäftsgang maßgeblich beeinflusst wird, und Forderungen nach vorinstallierten Programmiersprachen oder Laufzeitumgebungen. Neben dem eigentlichen Klassifizierungsprogramm werden Tools zur Datenhaltung (i.d.R. bereits vorhanden), bei Bedarf zum Konvertieren der Daten, evtl. zur Automatischen Indexierung und schließlich zur Integration in einen handhabbaren, in sich abgeschlossenen Workflow benötigt.

Bezug zu COBRA und den durchgeführten Experimenten

Die hier vorgestellte Programmumgebung „COBRA – Classification Of Bibliographic Records, Automatic“ übernimmt dabei vor allem die letztgenannte Rolle: Sie integriert

das Klassifizierungsmodul AI::Categorizer und bietet flexible Schnittstellen zum Datenhaltungssystem einerseits und zur Einbindung einer Automatischen Indexierung andererseits (siehe hierzu auch Abschnitt 3.2).

2.4 Lernverfahren

Auf die Funktionsweise und die zahlreichen Parameter maschineller Lernverfahren kann hier aus Platzgründen nicht näher eingegangen werden. Daher soll an dieser Stelle nur auf einschlägige Literaturquellen verwiesen werden: Für eine übersichtliche Einführung siehe Oberhauser 2005, Kap. 2, für eine detailliertere Behandlung Sebastiani 2002 oder Mitchell 1997; auch die Dokumentation zu AI::Categorizer[6] hält hierzu hilfreiche Informationen bereit.

2.5 Zusammenfassung

Abschließend eine Übersicht über die maßgeblichen Faktoren im Zusammenhang mit der Automatischen Klassifizierung bibliographischer Referenzdaten in der Zusammenfassung:

- Anzahl an Beispieldokumenten je Klasse (allgemein die Verteilung der Dokumente über die Klassen)
- thematische Homogenität der Ausgangsdaten
- sprachliche Homogenität der Ausgangsdaten (insbesondere Dokumentsprache(n) und fachsprachliche Terminologie)

[6] `http://search.cpan.org/dist/AI-Categorizer/` [26. 06. 2006]

- Datenschema; Anzahl und Art der Kategorien und deren Inhalte (z.B. kontrolliertes Vokabular oder Freitext)
- Anzahl der in den Dokumenten vertretenen Klassen
- Tiefe der Hierarchisierung des Klassifikationssystems
- eingesetzte(s) Lernverfahren
- evtl. Automatische Indexierung (welche wiederum eigene Rahmenbedingungen und Parameter mit sich bringt)

3 Aufbereitung der Daten

Damit die zu klassifizierenden Daten von der Klassifizierungssoftware verarbeitet werden können, ist es in der Regel erforderlich, jene aus dem Datenhaltungssystem zu exportieren und in eine geeignete Form zu bringen. Zugleich kann an dieser Stelle angesetzt werden, die Daten automatisch zu indexieren, mit dem Ziel, die Datensätze um die Indexate anzureichern bzw. durch selbige zu ersetzen.

3.1 Geeignetes Format

Für den Datenaustausch zwischen verschiedenen Programmen gibt es unterschiedliche Ansätze: Entweder die Daten werden in ein vom Zielprogramm unterstütztes Format konvertiert oder es steht ein gemeinsames Austauschformat zur Verfügung, welches das Ausgangssystem exportieren und das Zielsystem importieren kann (bibliographische Austauschformate wie MAB oder MARC wären hier etwa zu nennen – sofern vom Zielsystem unterstützt); schließlich verbleibt die Möglichkeit, das Zielsystem in geeigneter Weise anzupassen, was allerdings einschlägige Programmierkenntnisse und eine (idealerweise offen dokumentierte) Programmierschnittstelle oder zumindest offene Quellen des zu modifizierenden Programms voraussetzt.

Bezug zu COBRA und den durchgeführten Experimenten

Im Falle von COBRA wird zunächst nur das STN-Format der Datenbank SOLIS unterstützt (vgl. Listing 2.1), neben diesem Originalformat allerdings auch eine modifizierte Variante wie sie in dem Projekt „Automatische Klassifizierung von bibliographischen Referenzdaten“ zum Einsatz kam: Die einzelnen Datensätze beginnen mit einer Zeile, welche die Identifikationsnummer dieses Datensatzes enthält, und jede Zeile eines Datensatzes beginnt mit dem Feldbezeichner, gefolgt von einem Trennzeichen (bzw. einer Zeichenfolge), gefolgt von dem Feldinhalt. Listing 3.1 zeigt ein Beispiel (wobei auch hier der Zeilenumbruch darstellungsbedingt ist und somit einer der Hauptunterschiede zum Originalformat nicht hinreichend zur Geltung kommen kann).

Listing 3.1: SOLIS-Datensatz im „vereinfachten“ STN-Format

```
#20050100886
TI: Versuchungen der Unfreiheit: Erasmus-Intellektuelle im Zeitalter des
    Totalitarismus.
ME: deskriptive Studie; normativ
LA: Deutsch
DN: 20050100886
CT: Intellektueller; Totalitarismus; Widerstand; politische Kultur; Hu-
    manismus; Renaissance; Tugend; Popper, K.; Kritik; Kritikfaehigkeit;
    Kritischer Rationalismus; Persoenlichkeit
CC: *10501; 10101; 10220
AB: "Kommunismus und Faschismus waren im letzten Jahrhundert die gros-
    sen Versuchungen auchfuer Intellektuelle. Aber wer von ihnen war
    gegen diese Totalitarismen immun? Anhang von drei Helden - Karl Pop-
    per, Raymond Aron, Isaiah Berlin - zeigt der Autor auf, welche vier
    Tugenden ihnen die Kraft gaben, den Versuchungen der Zeit zu wider-
    stehen. Die Summe dieser Grundhaltungen laesst sich in einer Person
    - einem oeffentlichen Intellektuellen - auf den Begriff bringen:
    Erasmus von Rotterdam." (Autorenreferat)
```

Im ursprünglichen Standardausgabeformat von STN kann ein Feld über mehrere Zeilen gehen, wovon nur die erste den Bezeichner enthält. Außerdem wurde zum damaligen

Zeitpunkt die Logdatei der STN-Sitzung als Ausgangsdatei für die weitere Verarbeitung verwendet, welche eben auch für diese Zwecke ungewollte Informationen enthielt, die herausgefiltert werden mussten. Diese Variante wurde demzufolge während des Projektes gewählt, um eine Bereinigung der Ausgangsdaten vornehmen zu können und im Zuge dessen gleich ein leichter zu verarbeitendes Format zu erhalten.

Darüber hinaus wurde die Unterstützung für ein generisches Format implementiert, welches in seinen Grundzügen dem vereinfachten STN-Format entspricht, und es ist ohne weiteres möglich, COBRA um Unterstützung für das jeweils benötigte Format zu erweitern, was allerdings entsprechende Programmierkenntnisse erfordert.

3.2 Automatische Indexierung

Wie bereits angesprochen kann eine Automatische Indexierung dazu dienen, die sprachliche, vor allem grammatische, Vielfalt der Ausgangsdaten zu reduzieren und aber auch die Daten um weitere Begriffe anzureichern. Für eine Einführung in die Automatische Indexierung siehe z.B. Lepsky 1994 oder Nohr 2003.

Bezug zu COBRA und den durchgeführten Experimenten
COBRA bietet die Möglichkeit, unter Einsatz der Indexierungssoftware Lingo[1] die Datensätze automatisch zu indexieren, bevor sie der Klassifizierung zugeführt werden. Dabei besteht insbesondere die Wahl, die Datensätzen um die erzeugten Indexate zu ergänzen oder vollständig durch diese zu ersetzen. Lingo bietet seinerseits zahlreiche Optionen zur Erzeugung von Indexaten, siehe dazu die genannte Website oder auch Lepsky u. Vorhauer 2006.

Für die vorliegende Arbeit wurde auf Basis eines im Projekt „Automatische Klassifizie-

[1] `http://www.lex-lingo.de/` [26. 06. 2006]

rung von bibliographischen Referenzdaten“ erstellten Benutzerwörterbuches eine Lemmatisierung, Kompositumzerlegung und Mehrworterkennung vorgenommen. Die Relationierung mit Synonymen basierte auf dem Lingo-Synonymwörterbuch.

Die generierten Indexterme für die Kategorien `TI`, `CT`, `AB` und `ME` des Datensatzes aus Listing 3.1 sehen wie folgt aus:

Listing 3.2: Lingo-Indexat zu Listing 3.1 (Ausschnitt; mit Angabe der absoluten Termhäufigkeiten)

```
3 intellektuell
3 versuchung
2 intellektuelle
2 popper
2 totalitarismus
2 tugend
1 anhang
1 aron
1 autor
1 autorenreferat
1 begreifen
1 begriff
1 berlin
1 bringen
1 deskriptiv
1 einer
1 erasmus
1 erasmus-intellektuelle
1 faschismus
1 gabe
1 gross
1 grundhaltung
1 held
1 humanismus
1 immun
1 intellektueller
...
```

3.3 Zusammenfassung

Als vorbereitende Schritte zur Automatischen Klassifizierung sind bei Bedarf durchzuführen:

- Überführung der exportierten Daten in ein geeignetes, vom Klassifizierungssystem unterstütztes Format
- evtl. Automatische Indexierung (insbesondere Lemmatisierung, Kompositumzerlegung und Anreicherung mit Synonymen/Übersetzungen; evtl. Wortableitungen)

4 Durchführung

Dieses Kapitel stellt die Vorgehensweise einer Automatischen Klassifizierung exemplarisch anhand des Programms „COBRA – Classification Of Bibliographic Records, Automatic“ dar. Zwar sind die Grundelemente wie sie bei COBRA in Form sog. Aktionen auftreten als abstrakte Phasen des Klassifizierungsvorgangs verallgemeinerungsfähig und dementsprechend auf andere Programme übertragbar, aber die konkreten Details unterscheiden sich naturgemäß von Software zu Software.

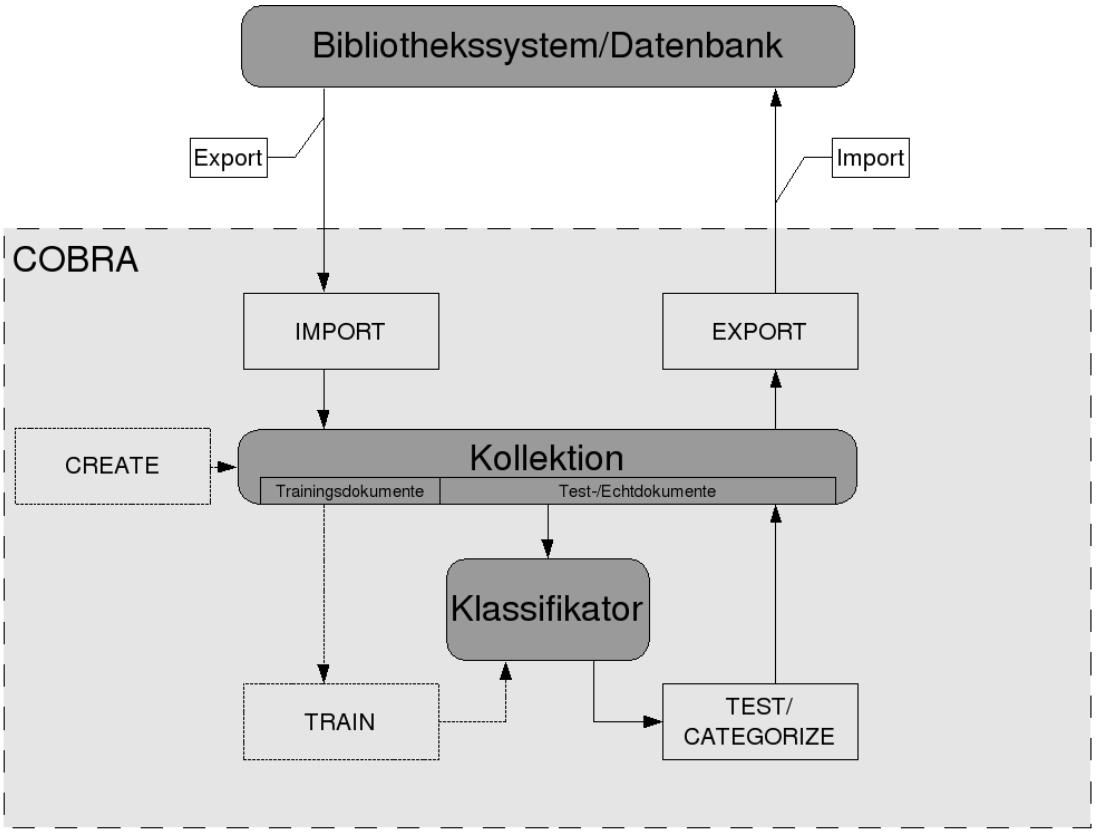
Da sich die Darstellungen dieses Kapitels wie eben erläutert sehr stark an COBRA orientieren und da die Funktionsweise und das Benutzungsinterface dieses Programms aber auch einem Wandel unterliegen, sei für detailliertere und vor allem aktuelle Informationen auf die entsprechende Dokumentation verwiesen.[1]

Der Klassifizierungsvorgang läßt sich grob unterteilen in den Import der zu klassifizierenden Daten (welche zuvor aus dem Datenhaltungssystem exportiert worden sind), das Trainieren eines Klassifikators, die eigentliche Klassifizierung und schließlich den Export (um die klassifizierten Daten wieder in das Datenhaltungssystem importieren zu können). Während das Training im Echtbetrieb nur einmal zu Beginn durchgeführt werden wird, kann zwischen das Training und das Klassifizieren oder auch anstelle des letzteren das Testen des erstellten Klassifikators treten; dies geschieht in der Regel aber auch nur

[1] Anhang A gibt hierzu nähere Auskunft.

jeweils nach dem erneuten Trainieren eines Klassifikators. Abbildung 4.1 veranschaulicht den gesamten Ablauf.

Abbildung 4.1: Klassifizierungsworkflow mit COBRA: Übersicht



COBRA enthält insofern eine Besonderheit als die Daten in sog. Kollektionen organisiert werden. Dies ermöglicht es, für bestimmte Konstellationen von Ausgangsdaten und Zielsetzungen (z.B. für das Testen verschiedener Lernverfahren oder den Vergleich von automatisch indexierten und unbehandelten Daten) jeweils eigene Kollektionen anzulegen. Dies entspricht dann der ersten Phase (=Aktion). Hinzu kommen weitere Aktionen, die aber für die Zwecke dieses Kapitels nicht weiter von Belang sind und deren Bedeutung und Optionen der COBRA-Dokumentation entnommen werden können.

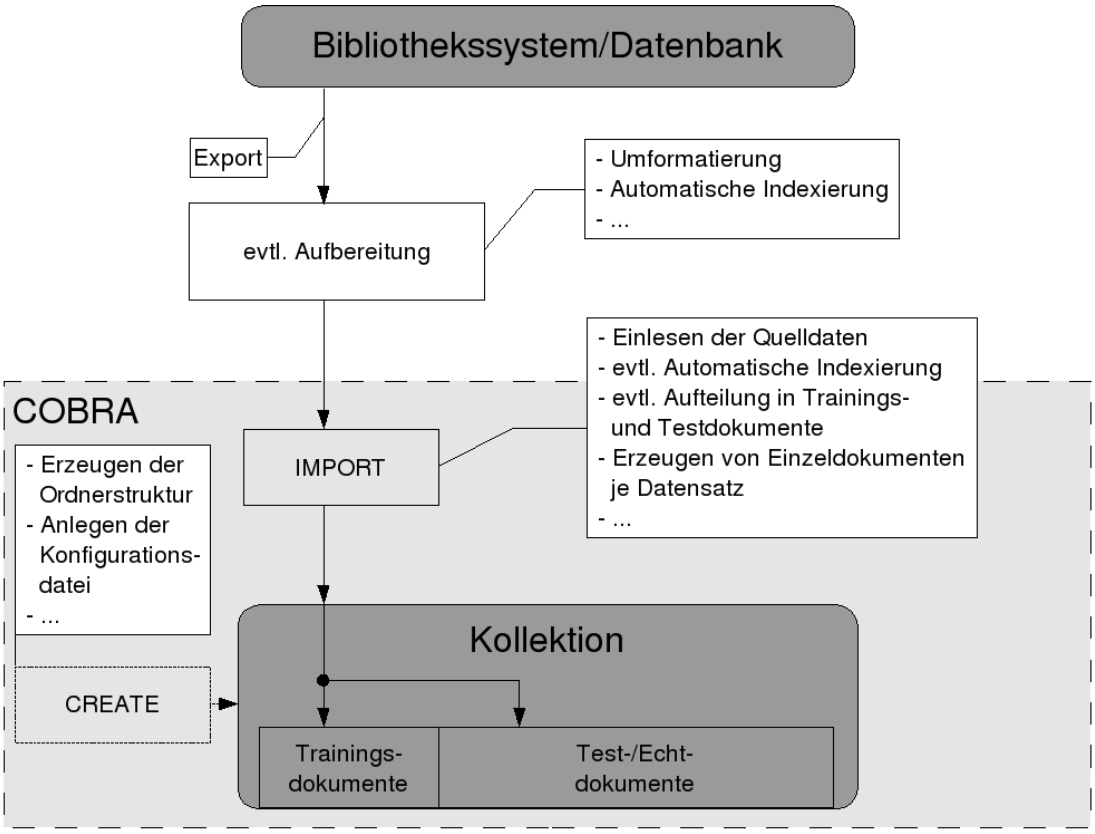
4.1 Datenaufbereitung und -import

Der Import, und in vergleichbarer Weise der Export (s.u.), wird in der Regel den meisten Aufwand mit sich bringen, insbesondere wenn es um die möglichst reibungslose Integration in einen vorhandenen Geschäftsgang geht. Die Seite der Datenhaltungssysteme kann hier nicht Gegenstand der Betrachtung sein, da diese Systeme schlichtweg zu vielfältig sind, aber es ist davon auszugehen, dass jedes System die Möglichkeit zum Ex- und Import bietet, oder zumindest auf Dateisystemebene einen Ansatzpunkt für derartige Operationen. Um diese Daten dann dem Klassifizierer zuführen zu können, müssen sie in ein für diesen geeignetes Format gebracht werden.

Hier setzt COBRA an und erzeugt für jeden zu importierenden Datensatz eine eigene Datei in einem bestimmten von AI::Categorizer erwarteten Verzeichnis. Gleichzeitig sind an dieser Stelle die für die Klassifizierung heranzuziehenden Kategorien des Datenschemas auszuwählen, mit der Konsequenz, dass nur deren Inhalte in die erzeugten Dateien gelangen. Wie bereits angesprochen ist es auch möglich, diese Daten zuvor einer Automatischen Indexierung zuzuführen (siehe Kapitel 3.2). Abbildung 4.2 zeigt eine Detaildarstellung der Importphase.

Des weiteren ist festzulegen, zu welchem Zweck und in welchem Verhältnis die Dateien verwendet werden sollen. D.h., ob sie für das Trainieren des Klassifikators, das Testen desselben oder eben für eine Klassifizierung heranzuziehen sind. In den beiden ersteren Fällen sind bereits im Vorhinein klassifizierte Datensätze vonnöten und es ist möglich, die Trainings- und Testmenge in einem einzigen Importschritt zu erzeugen. Dabei ist es wichtig, den Prozentsatz zu spezifizieren, in welchem die Aufteilung erfolgen soll. Als limitierende Bedingung tritt hier der Umstand in Erscheinung, dass sich einerseits zu wenige Trainingsdokumente negativ auf den Klassifikator auswirken (insbesondere dann, wenn nicht alle Klassen durch eine ausreichende Zahl an Beispieldokumenten vertreten

Abbildung 4.2: Klassifizierungsworkflow mit COBRA: Import



sind) und andererseits zu wenige Testdokumente (ebenfalls insbesondere je Klasse) keine ausreichend zuverlässigen Testergebnisse liefern.[2] Als Sonderfall können aber auch alle Datensätze sowohl für das Training als auch für das Testen herangezogen werden, um einen sog. Klassenintegritätstest durchzuführen.[3]

Bezug zu COBRA und den durchgeführten Experimenten

Die Experimente für diese Arbeit wurden mit folgenden Werten für die o.g. Parameter durchgeführt:

[2] Vgl. Oberhauser 2005, Kap. 2.3.1
[3] Vgl. Oberhauser 2005, S. 85

Typus:	vereinfachtes STN-Format (`Simple::SOLIS`)
Kategorien:	Titel (`TI`), Deskriptoren (`CT`), Abstract (`AB`), Methoden (`ME`)
Indexierer:	Lingo (nur für einen Teil der Experimente herangezogen)
Verhältnis:	10% (d.h. 10% Testdokumente, 90% Trainingsdokumente)

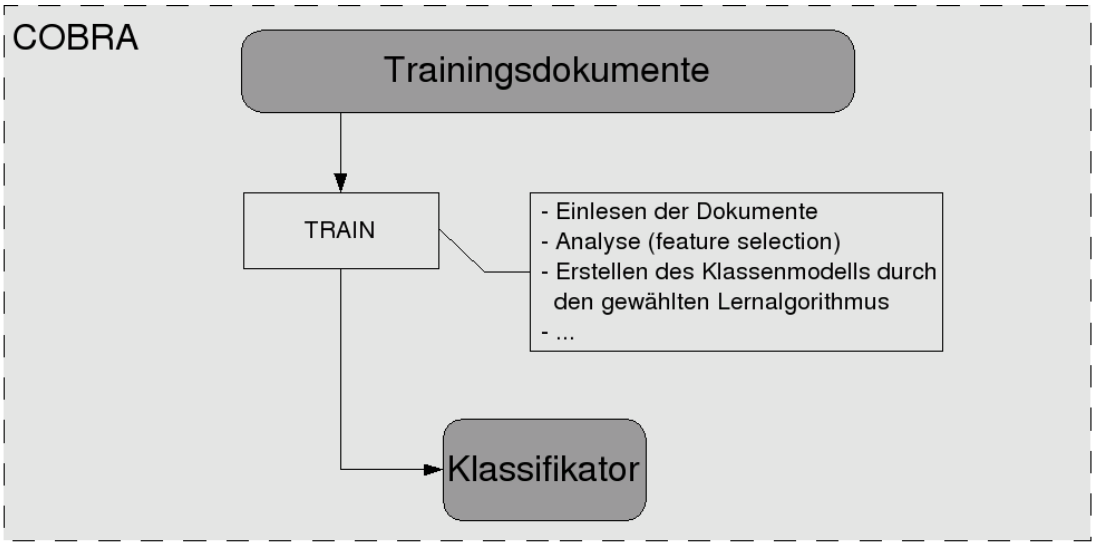
4.2 Training

Das Training anhand bereits klassifizierter Dokumente dient dazu, letztendlich ein Modell des Klassifikationssystems zu erhalten, welches zu entscheiden erlaubt, ob ein zu klassifizierendes Dokument einer bestimmten Klasse zuzuordnen ist oder nicht, wobei dies zumeist graduell in Form eines Zahlenwertes angegeben wird.[4] Hierfür stehen verschiedene Algorithmen zur Verfügung, deren praktischer Unterschied hauptsächlich darin besteht, dass sie verschiedene Ergebnisse liefern und dazu – zum Teil beträchtlich – verschieden Zeit benötigen. AI::Categorizer, und mithin COBRA, unterstützt eine Reihe dieser Algorithmen, welche der aktuellen Dokumentation zu entnehmen sind.

Während der Trainingsphase (vgl. Abbildung 4.3) werden also die importierten Dokumente eingelesen und analysiert, wofür als einzige Optionen der anzuwendende Algorithmus und evtl. dessen spezifische Parameter anzugeben sind. Das Ergebnis ist dann eine interne Repräsentation des besagten Modells, der sog. „Klassifikator“, welche im weiteren Verlauf zum Klassifizieren herangezogen wird. Dieser Klassifikator wird – evtl. nach mehreren Zyklen des abwechselnden Trainierens und Testens, um die Parameter zu optimieren – in der Regel nur einmal erstellt und dann immer wiederverwendet werden. Immer dann allerdings, wenn das zu Grunde liegende Klassifikationssystem verändert wird, muss auch der Klassifikator erneut trainiert werden. Ebenso kann sich die Notwendigkeit zum neuerlichen Trainieren ergeben, wenn plötzlich sehr viele zusätzliche oder in irgendeiner Weise neuartige Dokumente hinzukommen und klassifiziert werden sollen:

[4] Vgl. Oberhauser 2005, Kap. 2.5

Abbildung 4.3: Klassifizierungsworkflow mit COBRA: Training



Ein Klassifikator ist – zum gegenwärtigen Stand der Technik – immer nur (höchstens) so gut wie seine Trainingsmenge.

4.3 Test und Echtbetrieb

Die beiden Schritte Test und „Echtbetrieb“ sind insofern identisch als hier nur die eigentliche Klassifizierung stattfindet. Der einzige Unterschied liegt darin, dass für den Test bereits intellektuell klassifizierte Dokumente erforderlich sind, und dass im Anschluss an den Test dessen statistische Ergebnisse (Vergleich der automatisch zugewiesenen Klassen mit den intellektuell zugewiesenen) ausgegeben werden. Unter Echtbetrieb ist also im Gegensatz zum Test die Klassifizierung neuer, noch unklassifizierter Dokumente zu verstehen.

In beiden Fällen werden die entsprechend importierten Dokumente mit dem zuvor trainierten Klassifikator klassifiziert und es wird eine Liste mit den (gewichteten) Zuord-

nungen der Klassen zu den Dokumenten erstellt. In dieser Phase ist keine weitere Parameterangabe erforderlich.

4.4 Datenexport

Der Export schließlich dient dazu, die erzeugten Klassifizierungsergebnisse zusammen mit den zugehörigen Datensätzen in das Datenhaltungssystem zu integrieren, um sie beispielsweise für Suche oder Navigation im OPAC einsetzen zu können.

Allerdings ist an dieser Stelle auch anzumerken, dass die Ergebnisse der Automatischen Klassifizierung mglw. nicht direkt zu verwenden sind, sondern z.B. nur als Vorschläge für einen menschlichen Klassifizierer dienen (sog. semi-automatisches Klassifizieren).[5]

Ähnlich dem Import (s.o.) ist auch der Export stark an das Datenhaltungssystem gekoppelt und erfordert u.U. eine entsprechende Konvertierung, um die Daten bzw. die Klassifizierungsergebnisse in dieses importieren zu können. Bedauerlicherweise bietet COBRA derzeit noch keine solche Exportfunktionalität. Nichtsdestotrotz ist dies für die Zukunft vorgesehen.

4.5 Zusammenfassung des allgemeinen Ablaufs

Zum besseren Verständnis und der Übersichtlichkeit halber folgt abschließend eine Auflistung der durchzuführenden Schritte, wobei an dieser Stelle weitestgehend von COBRA abstrahiert wird. Darüber hinaus stellt Abbildung 4.4 diesen allgemeinen Ablauf graphisch dar.

[5] Vgl. hierzu auch Oberhauser 2005, Kap. 2.1.4

1. Export der Dokumente aus dem Datenhaltungssystem/Import in das Klassifizierungssystem

2. Training des Klassifikators anhand bereits intellektuell klassifizierter Dokumente

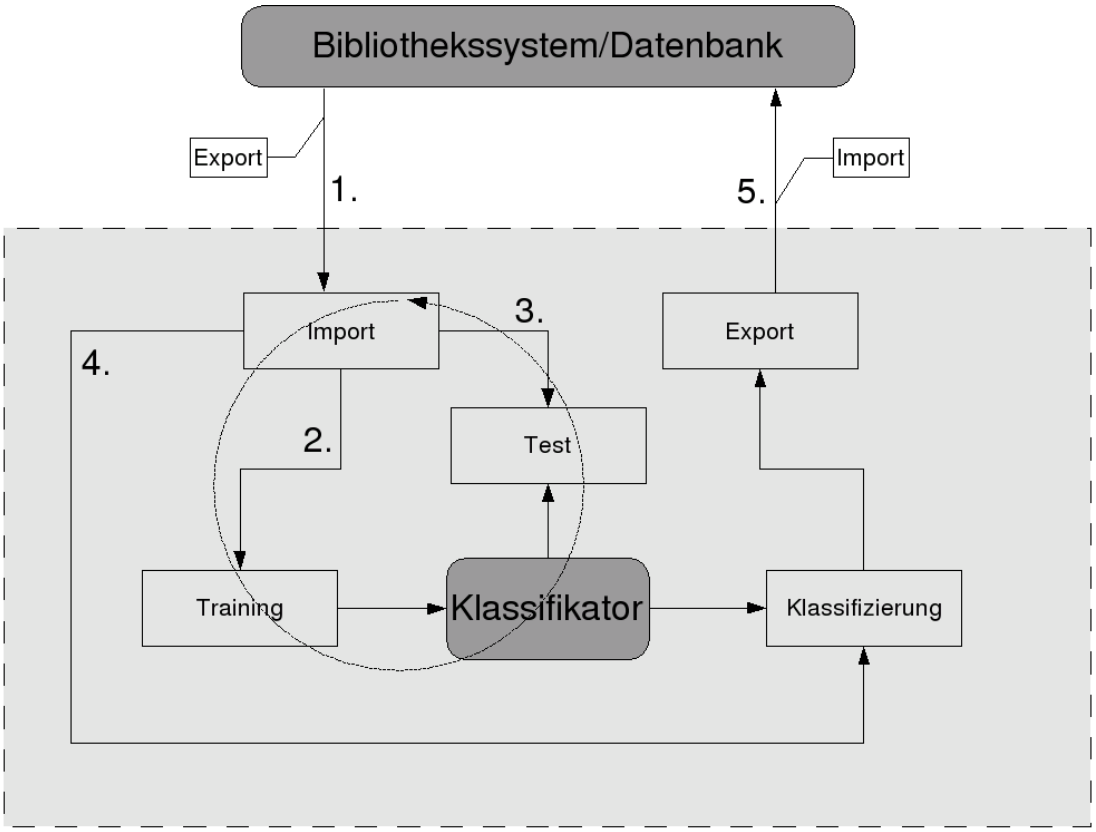
3. Testen des zuvor erstellten Klassifikators wiederum anhand (anderer!) bereits intellektuell klassifizierter Dokumente

   (Schritte 2. und 3. bei Bedarf mit veränderten Parametern wiederholen)

4. Klassifizierung von noch unklassifizierten Dokumenten

5. Export der klassifizierten Dokumente bzw. der Klassifizierungsergebnisse aus dem Klassifizierungssystem/Import in das Datenhaltungssystem

Abbildung 4.4: Übersicht über den allgemeinen Workflow einer Automatischen Klassifizierung



5 Ergebnisse

Wie bereits angedeutet sollen die hier durchgeführten Experimente die Vorgehensweise veranschaulichen und einen ersten Eindruck davon vermitteln, was von einer Automatischen Klassifizierung bibliographischer Daten zu erwarten ist. Für diese Experimente wurden 4.993 deutschsprachige Datensätze aus der sozialwissenschaftlichen Datenbank SOLIS mit COBRA, welches auf dem Perl-Modul AI::Categorizer aufsetzt, klassifiziert.

Die Daten lagen im vereinfachten STN-Format[1] vor und wurden durchgehend in einem Verhältnis von 10/90 in Test- und Trainingsdokumente aufgeteilt. Aus diesen Datensätzen wurden die Kategorien `TI` (Titel), `CT` (Deskriptoren), `AB` (Abstract) und `ME` (Methoden) extrahiert, wobei außerdem eine Serie von Experimenten ohne das Abstract durchgeführt wurde. Darüber hinaus wurde jeweils eine weitere Serie von Experimenten durchgeführt, wo ausschließlich die von der Automatischen Indexierung Lingo[2] erzeugten Indexate verwendet wurden (COBRA-Option `-effect` gleich `replace`).

Hier eine Übersicht über die vier Serien mit Angabe des entsprechenden Befehls[3] (1. für alle verfügbaren Lernalgorithmen, 2.–4. nur für eine Auswahl davon):[4]

[1] Siehe Kapitel 3.1
[2] Vgl. Kapitel 3.2
[3] Ein Backslash (\) bedeutet dabei, dass die Zeile fortgesetzt wird; m.a.W., jeder Befehl ist *in einer Zeile* einzugeben.
[4] Alle erzeugten Daten sind auf der beiliegenden DVD zu finden; näheres ist Anhang A zu entnehmen.

1. wie oben beschrieben (`solis_<learner>`)
   ```
   cobra.pl 1-2 solis_<learner> \
       -d data/solis_sample.txt
   ```

2. wie 1., jedoch ohne Abstract (`solis_<learner>_noAB`)
   ```
   cobra.pl 1-2 solis_<learner>_noAB \
       -u TI,CT,ME \
       -d data/solis_sample.txt
   ```

3. wie 1., jedoch auschließlich die Lingo-Indexate (`solis_<learner>_ai`)
   ```
   cobra.pl 1-2 solis_<learner>_ai \
       -i Lingo \
       -e replace \
       -d data/solis_sample.txt
   ```

4. wie 2., jedoch auschließlich die Lingo-Indexate (`solis_<learner>_ai_noAB`)
   ```
   cobra.pl 1-2 solis_<learner>_ai_noAB \
       -u TI,CT,ME \
       -i Lingo \
       -e replace \
       -d data/solis_sample.txt
   ```

Die zur Verfügung stehenden Algorithmen waren DecisionTree (`dt`), Guesser (`gu`), KNN (k-Nearest Neighbour; `knn`), NaiveBayes (`nb`), Rocchio (`ro`) und SVM (Support Vector Machine; `svm`),[5] wobei aufgrund der Erfahrungen im Projekt „Automatische Klassifizierung von bibliographischen Referenzdaten“ und weiter gestützt durch Studien wie z.B. Yang u. Liu 1999 hauptsächlich DecisionTree, KNN und SVM betrachtet wurden.

Eine wichtige Beobachtung in Bezug auf die Ausführungszeiten[6] sei noch angemerkt: Während bis auf DecisionTree[7] alle anderen Algorithmen innerhalb weniger Minuten, jedenfalls unterhalb einer Stunde, mit Training und Klassifizieren der Testdaten fertig waren, benötigte dieser jeweils zwischen 16 und 48 Stunden für die Abarbeitung der

[5] Für nähere Informationen dazu siehe Oberhauser 2005, Kap. 2.5.2 und die Dokumentation zu AI::Categorizer: `http://search.cpan.org/dist/AI-Categorizer/lib/AI/Categorizer.pm#Machine_Learning_Algorithms` [26. 06. 2006]

[6] Auf einem Athlon XP 1800+ mit 1 GB RAM, unter Linux mit Perl 5.8.1.

[7] AI::DecisionTree 0.08

vier Serien. Zwar lieferte der DecisionTree-Algorithmus im Test durchaus hervorragende Ergebnisse (s.u.) und das eigentliche Klassifizieren geht im Vergleich zum Training auch wesentlich schneller (innerhalb weniger Sekunden), aber dennoch liegt diese Ausführungszeit u.U. (auch je nach Größe des Klassifikationssystems bzw. der Kollektion) jenseits der Praxistauglichkeit.

5.1 Beschreibung der Klassifizierungsergebnisse

Für die Tests werden von AI::Categorizer statistische Werte ermittelt, welche die Übereinstimmung der automatisch zugeteilten Klassen (`zugewiesen=Ja`) mit den ursprünglich intellektuell zugewiesenen (`korrekt=Ja`) ausdrücken. Zur Veranschaulichung dient folgende Kontingenztabelle:

	korrekt=Ja	korrekt=Nein
zugewiesen=Ja	**A**	**B**
zugewiesen=Nein	**C**	**D**

Es folgt die Tabelle der Klassifizierungsergebnisse für die 1. Serie, diejenige für die Serien 2.–4. findet sich weiter unten:

	maR	**maP**	**maF1**	**miR**	**miP**	**miF1**	**Err**
dt	**0.3307**	0.3796	0.3418	**0.3477**	0.4006	0.3723	0.01952
gu	0.05149	0.04636	0.04529	0.02164	0.02284	0.02222	0.03195
knn	0.03836	0.04403	0.03888	*0.0007728*	*1.0000*	*0.001544*	0.01677
nb	0.1354	0.2344	0.1456	0.1940	0.4674	0.2742	0.01723
ro	*1.0000*	0.01678	0.03274	*1.0000*	0.01678	0.03301	0.9832
svm	0.2835	**0.5637**	**0.3502**	0.2952	**0.6773**	**0.4112**	**0.01419**

Zur Erläuterung:[8] Die Kürzel in der linken Spalte entsprechen den o.a. Lernalgorithmen, diejenigen in der ersten Zeile den berechneten statistischen Maßen. Dabei steht `R` für „Recall“, d.h. den Anteil der korrekt zugewiesenen Klassen (**A**) an allen korrekten Klassen (**A**+**C**), `P` für „Precision“, d.h. den Anteil der korrekt zugewiesenen Klassen (**A**) an allen zugewiesenen Klassen (**A**+**B**), und `F1` für den F1-Wert (auch Einheitswert), welcher Precision und Recall kombiniert (harmonisches Mittel): $F1 = 2 * P * R/(P + R)$.[9] `Err` schließlich gibt die Fehlerrate an, d.h. den Anteil der falschen Entscheidungen (also fälschlicherweise zugewiesen (**B**) oder fälschlicherweise nicht zugewiesen (**C**)) an allen Entscheidungen (**A**+**B**+**C**+**D**).[10]

Bei Precision, Recall und F1-Wert ist zu unterscheiden zwischen „macro-averaged“ (`ma`) und „micro-averaged“ (`mi`). Im erstgenannten Fall werden die Maße erst je Klasse berechnet und dann gemittelt, während im zweiten die Entscheidungen über alle Klassen aufaddiert und anschließend gemittelt werden. Dabei werden durch macro-averaging schwach besetzte Klassen überbewertet, durch micro-averaging hingegen stark besetzte.[11]

In beiden Tabellen wurden die besten Werte je Spalte bzw. Gruppe hervorgehoben, während „Ausreißer“ ebenfalls gekennzeichnet und nicht weiter berücksichtigt wurden.[12]

Zwar fällt es schwer, die Performanz der Algorithmen an nur einem Maß festzumachen, da jene je nach den eigenen Bedürfnissen verschiedenen Anforderungen genügen muss,[13] aber es zeigt sich dennoch eine klare Dominanz von DecisionTree und SVM vor allen

[8] Vgl. zu all dem auch Oberhauser 2005, Kap. 2.6.1 oder die Dokumentation zu dem Perl-Modul Statistics::Contingency von Ken Williams, welches intern von AI::Categorizer verwendet wird: `http://search.cpan.org/dist/Statistics-Contingency/Contingency.pm` [26. 06. 2006]

[9] Für alle vorgenannten Maße gilt, dass sie im Bereich von 0 bis 1 liegen, wobei 1 der beste Wert ist.

[10] Für das Fehlermaß gilt, dass es zwar ebenfalls im Bereich von 0 bis 1 liegt, aber dass hier 0 den besten Wert darstellt.

[11] Sebastiani 2002, S. 33

[12] Bspw. ist es für den Vergleich nicht weiter aussagekräftig, wenn ein Algorithmus wie Rocchio, der immer *jede* Klasse zuteilt (wenn auch gewichtet), einen Recall-Wert von 1 erhält.

[13] Bspw. ist in einem vollautomatischen Verfahren, gegenüber einem semi-automatischen, die Fehlerquote entscheidend, während bei letzterem die Präzision zugunsten der Vollständigkeit vernachlässigt werden könnte.

anderen:

$$svm > dt > nb \gg \{gu, knn, ro\}$$

Darüber hinaus fällt bemerkenswerterweise auf, dass der Guesser, welcher die Zuteilungen nur aufgrund der Klassenhäufigkeiten „errät“, besser abschneidet als KNN.

Wie bereits erläutert wurden die Testserien 2.–4. nur für die „vielversprechendsten“ Algorithmen durchgeführt:

	maR	**maP**	**maF1**	**miR**	**miP**	**miF1**	**Err**
dt	0.3307	0.3796	0.3418	0.3477	0.4006	0.3723	0.01952
dt_ai	0.3242	0.3517	0.3211	0.3128	0.3703	0.3391	0.01995
dt_ai_noAB	**0.3693**	**0.4556**	**0.3832**	**0.3749**	**0.4662**	**0.4156**	**0.01758**
dt_noAB	0.3622	0.4191	0.3599	0.3677	0.4419	0.4014	0.01887
knn	0.03836	0.04403	0.03888	*0.0007728*	*1.0000*	*0.001544*	0.01677
knn_ai	0.03165	0.03165	0.03165	*0.0000*	*0.0000*	*0.0000*	**0.01636**
knn_ai_noAB	**0.04575**	**0.1242**	**0.05401**	**0.01836**	**0.9600**	**0.03604**	0.01638
knn_noAB	0.04292	0.08176	0.04659	*0.009373*	*1.0000*	0.01857	0.01704
svm	0.2835	**0.5637**	0.3502	0.2952	**0.6773**	0.4112	**0.01419**
svm_ai	0.2331	0.5039	0.2945	0.2496	0.6300	0.3576	0.01468
svm_ai_noAB	0.3719	0.4956	0.4032	0.3933	0.5054	0.4423	0.01653
svm_noAB	**0.3924**	0.5410	**0.4271**	**0.4088**	0.5521	**0.4698**	0.01588

Hierbei fällt auf, dass die Varianten ohne die Abstracts (`*_noAB`) tendenziell besser abschneiden, obwohl zu erwarten gewesen wäre, dass die Ergebnisse umso besser ausfallen, je mehr Textmaterial für das Training bzw. die Zuteilungsentscheidung zur Verfügung steht.

5.2 Diskussion und Optimierungspotential

Nun stellt sich die Frage, worin die Unterschiede der Ergebnisse sowohl der einzelnen Algorithmen als auch der Zusammensetzungen der Kollektionen begründet liegen und wie sie sich verbessern lassen. Zwar würde dies – neben einer umfangreicheren Datenbasis – eine detailliertere Kenntnis der Lernverfahren erfordern, aber am Beispiel des KNN läßt sich die Problematik relativ leicht aufzeigen: Jeder Algorithmus hat spezifische Parameter, welche seitens AI::Categorizer bzw. der einzelnen Implementierungen der Algorithmen mit Standardwerten vorbelegt sind. Die zwei wichtigsten Parameter von KNN sind der sog. k-Wert, der angibt, wieviele „Nachbarn“ berücksichtigt werden, und der Schwellenwert, welcher darüber entscheidet ab welchem Gewicht eine Klasse einem Dokument zugewiesen wird. Während ersterer mit einem Wert von 20 durchaus sinnvoll erscheint,[14] ist letzterer mit 0.4 vermutlich zu niedrig angesetzt. Daraus resultiert das unerwartet schlechte Abschneiden von KNN.

Zum einen besteht also bei den algorithmen-spezifischen Parametern Optimierungspotential, zum anderen bietet das Textmaterial selbst Möglichkeiten, auf die Güte der Klassifizierungsergebnisse Einfluss zu nehmen. Wenn sich bspw. herausstellen sollte, dass die Ergebnisse ohne Abstracts besser ausfallen, steht für weitere Tests gleich eine viel größere Menge an Dokumenten zur Verfügung, da auf das Vorhandensein von Abstracts keine Rücksicht mehr genommen werden muss.[15] In ähnlicher Weise stellt sich die Frage, welche Auswirkungen das Vorhandensein von Schlagworten u.dgl. hat.

Darüber hinaus bestehen bei der Automatischen Indexierung zahlreiche Möglichkeiten, weitere Testkonfigurationen zu schaffen und deren Auswirkungen zu untersuchen.

Schließlich gilt es zu prüfen, ob das Einbeziehen des Klassifikationssystems (insbesondere

[14] Oberhauser 2005, S. 30

[15] Denkbar wäre auch eine Kombination wie Training mit und Test/Klassifizierung ohne Abstracts u.ä.

eines so vokabularreichen wie dem vorliegenden) für das Training zu einer Verbesserung der Ergebnisse führen kann. Und auch eine Optimierung der Verteilung der Beispieldokumente auf die einzelnen Klassen könnte dazu beitragen.

Wie bereits angemerkt, war der Umfang der hier durchgeführten Tests zu gering, um ausreichend Aussagekraft zu besitzen. Eine möglichst realitätsnahe Beurteilung der Güte von Automatischer Klassifizierung (gegenüber intellektueller oder untereinander) kann hingegen nur durch entsprechend breit angelegte Retrievaltests erzielt werden.[16] Dies ist es, was COBRA letztendlich ermöglichen soll, die einfache Anwendung und Durchführung von Automatischer Klassifizierung mit dem Ziel, einen größeren Erfahrungsschatz zu schaffen und somit der Entwicklung in der bibliothekarischen Sacherschließung neue Impulse zu verleihen.

[16] Zumal auch nicht vergessen werden sollte, dass die intellektuelle Klassifizierung als Status quo zwar zunächst das Maß ist, an dem sich die Automatische Klassifizierung messen lassen muss, aber letztendlich nicht zwangsläufig das Nonplusultra darstellt. Retrievaltests bieten hier ein wesentlich realistischeres Bild, da sie unmittelbar den praktischen Nutzen der Klassifizierung – sei sie nun intellektuell oder automatisch erfolgt – zu beurteilen erlauben.

6 Schlussbetrachtung

6.1 Zusammenfassung und Bewertung

Die vorliegende Arbeit kann als Fortführung der Arbeit von Herrn Dr. Oberhauser[1] angesehen werden, welche ursprünglich im Sommersemester 2004 als Master's Thesis unter Betreuung von Herrn Prof. Gödert am Institut für Informationswissenschaft der Fachhochschule Köln entstanden ist. Während letztere in erster Linie eine (deutschsprachige) theoretische Aufbereitung des Themenfeldes Automatische Klassifizierung lieferte, ist es das Anliegen dieser Arbeit und des im Zuge dessen entwickelten Programms „COBRA – Classification Of Bibliographic Records, Automatic", die praktische Umsetzung – vor allem im Hinblick auf bibliographische Daten – zu ermöglichen.

Es wurde versucht, die Rahmenbedingungen und Einflussfaktoren für das Automatische Klassifizieren von bibliographischen Referenzdaten aufzuzeigen und die einzelnen Schritte der Vorgehensweise zu erläutern. Die Durchführung von Experimenten anhand von sozialwissenschaftlichen Daten und die Präsentation der Ergebnisse sollten auch hier die Vorgehensweise (sowie die Funktionsweise von COBRA) veranschaulichen und einen Eindruck von der erwartbaren Güte vermitteln.

Zwar lassen die betrachteten Klassifizierungsergebnisse keine endgültige Schlussfolge-

[1] Oberhauser 2005

rung zu, aber sie zeigten dennoch, dass zumindest einige Verfahren (namentlich SVM und DecisionTree) potentiell geeignet sein könnten, brauchbare Ergebnisse zu liefern. Hierzu sollte mit dem Perl-Programm COBRA ein Werkzeug an die Hand gegeben werden, welches den Einsatz in der Bibliothekspraxis erlaubt. Zweifelsohne kann dieses zum jetzigen Stand noch nicht in ausreichendem Maße die praxisrelevanten Anforderungen erfüllen, aber zumindest steht es für Tests und prototypische Anwendungen zur Verfügung und kann mit den Anforderungen wachsen.

6.2 Ausblick

Um den Einsatz der Automatischen Klassifizierung in der bibliothekarischen Sacherschließung vorantreiben zu können, ist es erst einmal erforderlich, die Qualität derartiger Klassifizierungsergebnisse in umfassendem Maße zu untersuchen. Die Notwendigkeit für umfangreiche Evaluierungen und Retrievaltests wurde bereits im letzten Kapitel (Abschnitt 5.2) angesprochen. Erst wenn eine solide empirische Basis existiert, kann über den Echtbetrieb in einer Bibliothek oder anderen Informationseinrichtung ernsthaft nachgedacht werden.[2]

Im Hinblick auf COBRA bedeutet dies eine stetige Weiterentwicklung bis hin zur Praxistauglichkeit – denn diese muss freilich erst noch unter Beweis gestellt werden. Überhaupt bleibt abzuwarten, wie sich COBRA in der Handhabung durch Dritte bewähren wird.[3] Außerdem besteht bei den Integrationskomponenten (Import und Export) wie bereits erwähnt einiges an Handlungsbedarf, der sich aber insbesondere durch die Praxisanforderungen erst noch detaillierter herausbilden muss.

[2] Andererseits dürfte aber auch der prototypische Einsatz unter realen Bedingungen wertvolle Erfahrungen hierzu liefern.

[3] Jegliche Rückmeldungen sind erwünscht und herzlichst willkommen!

Schlussendlich bleibt zu wünschen, dass Verfahren zur Automatischen Klassifizierung Eingang finden mögen in die Praxis bibliothekarischer und anderer Informationseinrichtungen, und dass dadurch eine verbesserte Unterstützung der Nutzer bei Suche und Retrieval nach Information ermöglicht und erreicht wird.

Anhang A

Programm und andere Ressourcen

Die Programmumgebung COBRA: Quellcode und Dokumentation

Das Perl-Programm „COBRA – Classification Of Bibliographic Records, Automatic“ befindet sich mit allen zugehörigen Dateien auf der beiliegenden DVD im Verzeichnis `cobra/`. Die Dateien `README` und `INSTALL` geben nähere Auskunft, wie COBRA zu installieren und zu verwenden ist. Im Unterverzeichnis `doc/` befindet sich die Programmdokumentation.

Darüber hinaus ist COBRA über den URL `http://blackwinter.de/da/` verfügbar, wobei allerdings keine Gewähr für eine dauerhafte Verfügbarkeit gegeben werden kann. Ob COBRA in der Zukunft auch über das CPAN[1] zugänglich sein wird, ist noch ungewiss.

Zu dem Status von COBRA zum Zeitpunkt der Fertigstellung dieser Arbeit (Versi-

[1] `http://search.cpan.org/` [26. 06. 2006]

on 0.0.1) sei noch angemerkt: Dieses Programm ist ausdrücklich noch als „premature“ anzusehen! Die wichtigsten Aspekte, die aus Sicht des Autors zum Erreichen eines stabilen Standes noch umzusetzen sind, können der Datei `TODO` entnommen werden.

COBRA kann unter den Bedingungen der GNU General Public License[2] frei verwendet werden. Dies schließt insbesondere die Rechte ein, das Programm für jeden Zweck zu benutzen, den Quellcode zu studieren, Kopien des Programms weiterzuverbreiten sowie das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen.

Diese Arbeit

Ebenso wie COBRA ist diese Arbeit (einschließlich der LaTeX-Quellen) sowohl auf der DVD (im Verzeichnis `da/`) als auch auf `http://blackwinter.de/da/` verfügbar. Zudem wird sie voraussichtlich auch über E-LIS[3] zugänglich sein.

[2] `http://gnu.org/licenses/gpl.html` [26. 06. 2006]
[3] `http://eprints.rclis.org/` [26. 06. 2006]
[4] `http://creativecommons.org/licenses/by-sa/2.0/de/` [26. 06. 2006]

Verwendete Daten und Ergebnislisten

Die für die durchgeführten Experimente verwendeten Daten sind auf der DVD unterhalb des COBRA-Verzeichnisses zu finden, dsgl. die erhaltenen Ergebnisse. Ebenso liegen sie unter der bereits genannten Web-Adresse `http://blackwinter.de/da/`.

Das Informationszentrum Sozialwissenschaften[5] gestattet die Verwendung von Teilen der SOLIS-Datenbank zu Forschungszwecken.[6]

Sonstiges

Des Weiteren befinden sich auf der DVD bzw. unter der Adresse `http://blackwinter.de/da/` abrufbar:

- Die verwendeten Quellen, sofern elektronisch vorliegend: `lit/`
- Abbildungen und Diagramme: `da/files/`
- Die aktuelle Lingo-Version (1.6.1),[7] zzgl. der im Rahmen dieser Arbeit eingesetzten, vom Verfasser leicht modifizierten Version (1.6.1-jw; die Modifikationen sind der Datei `diff-jw` zu entnehmen), einschließlich der verwendeten Wörterbücher: `lingo/`

[5] `http://www.gesis.org/IZ/` [26. 06. 2006]
[6] Persönliche E-Mail von Dr. Maximilian Stempfhuber, Stellvertretender Direktor und Abteilungsleiter Forschung & Entwicklung, an den Verfasser, 13. 05. 2006.
[7] `http://www.lex-lingo.de/` [26. 06. 2006]

Literaturverzeichnis

Hinweis: Alle Weblinks wurden zuletzt überprüft am 26. 06. 2006.

[Lepsky 1994] Lepsky, Klaus: *Maschinelle Indexierung von Titelaufnahmen zur Verbesserung der sachlichen Erschließung in Online–Publikumskatalogen.* Köln : Greven, 1994 (Kölner Arbeiten zum Bibliotheks- und Dokumentationswesen ; Bd. 18). – ISBN 3–7743–0572–2

[Lepsky u. Vorhauer 2006] Lepsky, Klaus ; Vorhauer, John: Lingo : ein open source System für die Automatische Indexierung des Deutschen. In: *ABI–Technik* (2006), Nr. 1, S. 18–28. `http://www.lex-lingo.de/downloads/lingo-uebersichtsartikel.pdf`

[Mitchell 1997] Mitchell, Tom: *Machine Learning.* New York : McGraw-Hill, 1997. – ISBN 0–07–115467–1

[Nohr 2003] Nohr, Holger: *Grundlagen der automatischen Indexierung : ein Lehrbuch.* Berlin : Logos–Verl., 2003. – ISBN 3–8325–0121–5

[Oberhauser 2005] Oberhauser, Otto: *Automatisches Klassifizieren : Entwicklungsstand – Methodik – Anwendungsbereiche.* Frankfurt am Main : Lang, 2005 (Europäische Hochschulschriften : Reihe 41, Informatik ; Bd. 43). – ISBN 3–631–53684–4

[Sebastiani 1999] SEBASTIANI, Fabrizio: A Tutorial on Automated Text Categorisation. In: *Proceedings of ASAI–99, 1st Argentinian Symposium on Artificial Intelligence* (1999), S. 7–35. http://www.math.unipd.it/~fabseb60/Publications/ASAI99.pdf

[Sebastiani 2002] SEBASTIANI, Fabrizio: Machine Learning in Automated Text Categorization. In: *ACM Computing Surveys* 34 (2002), Nr. 1, S. 1–47. http://www.math.unipd.it/~fabseb60/Publications/ACMCS02.pdf

[Sebastiani 2006] SEBASTIANI, Fabrizio: Classification of text, automatic. In: BROWN, Keith (Hrsg.): *The Encyclopedia of Language and Linguistics* Bd. 2. 2. Aufl. Amsterdam : Elsevier, 2006, S. 457–463. http://www.math.unipd.it/~fabseb60/Publications/ELL06.pdf

[Yang u. Liu 1999] YANG, Yiming ; LIU, Xin: A re–examination of text categorization methods. In: *22nd Annual International SIGIR* (1999), S. 42–49. http://nyc.lti.cs.cmu.edu/yiming/Publications/sigir99.ps.gz

Eidesstattliche Erklärung

Hiermit versichere ich, die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

Hürth, den

Jens Wille