Received March 2004
Revised September 2004
Accepted January 2005

# Term conflation methods in information retrieval

## Non-linguistic and linguistic approaches

Carmen Galvez, Félix de Moya-Anegón and Víctor H. Solana
*Department of Information Science, University of Granada, Granada, Spain*

### Abstract

**Purpose** – To propose a categorization of the different conflation procedures at the two basic approaches, non-linguistic and linguistic techniques, and to justify the application of normalization methods within the framework of linguistic techniques.

**Design/methodology/approach** – Presents a range of term conflation methods, that can be used in information retrieval. The uniterm and multiterm variants can be considered equivalent units for the purposes of automatic indexing. Stemming algorithms, segmentation rules, association measures and clustering techniques are well evaluated non-linguistic methods, and experiments with these techniques show a wide variety of results. Alternatively, the lemmatisation and the use of syntactic pattern-matching, through equivalence relations represented in finite-state transducers (FST), are emerging methods for the recognition and standardization of terms.

**Findings** – The survey attempts to point out the positive and negative effects of the linguistic approach and its potential as a term conflation method.

**Originality/value** – Outlines the importance of FSTs for the normalization of term variants.

**Keywords** Information retrieval, Document management, Indexing, Variance reduction

**Paper type** Conceptual paper

### Introduction

In many information retrieval systems (IRS), the documents are indexed by uniterms. However, uniterms may result ambiguous, and therefore unable to discriminate only the pertinent information. One solution to this problem is to work with multiterms (multi-word terms or phrases) often obtained through statistical methods. The traditional IRS approach is based on this type of automatic indexing technique for representing documentary contents (Salton, 1980, 1989; Croft *et al.*, 1991; Frakes and Baeza-Yates, 1992).

The concepts behind such terms can be manifested in different forms, known as *linguistic variants*. The variants are defined as a text occurrence that is conceptually related to an original term. In order to avoid the loss of relevant documents, an IRS recognizes and groups variants by means of so-called conflation methods, or term normalization methods. The process of conflation may involve linguistic techniques such as the segmentation of words and the elimination of affixes, or lexical searches through thesauri. The latter is concerned with the recognition of semantic variants.

The grouping of morphological variants would increase average recall, while the identification and grouping of syntactic variants is determinant in increasing the accuracy of retrieval. One study about the problems involved in using linguistic variants in IRS can be found in Sparck Jones and Tait (1984).

Journal of Documentation
Vol. 61 No. 4, 2005
pp. 520-547

0022-0418
DOI 10.1108/00220410510607507

The application of conflation techniques to single-word terms is a way of considering the different lexical variants as equivalent units for retrieval purposes. One of the most widely used non-linguistic techniques is that of stemming algorithms, through which the inflectional and derivational variants are reduced to one canonical form. Stemming or suffix stripping uses a list of frequent suffixes to conflate words to their stem or base form. Two well known stemming algorithms for English are the Lovins (1968) and the Porter (1980).

Another means of dealing with language variability through linguistic methods is the fusion of lexical variants into lemmas, defined as a set of terms with the same stem and, optionally, belonging to the same syntactic category. The process of lemmatization, or morphological analysis of the variants and their reduction to controlled forms, relies on lexical information stored in electronic dictionaries or lexicons. One such example is the morphological analyzer developed by Karttunen (1983).

In addition to these approaches, it is possible to group multi-word terms within a context, assigning specific indicators of relationship geared to connect different identifiers, so that noun phrases (NPs) can be built (Salton and McGill, 1983). NPs are made up of two or more consecutive units, and the relationships between or among these units are interpreted and codified as endocentric constructions, or modifier-head-structures (Harris, 1951). When we deal with single-word terms, the content identifiers are known as indexing terms, keywords or descriptors, and they are represented by uniterms. Uniterms may on occasion be combined or coordinated in the actual formulation of the search. When multi-word terms or NPs are used for indexing purposes, they can include articles, nouns, adjectives or different indicators of relationship, all parts of a process known as pre-coordination (Salton and McGill, 1983). In indexing multi-word terms, most extraction systems employ part-of-speech (POS) taggers, which reflect the syntactic role of a word in a sentence, then gather together the words that are components of that NP (Church, 1988; Brill, 1993; Voutilainen, 1997; Tolle and Chen, 2000).

When conflation algorithms are applied to multi-word terms, the different variants are grouped according to two general approaches: term co-occurrence and matching syntactic patterns. The systems that use co-occurrence techniques make term associations through different coefficients of similarity. The systems that match syntactic patterns carry out a surface linguistic analysis of certain segments or textual fragments. In addition to the surface analysis and the analysis of fragments from the corpus, many systems effectuate a POS category disambiguation process (Kupiec, 1993). The syntactic variants identified through these methods can be grouped, finally, in canonical syntactic structures (Schwarz, 1990; Sheridan and Smeaton, 1992; Smadja, 1993; Strzalkowski, 1996). The problems that linguistically based NP in IRS have are, according to Schwarz (1990): NP recognition, selection, normalization, matching, and ranking.

The recognition and standardization of linguistic structures in IRS is an area pertaining to natural language processing (NLP). Within the NLP understanding of the mathematical modeling of language, there are two clearly distinguished conceptions: *symbolic models* and *probabilistic* or *stochastic models*. These models can be traced back to the Turing Machine (Turing, 1936); the contribution of Klenee (1956) regarding finite-state mechanisms and regular expressions; and to the work by Shannon

and Weaver (1949) on the application of the probabilistic processes to finite automatas, incorporating Markov Chains (Kemeny and Snell, 1976). Chomsky was the first to consider automatas as mechanisms characterizing the structures of language through grammars (Chomsky, 1957), thereby setting the foundations for the theory of formal languages. Finite-state mechanisms are efficient for many aspects of NLP including morphology (Koskenniemi, 1983) and parsing (Abney, 1991; Roche, 1999).

The extreme complexity of NLP and the necessity of a deep knowledge about language itself become obstacles for IRS. To this we should add that there is no generally accepted retrieval model making use of compound terms obtained using linguistic techniques, another major obstacle in the overall viability of NLP in IR (Strzalkowski *et al.*, 1999). Nonetheless, new research proposes combining linguistic techniques with statistical techniques (Feng and Croft, 2001).

The present paper focuses on the initial stage of automatic indexing in natural language – that is, on the process of algorithmically examining the indexing terms to generate and control the units that will then be incorporated as potential entries to the search file. The recognition and grouping of lexical and syntactic variants can thus be considered a process of standardization; when a term does not appear in a normalized form, it is replaced with the canonical form. Along these lines, we will review the most relevant techniques for merging variants, departing from the premise that term conflation can be considered as a normalizing method, its function being the standardization of term variants.

## The problem of term variants

The objective of IRS consists of retrieving, from amongst a collection of documents, those that respond to an informational need, and to reorganize these documents according to a factor of relevance. This process normally involves statistical methods in charge of selecting the most appropriate terms for representing documental contents, and an inverse index file that accesses the documents containing these terms (Salton and McGill, 1983). The relationship of pertinence between queries and documents is established by the number of terms they have in common. For this reason the queries and documents are represented as sets of characteristics or indexing terms, which can be derived directly or indirectly from the text using either a thesaurus or a manual or automatic indexing procedure.

Matching query terms to documents involves a number of advanced retrieval techniques, and one problem that has not yet been solved is the inadequate representation of the two (Strzalkowski *et al.*, 1999). At the root of this problem is the great variability of the lexical, syntactic and morphological features of a term, variants that cannot be recognized by simple string-matching algorithms without some sort of NLP (Hull, 1996). It is generally agreed that NLP techniques could improve IRS yields; yet it is still not clear exactly how we might incorporate the advancements of computational linguistics into retrieval systems.

During the first stage of automatic indexing in natural language we encounter a tremendous number of variants gathered up by the indexing terms. The variants can be used to extract information in the textual databases (Jacquemin and Tzoukermann, 1999). In the design of a term extraction system, single-word terms are generally polysemic and multi-word terms have a phrase structure that is prone to variations (Savary and Jacquemin, 2003). Arampatzis *et al.* (1998) identify three main types of variations.

(1) *Morphological variation* linked to the internal structure of words, by virtue of which a term can appear in different forms. For instance, "connect", "connected", "connecting", "connection" are reduced to "connect" which is considered to be identical for all these morphologically and conceptually related terms.

(2) *Lexico-semantic variation* linked to the semantic proximity of the words, so that different terms can represent the same meaning, and multiple meanings can be represented by the same term: "anoxaemia", "anoxemia" and "breathing problems" are reduced to "breathing disorders".

(3) *Syntactic variation* linked to the structure of the multi-word terms, where alternative syntactic structures are reduced to a canonical syntactic structure. Constructions that are structurally distinct but semantically equivalent, such as "consideration of these domain properties" and "considering certain domain properties", are conflated to the single structure "considering domain properties".

In most cases, the variants are considered semantically similar units that can be treated as equivalents in IRS (Hull, 1996). To arrive at these equivalencies, conflation methods of variants are used, grouping the terms that refer to equivalent concepts. The most readily used procedures are the reduction of morphological variation of single-word terms by means of stemming algorithms, and lexico-semantic variation reduction methods with lexical lookup, or a thesaurus search (Paice, 1996). Semantic variation is dealt with in query expansion with semantically related terms, while matching is based on word-to-word semantic similarity measures (Arampatzis *et al.*, 2000). The problems involved in fusing the lexico-semantic variants remain beyond the scope of the present review.

## Term conflation methods: non-linguistic vs linguistic approaches

In this work, we propose the categorization of term conflation methods according to two extensive branches: non-linguistic and linguistic approaches. The fundamental difference between one and another depends on the criterion behind the application of NLP tools. The conflation methods within the non-linguistic approach do not apply NLP techniques, while the ones that are found in the linguistic approach do apply them. We should point out that although stemming algorithms are based on linguistic studies and on word morphology, they do not utilize methods pertaining to NLP. Figure 1 shows a concise classification of term conflation methods.

Stemming algorithms, which eliminate all affixes, give good results for the conflation and normalization of uniterm variants (Porter, 1980; Frakes, 1992). Within this group, the most effective are the longest match algorithms. They conflate morphologically similar terms into a single term without performing a complete morphological analysis. The resulting stems are often not legitimate linguistic units and they do not lend themselves to other sorts of processing such as syntactic parsing, because non-linguistic techniques do not carry out an authentic morphological analysis, nor assign POS tags. Likewise, the units left after affix elimination can hardly be used for other IR purposes, such as interactive techniques, which require user input, to select terms for a possible

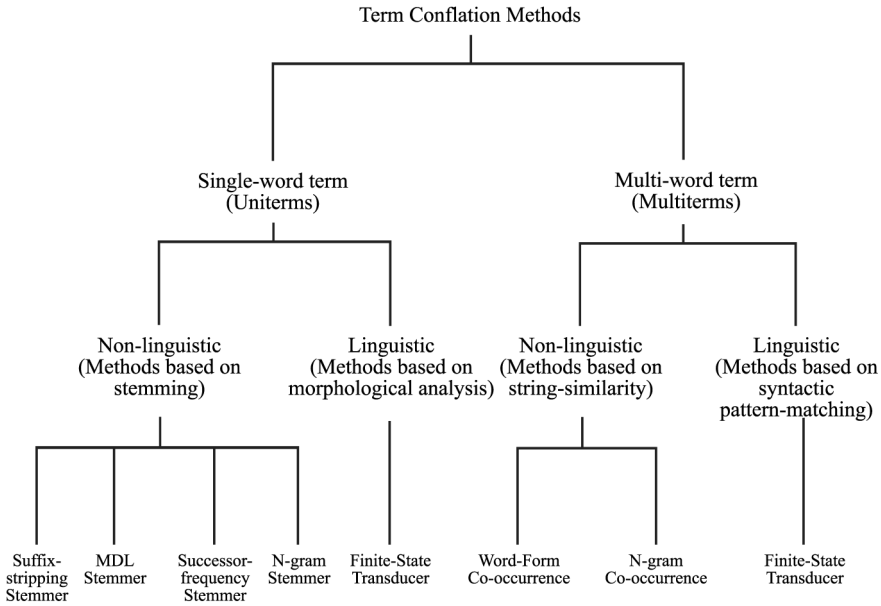



**Figure 1.**
Classification of term conflation methods in IR

query expansion. The solution to this problem resides in doing a full morphological analysis.

In a recent article, Goldsmith (2001) argues that automatic morphological analysis can be divided into four major approaches. The first approach, based on the work by Harris (1955) and further developed by Hafer and Weiss (1974), gives a goodness of-break between letters that can be measured by the successor frequency there, compared to the successor frequency of the letters on either side. The second approach seeks to identify $n$-grams, which are likely to be morpheme-internal. A $n$-gram is a sub string of a word, where $n$ is the number of characters in the sub string, typical values for $n$ being bigrams ($n = 2$) or trigrams ($n = 3$). The third approach focuses on the discovery of patterns of phonological and morphological relationships between pairs of words: a base form and an inflected form. The fourth approach focuses on unsupervised learning techniques, yielding a partition of stems and affixes, segmenting longer strings into smaller units (Kazakov, 1997; Kazakov and Manandhar, 2001).

Techniques that perform an authentic morphological analysis (the third approach above), supply linguistically correct units or lemmas. The linguistic methods used to obtain them are called lemmatization techniques, and they rely on entries to dictionaries or lexicons able to represent finite-state mechanisms. While the acquisition of linguistically correct units may seem irrelevant for information retrieval, they are very useful in the later recognition of NP. All this leads us to a distinction of two basic approaches for conflating uniterm variants.

(1) *Non-linguistic techniques*. Stemming methods consisting mainly of suffix stripping, stem-suffix segmentation rules, similarity measures and clustering techniques.

(2) *Linguistic techniques*. Lemmatisation methods consisting of morphological analysis. That is, term conflation based on the regular relations, or equivalence relations, between inflectional forms and canonical forms, represented in finite-state transducers (FST).

Stemming and lemmatization methods are applied when the terms are morphologically similar. But when the similarity is semantic, lexical search methods are used. To reduce semantic variation, most systems resort to lexical lookup, with dictionaries or thesauri to relate two words that are completely different in form (Paice, 1996). The two procedures are complementary in that stemming checks graphic similarities to infer lexical proximity, whereas lexical lookup refers to terminographic data with links to synonyms (Jacquemin and Tzoukermann, 1999).

The techniques based on single-word terms assume that terms are independent, but this is not true in many cases. Most IRS uses these models, in which the content of each document is represented by a non-structured collection of uniterms (stems or lemmas) without including any type of relationship. The lack of term interrelation translates as statistical independence, and this results in inexact representations that reduce the effectiveness of the IRS.

An appropriate procedure for indexing would be to identify multi-word terms or meaningful phrases, and represent important concepts in the database domain (Strzalkowski *et al.*, 1999). One of the first IRS to use phrase indexing was the SMART system (Salton, 1980; Buckley *et al.*, 1995). Another processing mode for NP is the IRENA system (Information Retrieval Engine Based Natural Language Analysis) (Arampatzis *et al.*, 1998). It assigns to any NP an equivalent structure that consists of nucleus and modifiers, Phrase Frame = [head, modifier]. In a similar model to the previous developed by Strzalkowski *et al.* (1999) the phrase structures are reduced to the normalized string: head + modifier pairs stream. Under the CLARIT system (Evans *et al.*, 1996), the control of syntactic structures is based on the generation of a lexicon phrasal. A combination of statistical and linguistic methods is present in the XTRACT system (Smadja, 1993), based on collocations, or cohesive word clusters. A natural processor for the normalization of term occurrences is the automatic indexing tool FASTR (Jacquemin, 2001). The linguistic knowledge used by FASTR is divided into two databases: a grammar of term rules which represents the syntactic structure of the term, generally a noun phrase, and a met grammar used to transform the term rules into term variant rules. As NLP tools, FASTR use programs for POS tagging and morphological analysis using finite-state techniques and a library of finite-state transducer (FST) developed at Bell Laboratories by Mohri and Sproat (1996).

All this leads us to a distinction of two fundamental approaches for conflating multiterm variants:

(1) *Non-linguistic techniques*. Statistical methods based on the computation of similarity coefficients, association measures and clustering techniques, by means of word and *n*-gram co-occurrence.

(2) *Linguistic techniques*. Syntactic methods based on syntactic pattern-matching according to Local Grammars, represented in finite-state automata (FSA), and pattern conflation through regular relations, or equivalence relations, established between syntactic structure variants and canonical syntactic structures, represented in FST.

## Aims and objectives

Because our purpose is to justify a specific approach such as the application of linguistic techniques within the more general framework of the conflation methods used in IR for the English language, we do not attempt to analyze in-depth all the conflation methods possible nor discuss how well those methods are used. Moreover, not all the methods presented in this work have been properly evaluated to date. Another reason for exploring this area is to arrive at an idea of the potential of NLP techniques in IR research, in view of the failures inherent in the use of non-linguistic methods. Experiments with uniterm and multiterm conflation show a wide variety of results, also depending on the language involved.

Conflating methods have essentially been developed for English because it is the predominant language in IR experiments. However, with a view to the reduction of uniterm variants, English features a relatively weak morphology and therefore linguistic techniques are not necessarily the most suitable ones. To the contrary, because English relies largely on the combination of terms, the linguisitc techniques would indeed be more effective in merging multiterm variants.

Some studies have found that indexing by the stem does not substantially improve the efficacy of retrieval, at least not in the English language (Harman, 1991). This author concludes that the use of a stemmer in the query is intuitive to many users, and reduces the number of terms decreasing the size of the index files, but produces too many non-relevant documents. Meanwhile, experiments by Popovic and Willett (1992) show significant improvement in precision when languages with a more complex inflectional morphology than English are used. They argue that suffix stripping would be effective for languages such as Slovene, and conclude that the effectiveness of a stemmer is a function of the morphological complexity of the language in the document set (Popovic and Willett, 1992). Hull (1996) evaluates the performance of five different stemming algorithms (S-stemmer, Lovins stemmer, Porter stemmer, xerox inflectional stemmer, and xerox derivational stemmer). On the basis of TREC test collection results, he concludes that:

(1) some form of stemming is almost always beneficial,

(2) important factors are the language, document length, and evaluation measures, and

(3) linguistic approaches based solely on a lexicon cannot correctly stem words not contained in the lexicon.

For English, the results of Hull (1996) show that there is no significant difference between suffix-stripping stemmers and the techniques based on morphological analysis through finite-state machines. Nevertheless, for languages with strong morphology, such as Finnish, the morphological analyzers improve average recall in IR (Koskenniemi, 1996).

On the other hand, the multiterms that represent concepts are included among what are known as *complex descriptors*. Fagan (1989) suggests two types of relationships: *syntactic* and *semantic*. First, the syntactic relationships depend on the grammatical structure of these same terms and are represented in phrases. The syntactic relationships are of a syntagmatic type, allowing the reduction of terms used in document representation, and their contribution in the IRS is to increase average precision. Second, the semantic relationships depend on the inherent meaning of the terms involved and are represented in the classes of a thesaurus. The semantic relationships are of a paradigmatic type, allowing us to broaden the terms used in the representation of the documents, and their purpose in the retrieval systems is to increase average recall.

Previous work demonstrates that statistical methods are more effective than linguistic methods in identifying meaningful phrases (Fagan, 1989). Nevertheless, other studies with linguistic approximations to NP recognition – in contrast to the classic phrase construction methods used in IR such as SMART – have pointed to improved retrieval (Hull *et al.*, 1996). Experiments presented in the Sixth Text REtrieval Conference (TREC-6) (Harman, 1997) propose the combination of phrases to improve the efficacy of the retrieval systems, although they do not obtain the level of success desired. The whole group of constituent terms can be substituted by a component of the NP that is called the nucleus, corresponding to the head of the construction. The key is to identify the nucleus and to distinguish which are the satellite elements that modify it. These two components, taken together, form multi-word terms that refer to more specific concepts such as in the bi-member NP "document clustering", "Internet browsing", or "digital libraries", in which the first element modifies the second, and therefore it is important to identify the type of relationship that the terms maintain.

The purely quantitative methods for generating phrase identifiers calculate the statistical association of terms, or co-occurrence of terms, but they are not able to identify the type of modifying relationship. In contrast, linguistic methods may be applied to identify a modifier-head-structure through pattern matching. The patterns are described using expressions that are transferred to finite-state machines, a complex procedure that requires the construction of grammars restricted to NP structures, the use of POS taggers and the development of tools for disambiguation. The efficacy of syntactic pattern-matching in IR resides in the capacity of the system to recognize the local syntactic constructs in specific domains.

Again, we must insist on the influence of language on the results of term conflation. The complexity of terms varies along with the inflectional structure of a language. One interesting study about the morphological phenomena in IRS can be found in Pirkola (2001). Roughly speaking, synthetic languages, including French, Spanish, Italian and the other romance languages, require term inflection to indicate term function in the sentence. Yet analytic languages such as English and German rely on the placement or the combination of terms to indicate their function in the sentence. The synthetic languages have many morphologic variants of single-word terms, whereas the analytic languages have many syntactic variants of multi-word terms. Further study should help clarify the positive and negative end effects of these factors on retrieval effectiveness.

## Conflation methods for normalizing uniterm variants

The procedures or programs for the reduction of variants of single-word terms are called *stemmer programs*, when this process involves non-linguistic techniques, or stemming algorithms (such as rules for reducing a family of words to a common root and similarity measures), and *lemmatization programs*, when this process involves linguistic techniques, or lemmatization algorithms, such as morphological analysis for reducing a family of words to a lemma through regular relations compiled in FSTs.

The non-linguistic techniques are very diverse, and we should begin with a distinction between manual methods and automatic methods. The latter, according to Frakes (1992), include: affix removal, successor variety, *n*-gram matching, and table lookup. A stemming algorithm strips all the words that share a canonical form, represented by a stem, normally eliminating all derivational and inflectional affixes (Lovins, 1968). Conflation with stemming techniques entails removing from a term the longest affixes, according to a set of rules, and repeating the process until no more characters can be eliminated. For this reason, these stemming algorithms are also called longest match algorithms. Xu and Croft (1988) argue that the errors often arising are:

(1) the units obtained are not linguistically correct, and this reduces the level of comprehension of the indexes,

(2) the elimination of fewer suffixes than should be the case, making fusion or conflation of related terms impossible, produces errors of under stemming, and

(3) the elimination of too many suffixes causes the linking of unrelated terms, or overstemming.

These obstacles can be overcome by means of lemmatization algorithms, which allow the reduction to one single form of terms that share a common stem, the same syntactic POS category, and the same meaning. Affixes are eliminated with reference to the entries of a dictionary, configured as a lexical database that serves to carry out a lexical analysis of the input terms and relate them with a canonical form, represented by a lemma. However, there are drawbacks:

(1) the creation of indexes by lexical analysis is very time-consuming, and

(2) the irregularities in inflectional forms often lead to non-matches between inflected forms and the canonical forms stored in the dictionary, thus requiring a complex series of transformations.

One advantage of this method is that the units obtained are linguistically correct and, depending on the lemmatizing program used, can each be assigned to a syntactic category. This is an essential pre-requisite for any subsequent syntactic processing. The dictionary lemmas, then, would be inflected forms of nouns reduced to the singular, or inflected verbs reduced to the infinitive.

### *Suffix-stripping stemmer*

Most affix removal algorithms were developed for the English language, though they can be found for other specific languages, such as Slovene (Popovic and Willett, 1992), French (Savoy, 1993, 1999), Dutch (Kraaij and Pohlmann, 1994, 1995), Latin (Schinke *et al.*, 1996), Malaysian (Ahmad *et al.*, 1996), Greek (Kalamboukis, 1995) and Arabian (Abu-Salem *et al.*, 1999). Meanwhile, Porter's small string processing language,

SNOWBALL, allows the design of stemming algorithms through simple scripts for their application to IR. SNOWBALL stemmers have been implemented with French, Spanish, Italian, Portuguese, German, Norwegian, Swedish, and other languages.

The algorithms best known for the English language are those of Lovins (1968), Dawson (1974), Porter (1980), and Paice (1990). The most aggressive is the Lovins algorithm, according to a comparative study of three models (the S-stemmer, the Lovins stemmer and the Porter stemmer) by Harman (1991). With some models, incorrect stems may be obtained, as in many cases the rule to be applied is not clearly specified. To solve this problem, the Lovins stemmer checks the longest match taken from an extensive list of suffixes, whereas the Porter stemmer is based on an algorithm with a very limited number of suffixes and a few rewriting rules that take into account the context of appearance of the suffixes to be eliminated.

Although the Porter stemmer increases average recall (Porter, 1980), and can be successfully adapted to languages other than English, it is difficult to understand and modify, it may incur in errors, due to excessive conflation or else a lack of conflation and the stems it produces stems are not real words, and are hard for the non-expert user to interpret (Xu and Croft, 1998). For example, the Porter's stemmer groups "study" and "studies" under "studi" (a linguistically ambiguous term). For the purpose of retrieval, it seems reasonable to suppose that a query related to "study", would give documents in which the stem "studi" appears, and enhance retrieval effectiveness despite using a stem that is not an authentic word. At the same time, stemming algorithms can make incorrect groupings. Such problems arise because most stemmers operate without a lexicon and they ignore the meaning of the terms (Krovetz, 1993). For instance, the Porter stemmer groups "general", "generous", "generation", and "generic" under the same stem, while terms like "recognize" and "recognition" would not be grouped (Hull, 1996).

To solve this problem, conflation with the Krovetz stemmer or KSTEM relies on automatized dictionaries and well-defined rules for inflectional and derivational morphology. Although the Krovetz stemmer resolves some conflation errors, it does not ensure better results than the Porter stemmer, in fact, it depends too heavily on dictionary entries for reference, and conflation is therefore too conservative (Xu and Croft, 1998).

*Minimal description length (MDL) stemmer*

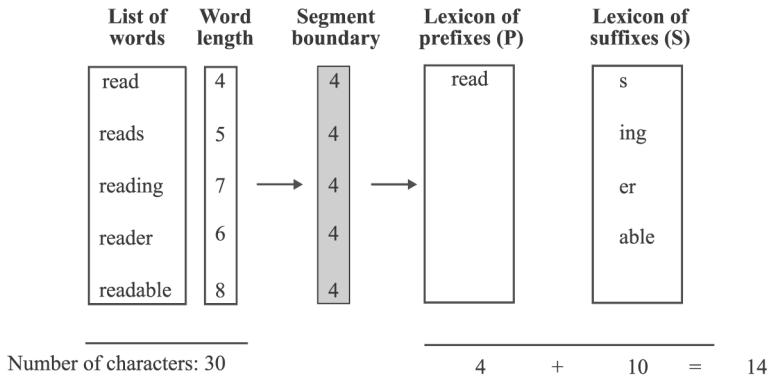
Meanwhile, Kazakov and Manandhar (2001) work with morphological analysis: by counting the number of letters in two lexicons of stems and suffixes, and dividing by the number of letters in a list of words, they applied a uniterm genetic algorithm (GA) (Goldberg, 1989) to explore the optimal space of the segmentation for analysis. Before segmenting a list of words, the morpheme boundaries for each are represented by a vector of integers that indicate where the morphological split should be. The list of words and the vector, however, introduce a representational bias known as the Naïve Theory of Morphology (NTM) (Kazakov, 1997). This theory serves as the basis for two lexicons where prefixes (P), or stems, and suffixes (S) are enumerated without repetition (Figure 2).

Kazakov (1997) notes that the quality of the theory will be estimated by the number of characters $N$ that both lexicons contain, the upper bound $N_{\max}$ of that measure being given by the number of characters $W$ in the word list. His formula is expressed as "among a set of naïve theories of word morphology, select the one with the lowest number of characters in the corresponding pair of lexicons ($P + S = N$) : the smaller that number, the better the theory". Yet as the NTM bias only orders hypothetical

segmentations for a given list of word, it must then be associated with a search algorithm to search the space of plausible segmentations and use a GA to find the best cut in each word (Kazakov and Manandhar, 2001). The search for a theory minimizing $N$ can be seen as a segmentation task or minimal description length (MDL) as described by Brent *et al.* (1995). Briefly, MDL framework is based essentially on that it recommends choosing the hypothesis that minimizes the sum of the description length of the hypothesis (Mitchell, 1997). The MDL induction procedure is a criterion for evaluating hypotheses in terms of how they explain the regularities in the input (Brent *et al.*, 1995). Kadakov proposes applying MDL search with genetic algorithms.

*Successor-frequency stemmer*

Another well-known conflation technique is the *successor stemmer*, based on the work of Harris (1955), segmenting utterances that are spelt phonetically, and creating trees whose labels correspond to a single character (Knuth, 1973). The successor stemmer establishes limits for the strings and the number of different characters that follow a word string in a corpus (Hafer and Weiss, 1974). For example, to determine the letter successor varieties of a word like "child" from a collection containing "children", "chief", "childless", "chill", "childlike", "childish" the number of different characters that follow the word string "chi" would be calculated (Figure 3). This information can be used to segment a term by:




**Source:** Adapted from Kazakov (1997)

**Figure 2.**
A *Naïve Theory of Morphology*, in which $N$ is the number of characters that both lexicons contain ($P + S = N$) and where *the smaller that number, the better the theory*

| Prefix | Successor variety | Letters |
|---|---|---|
| C | 1 | H |
| CH | 1 | I |
| CHI | 3 | L, E, D |
| CHIL | 2 | D, L |
| CHILD | 3 | R, L, I |
| CHILDR | 1 | E |
| CHILDRE | 1 | N |
| CHILDREN | 0 | BLANK |
| CHILDL | 2 | E, I |
| CHILDLE | 1 | S |
| CHILDLES | 1 | S |
| CHILDLESS | 0 | BLANK |

**Figure 3.**
Successor variety of a string

(1) *Cutoff*: a threshold value is set for successor variety, and the limit is identified each time that value is reached,
(2) *Peak and plateau*: the segment is cut after the characters whose successor variety is greater than that of the character that precedes or follows it,
(3) *Complete word*: the segment is cut when a complete word from the list of the corpus is formed; or
(4) *Count*: the most frequent prefix is used to determine the stem.

n-*gram stemmer*

An alternative is using similarity measures based on the number of diagrams in common instead of terms, then applying clustering techniques. The measures are based on *n*-gram similarities, where the *n*-gram of string is any substring of some fixed length. They have been extensively applied to tasks related to IR, such as query expansion (Adamson and Boreham, 1974; Lennon *et al.*, 1981; Cavnar, 1994; Damashek, 1995). At the same time, *n*-grams have been used in the automatic spelling correction (Angell *et al.*, 1983; Kosinov, 2001), on the assumption that the problems of morphological variants and spelling variants are similar.

*N*-gram stemmers conflate terms based on the number of *n*-grams that are shared by the terms, and are language independent. Adamson and Boreham (1974) calculated a similarity co-efficient between words as a factor of the number of shared sub-strings, to then pair words according to the number of *n*-grams. After counting the number of *n*-grams among the word pairs, the degree of similarity is calculated using the Dice coefficient (Adamson and Boreham, 1974; Robertson and Willett, 1998) or some other means of determining the degree of association between two binary variables ($\phi$-coefficient, odds ratio, or *t*-score). To assess the degree of association of the terms "statistics" ($w_1$) and "statistically" ($w_2$) they could be represented, for example, as in Figure 4.

Once we have the only bigrams shared by the two words, we apply the Dice coefficient:

$$\text{Similarity score } (w_1, w_2) = \frac{(2\chi)}{(\alpha + \beta)}$$

where $\chi$ is the number of unique bigrams shared, $\alpha$ the number of unique bigrams of the first word, and $\beta$ is the number of unique bigrams of the second word.

The similarity measure is performed on all pairs of terms in the IRS database or dictionary, giving a matrix of word-word similarities. The words are associated using a technique of grouping and classification, such as clustering. There are a number of variants of clustering which can give different groupings depending on

| Word | 2-grams | Unique 2-grams | Share Unique 2-grams |
|---|---|---|---|
| $W_1$ | {*s, st, ta, at, ti, is, st, ti, ic, cs, s*} | {*s, st, ta, at, ti, is, ic, cs, s*} | {*s, st, ta, at, ti, is, ic} |
| $W_2$ | {*s, st, ta, at, ti, is, st, ti, ic, ca, al, ll, ly, y*} | {*s, st, ta, at, ti, is, ic, ca, al, ll, ly, y*} | |

**Figure 4.**
Bigram matching

the agglomeration rule applied. The simplest are the single link and the complete link rules, while more complex rules include the widely used Ward method (Egghe and Rousseau, 1990). Each *n*-gram is represented as coordinate on a vector and, on the basis of vector similarity measures, word clustering is effected. Finally, the stem is identified for each word cluster with the same prefix.

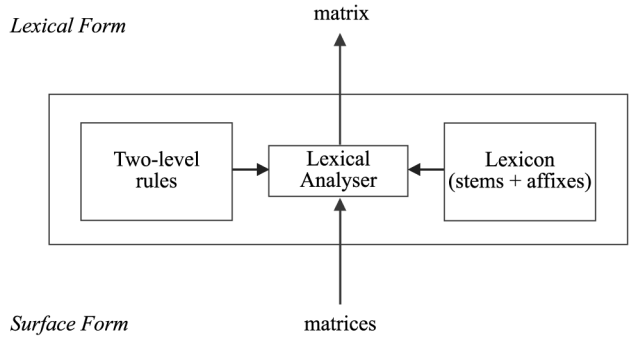
*Morphological analysis through FST*

Stemmers featuring dictionary lookup, or table lookup, provide a morphological analysis for any term included in the dictionary. The conflation techniques based on morphological analysis were first presented in a lexical analyzer developed by a group of computational linguists at xerox, the Multi-Lingual Theory and Technology Group (MLTT). One of the most important applications of this tool is morphological parsing, which can be used to reduce the single-word term variants in IRS. The analysis of the inflected forms of terms is done using a lexical database represented by finite-state mechanisms.

The xerox analyzer is based on the model of two-level morphological analysis proposed by Koskenniemi (1983). The premise behind this model is that all lexical units can be represented as a correspondence between a lexical form and surface form (canonical form, or lemma, and inflected form, respectively). Further computational development of the Koskenniemi model led to the lexical analyzer by Karttunen known as PC-KIMMO (Karttunen, 1983), the more direct forerunner of the xerox morphological analyzer.

The earliest version of the PC-KIMMO parser managed to break down words by integrating them into two analytical modules: on the one hand, a component based on the two-level morphology, and on the other hand, a lexical component including a list of morphemes including stems as well as affixes (Figure 5). Nonetheless, this early version did not provide the POS categories of the terms, and was therefore not suitable for later syntactic parsing, which needs the input of the text previously tagged with the syntactic role of a word in a sentence. This limitation was corrected in a second version of the PC-KIMMO, which incorporated the POS categories as part of the lexicon.

With PC-KIMMO a surface entry word is analyzed as structures of sequences of morphemes by two components: the lexicon, and the two-level morphological rules. Between the surface forms and the corresponding lemmas, there exists a regular




**Source:** Adapted from Karttunen (1983)

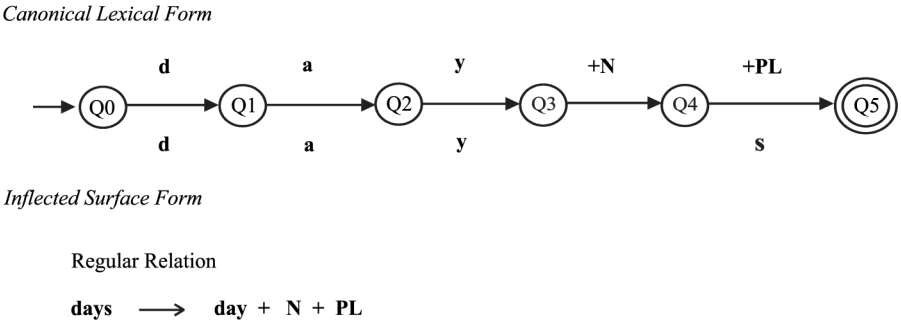
**Figure 5.**
Components of the PC-KIMMO lexical analyser

relation, defined as a relationship of equivalence, that can be compiled in a FST (Kaplan and Kay, 1994). The lemmas are stored in a dictionary, represented by transducers, where each lemma is assigned its corresponding POS category. The transducer follows a path, or sequence of states and transitions, from an initial state to a final one, to arrive at an association of the inflected surface forms with the canonical lexical forms (Figure 6).

The correspondence between inflectional variants and lemmas is complicated, however, when context-dependent morphological alterations mask the equivalency of the two forms. The xerox lexical analyzer features two module components (Karttunen and Kay, 1994): a lexicon that defines the set of lexical forms of a language; and a set of rules that connect the surface forms to lexical forms and POS categories. A morphological lexicon for English containing over 317,000 inflected forms derived from over 90,000 stems is available (Karp *et al.*, 1992).

An alternative lexical analyzer based on finite mechanisms is the one proposed by Silberztein (1993), which works without morphological rules; rather, the irregularities are represented directly in a graph editor. Its technology has been described by Roche and Schabes (1997). FST associate sets of suffixes to the corresponding inflectional information. In order to produce the inflected forms, on needs to be able to delete characters from the lemma. For this purpose, a delete character operator (*L*) is used, which does not require morphological rules nor the help of a finite-state calculus (Silberztein, 1993, 2000). The application developed by Silberztein consists of a dictionary (known as DELAS) of canonical forms with syntactic codes that indicate the POS category of each entry. Each code is linked to a graphic FST made up of an initial node and a final node that describe the path the morphological analyzer should trace. For instance, all the nouns associated with the same inflectional information are associated with the same inflectional FST. In Figure 7, we show the inflection of nouns with the code N01.

Once the FST are compiled, they are projected upon the dictionary of canonical forms, automatically producing the expanded dictionary of inflected forms (known as DELAF) that contains the canonical forms along with inflected forms, POS categories, and inflectional information, such as singular (s) or plural (p). With the application of




**Note:** The first member of the pair is the upper string, or the canonical form made up of the lemma + POS category (*day* + *Noun* + *Plural*): and the second one is the lower string, the inflected form consisting of the variant (*days*)

**Source:** Adapted from Karttunen *et al*. (1992)

**Figure 6.**
A transducer encodes a regular relation between a set of pairs of strings

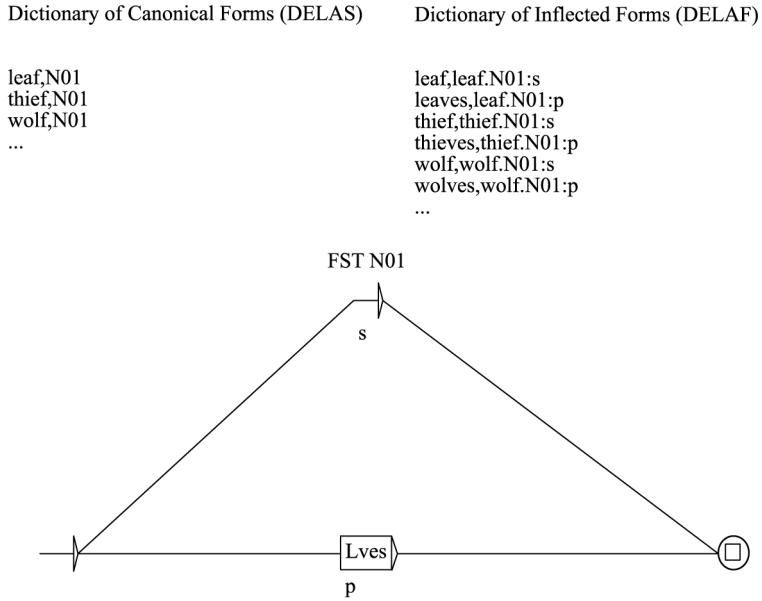



**Figure 7.**
The FST N01 associates the sets of suffixes of the DELAS entries (such as "leaf", "thief" or "wolf") to the corresponding inflectional codes (s, singular and p, plural).

**Note:** In order to obtain the inflected forms from the lemmas in the DELAF entries (such as '*leaves*', '*thieves*', or '*wolves*') the last letter '*f* ' of the lemma should be eliminated using the delete operator *L* (*Left*)

the dictionaries on the lexical units of a corpus, we finally effect two transformations: lemmatization of the inflected forms and POS tagging.

## Conflation methods for normalizing multiterm variants

Multi-word terms are considered to be more specific indicators of document content than are single words, and for this reason many methods have been developed for their identification. Basically there are two approaches, non-linguistic methods and linguistic methods, which respectively provide statistical phrases and syntactic phrases.

The identification of phrases using statistical techniques is based on the co-occurrence of the terms, on the application of similarity coefficients and clustering techniques. Although co-occurrence analysis has been used mainly in the construction of thesauri (Sparck Jones and Tait, 1984), it can also be applied to the extraction and conflation of multi-word terms. To identify these, the text must be pre-processed to obtain a phrasal lexicon, defined as a list of NP appearing with certain frequency (Fagan, 1989). The subsequent indexing of the documents is based on the identification of the phrases using the lexicon. Salton and McGill (1983) demonstrate that the statistical procedures suffer from certain weaknesses:

(1) the selected phrases are very often improperly structured from a syntactic standpoint, and

(2) the lack of control in the selection of the phrases may lead to errors that reduce the efficiency of the IRS.

To reduce these problems, we need NLP linguistic methods that can identify the syntactic structures of these constructions and establish some sort of control in the selection of multi-word terms. However, in order that the application of NLP techniques to IRS be effective, certain conditions must prevail (Evans and Zhai, 1996). First, they must be able to process a great deal of texts. Second, they must process texts without restrictions, in which unknown words, proper names, or transcription errors may appear. And third, they must provide the surface representation of the contents of the texts.

The above conditions help simplify NLP techniques when applied to IRS, because although the lexical and syntactic analyzers act upon the texts without restrictions from the databases, in-depth analysis of the documents is not required. The application of NLP to texts involves a sequence of analytical tasks performed in the separate modules that constitute the linguistic architecture of the system. Among available tools for NP extraction: the category tagger based on Brill's rules (Brill, 1992); the Xerox morphological analyzer (Karttunen, 1983; Karttunen *et al.*, 1992); disambiguation devices of POS categories based on stochastic methods, such as the Hidden Markov Model (HMM) (Cutting *et al.*, 1992; Kupiec, 1992, 1993); the NPtool phrase analyzer (Voutilainen, 1997); or the *AZ* noun phraser, an analyzer developed by the artificial intelligence laboratory of the University of Arizona (Tolle and Chen, 2000), which combines tokenizing with POS tagging (Brill, 1993).

Whether general linguistic resources or specific tools are used, recognizing the variants of phrases continues to be a problem. Ideally, programs would be able to reduce all the variants to canonical or normalized forms, where each phrase would be assigned a clearly defined role reflecting the complexity of the syntactic structure. This network of nodes and transitions tagged with POS categories determines sequences in the input, and supplies some form of linguistic information as the output. An entry stream is recognized and transformed into a normalized stream if a path is produced from one node, considered the initial state, to another node, constituting the final state. Nonetheless, despite the simplicity of the finite-state mechanisms, a number of complications might arise in the detection of phrasal structures:

(1) Structural ambiguity, a given construction may be analyzed with regard to different syntactic patterns that are all correct according to the grammar used, and this leads to overanalysis,

(2) Lack of coverage, or underanalysis, when the grammatical formalisms can only detect combinations specified by the grammar rules, and

(3) Determining the type of relationship shared by the constituent parts of the NP, which must be more than the simple juxtaposition of components.

Moreover, the application of FST requires overcoming one of the oldest and greatest problems surrounding the recognition of syntactic variants, which is the need to store and manage the thousands of variants that NP may have, making their identification and conflation unfeasible (Salton, 1989).

*String-similarity through word-form co-occurrence*

The application of non-linguistic techniques for the extraction of multi-word terms is based on the statistical association or conflation of words that represent similar concepts. As in the case of lexical conflation – in which meanings are grouped

according to the forms of the terms – in the conflation of multi-word terms, specific identifiers of documentary contents are created to enhance the precision of the IRS. This involves checking word-co-occurrence and the similarity of strings, and using clustering techniques. After binary relations are established, the terms are grouped in clusters of highly similar terms, yet these have very low similarity with the terms of other clusters.

Similarity measures are based on the attributes describing terms, and in this case the frequency of multi-word terms is greater than the frequency of their separate components. One measure of association of compound terms based on a probabilistic model is mutual information (MI). The association between terms has been used profusely over the past two decades to improve retrieval effectiveness. The co-occurrence research in IR carried out by Van Rijsbergen (1977), Harper and van Rijsbergen (1978) and Smeaton and van Rijsbergen (1983) attempted to identify significantly associated document-level co-occurrence by means of Expected Mutual Information (EMIM) to build dependence trees called maximum spanning trees (MST). Closely related terms from the MST were then used for query expansion.

In MI the joint probability of two words is compared with the probability that they appear independently. If two words $w_1$ and $w_2$ have the probabilities $P(w_1)$ and $P(w_2)$ of occurrence, then the mutual information $\mathrm{MI}(w_1, w_2)$ similarity coefficient is obtained by applying the following equation (Church and Hanks, 1990):

$$\mathrm{MI}\,(w_1,\ w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$

where $P(w_1,w_2)$ is the joint probability of two words $(w_1, w_2)$ in the corpus, $P(w_1)$ the probability of the independent occurrence of the word $w_1$ in the corpus, and $P(w_2)$ is the probability of the independent occurrence of the word $_{\mathrm{w2}}$ in the corpus.

Probabilities $P(w_1)$ and $P(w_2)$ are calculated by counting the frequency of occurrence of $f(w_1)$ and $f(w_2)$ in a corpus. The joint probability $P(w_1,w_2)$ is the count of the frequency with which word $w_1$ is followed by $w_2$, within a parameter or set of words $f_w(w_1w_2)$, normalized in corpus $N$:

$$MI(w_1, w_2) = \log_2 \frac{Nf(w_1, w_2)}{f(w_1)f(w_2)}$$

A high score of $MI(w_1, w_2) > 0$ indicates that there is a genuine association between two words, then are very likely to occur together. The lowest coefficient, $MI(w_1, w_2) < 0$, indicates a complementary distribution. A low score of $MI(w_1, w_2) \approx 0$ would point to an irrelevant association (Church and Hanks, 1990).

The result of the above procedure is a normalized similarity matrix representing the associations of compound terms, from which they can be conflated into a single NP by means of clustering techniques such as the hierarchical method of simple linking. Under single-link clustering, the two elements of the matrix that are nearest are selected and fused, and the least possible distance between this new cluster and the other elements of the matrix is calculated. In the case of single-link clustering, chaining relationships are established. Though this is not the best means of obtaining general classification schemes, it is the most appropriate for identifying the proximity among elements. In an iterative process, the distance between two clusters is calculated as

the distance between their two nearest elements; thus the distance $d_{AB}$ between clusters A and B is expressed as:

$$d_{AB} = \min(d_{ij})$$

where $d_{ij}$ is the distance between element $i$, belonging to cluster A, and element $j$, belonging to cluster B. There is no single solution to this conflating process. Rather, at any stage in the fusing process, the most satisfactory term for the grouping of the different strings is selected, a decision depending largely on the structures of the different phrases.

The statistical methods for extracting phrases also have their weak points. Firstly, they may produce a grouping of words that actually represent very different concepts, because the number of frequencies is different than the similarity coefficient. The joint probabilities are symmetric, $P(w_1,w_2) = P(w_2,w_1)$, and therefore the similarity coefficient is also symmetric, yet the ratio of frequencies is asymmetric, $f(w_1,w_2) \neq f(w_2,w_1)$, (Church and Hanks, 1990). The second weakness is manifest when dealing with infrequent terms in the corpus. The third weakness is that the process of forming phrases using only the co-occurrence of terms has been shown to generate statistically significant phrases that are syntactically incorrect (Salton, 1989). All these negative effects on retrieval performance underline the need for linguistic methods to identify and to conflate multi-word terms.

*String-similarity through* n-*gram co-occurrence*

The MI coefficient can also be used to group words in clusters in accordance with the shared *n*-grams. The probabilistic modeling of the language with *n*-grams takes into account the context where the word appears. The co-occurrence of *n*-grams for a word $w_i$ is the set of probabilities that the word is followed by another word string, for each possible combination of $w_i$ in the vocabulary of that language. A likeness approach would be to extract NP, using another unit of text as POS tags, on the basis of the number of shared common *n*-grams. As POS tagging depends on context, Church (1988) proposed a stochastic method to extraction NP based on statistical information. The contextual probability is estimated counting the frequencies for POS tags from the Tagged Brown Corpus (Francis and Kucera, 1979). Thus, for instance, the probability of observing a Verb (V) before an Article (AT) and a Noun (N) is estimated to be the ratio of the trigram frequency (V, AT, N) over bigram frequency (AT, N).

The linking coefficient between *n*-grams can also be calculated by MI to obtain the association matrix of the conditioned probability of a word $w_1$ being followed by other words. To arrive at the conditioned probability in a text sample $W = w_1, w_2 \ldots w_n$, the following formulas would be applied in the case of unigrams, bigrams, or trigrams:

$$P(W) = \prod_{i}^{n} P(w_i)$$

$$P(W) = \prod_{i}^{n} P(w_i|w_{i-1})$$

$$P(W) = \prod_{i}^{n} P(w_i|w_{i-2}w_{i-1})$$

To calculate these probabilities, we would need the associated expected frequencies, from the following estimation parameters:

$$P(w_i) = \frac{f(w_i)}{N}$$

$$P(w_i|w_{i-1}) = \frac{f(w_{i-1}, w_i)}{f(w_{i-1})}$$

$$P(w_i|w_{i-2}w_{i-1}) = \frac{f(w_{i-2}, w_{i-1}, w_i)}{f(w_{i-2}, w_{i-1})}$$

To construct the association matrix in the case of bigrams, we could apply an adaptation of the MI similarity coefficient, with which the probability of the co-occurrence of word $w_1$ along with word $w_{1-1}$ is obtained, as well as the independent probability of the occurrence of word $w_1$and of word $w_{1-1}$:

$$MI(w_1, w_{1-1}) = \log_2 \frac{P(w_1|w_{1-1})}{P(w_1)P(w_{1-1})}$$

where $P(w_1|w_{1-1})$ would be the conditioned probability of the co-occurrence of two words $(w_{1-1}, w_1)$ in the corpus, $P(w_1)$ would be the probability of the independent occurrence of word $w_1$ in the corpus, and $P(w_{1-1})$ would be the probability of the independent occurrence of word $w_{1-1}$ in the corpus.

The probabilities $P(w_1)$ and $P(w_{1-1})$ are calculated from the frequency of $f(w_1)/N$ and $f(w_{1-1})/N$. The conditioned probability $P(w_1|w_{1-1})$ is calculated as the frequency of $f(w_{1-1}, w_1)/f(w_{1-1})$, normalized in corpus $N$:

$$MI(w_1, w_{1-1}) = \log_2 \frac{\frac{Nf(w_{1-1}, w_1)}{f(w_{1-1})}}{f\frac{(w_1)}{N} f\frac{(w_{1-1})}{N}} = \log_2 \frac{Nf(w_{1-1}, w_1)}{f(w_1)f(w_{1-1})}$$

The result is also a similarity matrix, representing in this case the associations conditioned by the multi-term context. They could then be grouped by single-link clustering until a cluster is obtained, to select the most adequate representation of the different variants. On the other hand, in addition to MI, some other statistical measure could be applied such as those habitually used by IRS: Cosine coefficient (Salton and McGill, 1983), or Jaccard coefficient (Hamers *et al.*, 1989).

*Syntactic pattern-matching through FSTs*

The multiterms analysis through the use of linguistic techniques requires the development of rule-based methods, such as local grammars, that make underlying syntactic structures explicit. The most extensive classification of such grammatical formalisms is the Chomsky hierarchy (1957), where grammars are defined on the basis of their potential to generate the different linguistic constructions of a language. The least expressive or least powerful would be a regular grammar (Type 3), whereas

the most expressive would be the Syntagmatic Grammars (Type 0). Grammatical formalisms are needed to detect multi-word terms because the output from a parser is limited by information from the lexicons and electronic grammars. Because IRS analysis is restricted to NP, it is not necessary to use robust formalisms; even weak formalisms are quite effective, developed using relatively simple parsing techniques.

The construction of grammars to identify and group syntactic structures relies on the drawing of parallels between natural languages and artificial ones; both types are defined by a set of mathematical formalizations called regular expressions (RE). The RE represents syntactic patterns, or NP. Consequently, the first step for identifying NP is to create the grammars that will reflect the correct RE, and then transfer them to a mechanism that will acknowledge them and group all their syntactic variants.

The extraction of syntactic patterns using FSA is based on a finite set of states and a set of transitions from state to state that occur on input symbols chosen from a alphabet $\Sigma$ (Hopcroft and Ullman, 1979). This mathematical model can be further classified as a deterministic finite automata or a non-deterministic finite automata. The basic difference lies in their capacity for changing from one state to another, depending on the input. These techniques, when used in syntactic analysis, have given rise to transition networks (TN), which are networks of nodes and arcs, tagged with terminal symbols that may be words or POS categories. In this sense, we can say that a TN is the equivalent of an FSA.

To recognize multi-word terms through FSA, their structures must be described using RE, defined as a metalanguage for the identification of syntactic patterns. To extract and identify RE, two procedures can be used. First, generating a list with all the language strings, which would be then be compared or considered the equivalent of the given string. Second, constructing an FSA. Thus, if $r$ is a RE, then there exists an FSA, $A$, that includes $r$, leading to the following equivalency (Hopcroft and Ullman, 1979): $L(r) = L(A)$. In other words, the metalanguage represented by a RE, $L(r)$, is only equivalent to the set of string belonging to the language that is recognized or accepted by the automata, $L(A)$. Through this technique, we use the specification of RE to determine the language formed by syntactic patterns, such as:

$$\mathrm{NP} \rightarrow \mathrm{N}\,[\mathrm{noun}]$$

$$\mathrm{ER}_0 = \mathrm{N}$$

$$\mathrm{NP} \rightarrow \mathrm{AT}\ \ \mathrm{N}\,[\mathrm{article\_noun}]$$

$$\mathrm{ER}_1 = \mathrm{AT}\ \ \mathrm{N}$$

$$\mathrm{NP} \rightarrow \mathrm{DEM}\ \ \mathrm{N}\,[\mathrm{demonstrative\_noun}]$$

$$\mathrm{ER}_2 = \mathrm{DEM}\ \ \mathrm{N}$$

$$\mathrm{NP} \rightarrow \mathrm{AT}\ \ \mathrm{ORD}\ \ \mathrm{N}\,[\mathrm{article\_ordinal\_noun}]$$

$$\mathrm{ER}_3 = \mathrm{AT}\ \ \mathrm{ORD}\ \ \mathrm{N}$$

$$\mathrm{NP} \rightarrow \mathrm{AT}\ \ \mathrm{CARD}\ \ \mathrm{N}\,[\mathrm{article\_cardinal\_noun}]$$

$$\mathrm{ER}_4 = \mathrm{AT}\ \ \mathrm{CARD}\ \ \mathrm{N}$$

Bearing in mind the connection between RE and FSA, the Kleene theorems (Klenee, 1956) stand as the methodological basis of the theory of finite automata. The Theorem

of Analysis states that every language that can be defined by transition graph can also be defined by a RE. This theorem demonstrates that an acceptable language can be obtained from a FSA and can generate the RE that represents it. The Theorem of Synthesis states that every language that can be defined by a regular expression can also be defined by a FSA. This theorem demonstrated that a RE can be used to generate the FSA that accepts the language described by a particular RE. Within the theorem of synthesis, there are two procedures for obtaining the FSA recognized by an RE:

(1) the association of each possible RE with the FSA that recognizes the language describing the RE, and

(2) the calculation of derivatives of the RE to obtain the Regular Grammar (Type 3) equivalent to the FSA able to recognize the languages.

Adopting the first procedure, without a finite-state calculus, the RE are represented graphically, with the graphic editor FSGraph (Silberztein, 1993, 2000). The RE, or sets of graphs, can be compiled into FSA. In order that the FSA themselves recognize the syntactic patterns, a previous morphological analysis will be needed, giving POS tags to the lexical units. A path between two FSA nodes takes place only if the input chain string belongs to the category with which the transition is tagged.

Given that a NP is a complex construction containing a noun as its nucleus, and possibly one or more modifiers, all these elements are considered to be constituents, and they function as units of nominative constructions. In the structure of constituents of NP, any component that is subordinated to the nucleus is commonly called a modifier, and its function is to specify and constrain the head noun. This is a functional approach for distinguishing constituents and it may be represented using the same graphic interface, as illustrated in Figure 8.

The similar structures can then be transferred, using the same graphic tool, to an FST, where the syntactic patterns will be recognized and be conflated into hand-made canonical structures. Thus, we considered that an FST is a method for reducing syntactic structures, comprising two automata that work in a parallel manner.

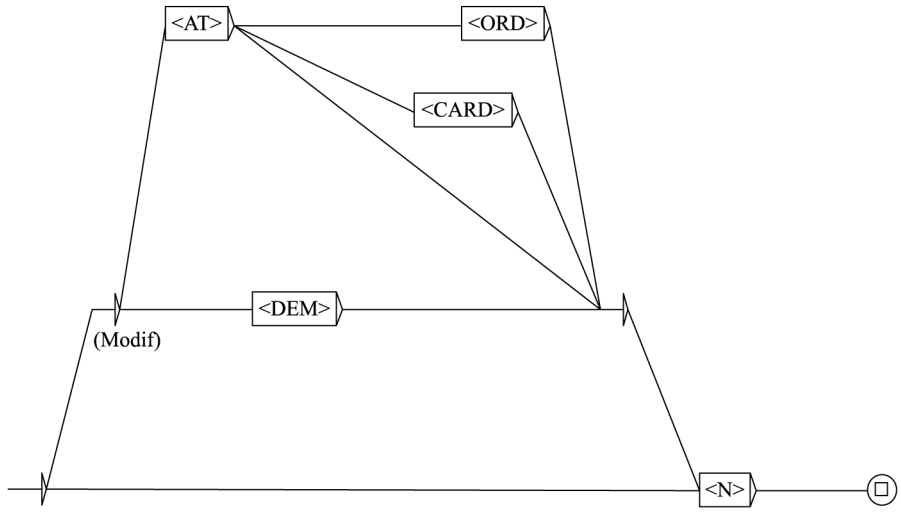



**Figure 8.**
Syntactic patterns-matching through FSA

One automata identifies the surface strings, and the other establishes a regular relation, defined as an equivalence relation, between the different syntactic structures and a normalized structure. In order to use this formalism of the IRS as a means of controlling NP structures, we propose the transformation of the canonical syntactic forms into identifiers of enumerated NP (Figure 9) which will be implemented as groupers of structures.

Nevertheless, and despite the efficacy of the state-finite mechanisms in identifying and grouping thousands of syntactic variants over thousands of transitions, we cannot entirely avoid situations of ambiguous tagging, or the problems of overanalysis and underanalysis. All this reconfirms the deficiencies of linguistic methods in extracting and conflating multi-word terms in IRS.

Salton and McGill (1983) argued early on that linguistic procedures can be quite effective if analysis is restricted to canonical representations into:

(1) limited subject areas,

(2) limited vocabulary, and

(3) limited syntactic patterns.

Consequently, we believe at this point in time that the solution to indexing problems using syntactic phrases could reside in the capacity for developing linguistic models that allow the generation and recognition of normalized forms within well-defined domains.

## Conclusions

In IRS, the textual documents are habitually transformed into document representatives by means of linguistic structures configured as indexing terms, classified essentially as single-word terms or uniterms, and multi-word terms or multiterms. The single-word terms have morphological variants that refer to the same meaning, and their grouping would improve average recall. Although uniterms may be ambiguous, they usually have relatively few variants, and from a computational

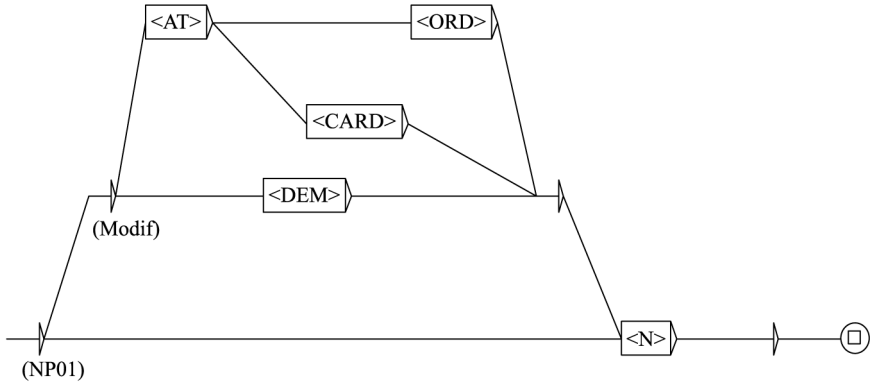



**Figure 9.**
Regular relation between variants of syntactic patterns and normalized NP

treatment, they are easier to formalize. In contrast, multiterms are much more specific, but the grouping of their variants is plagued by difficulties in their identification, because IR systems tend to work under the assumption that similar synctactic structures have similar meanings, and should be treated as equivalents, and this is very difficult to regulate in view of the variability of syntactic structures.

There are morphological, lexical and syntactic variants that cannot be recognized other than through term conflation. The standardization methods most widely evaluated on retrieval performance involve stemming, segmentation rules, assessing similarity measures of pairs of terms, and clustering techniques. In the linguistic framework, term conflation methods could be considered equivalence techniques, employed to regulate linguistic variants and optimize retrieval performance.

The application of NLP tools in IR involves morphological analysis, POS taggers, disambiguation processes, lemmatization and shallow parsing for syntactic pattern-matching. This implies more work than the classic IR approach, and the results of evaluations have been overall discouraging. Nevertheless, for languages characterized by strong morphology, the linguistic techniques may constitute adequate normalizing methods. Similarly, for the recognition of patterns within specific and well-defined domains, the linguistic techniques may prove effective conflation procedures.

With this functional aspect in mind, we believe a optimal solution for term normalization is the construction of morphological and syntactic analyzers by means of finite-state mechanisms that would identify both uniterms and multiterms. To transform these expressions into canonical forms, we propose the use of FST, a method that allows the control of candidate terms, and has the potential as well to follow up their processing so that they may eventually be added to the search file. The application of FST may also, however, afford advantages that have not yet been properly explored and evaluated, and their alternative or complementary use might enhance the management of term variants in retrieval performance.

As a final consideration, and a somewhat risky one at that, we put forth that the linguistic techniques of uniterm conflation could enhance average recall in IR when dealing with synthetic languages, whereas the linguistic techniques of multiterm conflation could enhance average precision in the case of IR using analytic languages. Further evaluation is necessary to demonstrate this hypothesis.

## References

Abney, S. (1991), "Parsing by chunks", in Berwick, R., Abney, S. and Tenny, C. (Eds), *Principle-Based Parsing*, Kluwer Academic Publishers, Dordrecht.

Abu-Salem, H., Al-Omari, M. and Evens, M.W. (1999), "Stemming methodologies over individual queries words for an Arabian information retrieval system", *Journal of the American Society for Information Science*, Vol. 50 No. 6, pp. 524-9.

Adamson, G.W. and Boreham, J. (1974), "The use of an association measure based on character structure to identify semantically related pairs of words and document titles", *Information Storage and Retrieval*, Vol. 10 No. 1, pp. 253-60.

Ahmad, F., Yussof, M. and Sembok, M.T. (1996), "Experiments with a stemming algorithm for malay words", *Journal of the American Society for Information Science*, Vol. 47 No. 1, pp. 909-18.

Angell, R.C., Freund, G.E. and Willett, P. (1983), "Automatic spelling correction using a trigram similarity measure", *Information Processing and Management*, Vol. 19 No. 4, pp. 255-61.

Arampatzis, A.T., Tsoris, T., Koster, C.H.A. and van der Weide, P. (1998), "Phrase-based information retrieval", *Information Processing and Management*, Vol. 14 No. 6, pp. 693-707.

Arampatzis, A.T., van der Weide, P., van Bommel, P. and Koster, C.H.A. (2000), "Linguistically motivated information retrieval", in Kent, A. (Ed.), *Encyclopedia of Library and Information Science*, Marcel Dekker, New York, NY Basel.

Brent, M., Lundberg, A. and Murthy, S.K. (1995), "Discovering morphemic suffixes: a case study in minimum description length induction", *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Vanderbilt University, Ft. Lauderdale, FL.

Brill, E. (1992), "A simple rule based part-of-speech tagger", *Third Conference on Applied Natural Language Proceedings*, Trento, pp. 152-5.

Brill, E. (1993), "A corpus-based approach to language learning", PhD thesis, Department of Computer and Information Science, University of Pennsylvania, University Park, PA.

Buckley, C., Alland, J. and Salton, G. (1995), "Automatic routing and retrieval using SMART: TREC-2", *Information Processing and Management*, Vol. 31 No. 3, pp. 315-26.

Cavnar, W.B. (1994), "Using an n-gram based document representation with a vector processing retrieval model", *Proceedings of the Third Text REtrieval Conference (TREC-3)*, Special Publication 500-226, National Institute of Standards and Technology (NIST), Gaithersburg, MA.

Chomsky, N. (1957), *Syntactic Structures*, Mouton, The Hague.

Church, K. (1988), "A stochastic parts program and noun phrase parser for unrestricted text", paper presented at Second Conference on Applied Natural Language Processing, Austin, TX.

Church, K.W. and Hanks, P. (1990), "Word association norms, mutual information and lexicography", *Computational Linguistics*, Vol. 16, pp. 22-9.

Croft, W.B., Turtle, H.R. and Lewis, D.D. (1991), "The use of phrases and structured queries in information retrieval", *Proceedings, SIGIR 1991*, pp. 32-45.

Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P. (1992), "A practical part-of-speech tagger", paper presented at Third Conference on Applied Natural Language Processing, Trento, pp. 133-40.

Damashek, M. (1995), "Gauging similarity with n-grams: language independent categorization of text", *Science*, Vol. 267, pp. 843-8.

Dawson, J.L. (1974), "Suffix removal for word conflation", *Bulletin of the Association for Literary and Linguistic Computing*, Vol. 2 No. 3, pp. 33-46.

Egghe, L. and Rousseau, R. (1990), *Introduction to Informetrics: Quantitative Methods in Library, Documentation and Information Science*, Elsevier, Amsterdam.

Evans, D.A. and Zhai, C. (1996), "Noun-phrase analysis in unrestricted text for information retrieval", *Proceedings of the 34th Annual Meeting of Association for Computational Linguistics*, University of California, Santa Cruz, CA, pp. 17-24.

Evans, D.A., Milic-Frayling, N. and Lefferts, R.G. (1996), "CLARIT TREC-4 experiments", in Harman, D.K (Ed.), *The Fourth Text REtrieval Conference (TREC-4)*, Special Publication 500-236, National Institute of Standards and Technology(NIST), Gaithersburg, MD.

Fagan, J.L. (1989), "The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval", *Journal of the American Society for Information Science*, Vol. 40 No. 2, pp. 115-32.

Feng, F. and Croft, W.B. (2001), "Probabilistic techniques for phrase extraction", *Information Processing and Management*, Vol. 37 No. 2, pp. 199-220.

Frakes, W.B. (1992), "Stemming algorithms", in Frakes, W.B. and Baeza-Yates, R. (Eds), *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.

Frakes, W.B. and Baeza-Yates, R. (1992), *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.

Francis, W. and Kucera, H. (1979), "Brown corpus manual", *Technique Report*, Department of Linguistics, Brown University, Providence, RI.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Goldsmith, J. (2001), "Unsupervised learning of the morphology of a natural language", *Computational Linguistics*, Vol. 27 No. 2, pp. 153-98.

Hafer, M.A. and Weiss, S.F. (1974), "Word segmentation by letter successor varieties", *Information Processing and Management*, Vol. 10 Nos 11/12, pp. 371-86.

Hamers, L., Hemerick, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R. and Vanhoutte, A. (1989), "Similarity measures in scientometric research: the Jaccard index versus Salton's cosine formula", *Information Processing and Management*, Vol. 25 No. 3, pp. 315-8.

Harman, D.K. (1991), "How effective is suffixing?", *Journal of the American Society for Information Science*, Vol. 47 No. 1, pp. 70-84.

Harman, D.K. (1997), *The sixth Text REtrieval Conference (TREC-6)*, Special Publication 500-240, National Institute of Standards and Technology (NIST), Gaithersburg, MD.

Harper, D.J. and van Rijsbergen, C.J. (1978), "An evaluation of feedback in document retrieval using co-occurence data", *Journal of Documentation*, Vol. 34 No. 3, pp. 189-216.

Harris, Z.S. (1951), *Methods in Structural Linguistics*, University of Chicago Press, Chicago, IL.

Harris, Z.S. (1955), "From phoneme to morpheme", *Language*, Vol. 31 No. 2, pp. 190-222.

Hopcroft, J.E. and Ullman, J.D. (1979), *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA.

Hull, D.A. (1996), "Stemming algorithms – a case study for detailed evaluation", *Journal of the American Society for Information Science*, Vol. 47 No. 1, pp. 70-84.

Hull, D.A., Grefenstette, G., Schulze, B.M., Gaussier, E., Schutze, H. and Pedersen, J.O. (1996), "Xerox TREC-5 site report: routing filtering, NLP and Spanish tracks", in Harman, D.K. and Voorhees, E.M. (Eds), *The Fifth Text REtrieval Conference (TREC-5)*, Special Publication 500-238, National Institute of Standards and Technology (NIST), Gaithersburg, MD.

Jacquemin, C. (2001), *Spotting and Discovering Terms Through Natural Language Processing*, MIT Press, Cambridge, MA.

Jacquemin, C. and Tzoukermann, E. (1999), "NLP for term variant extraction: synergy between morphology, lexicon, and syntax", in Strzalkowski, T. (Ed.), *Natural Language Information Retrieval*, Kluwer, Dordrecht.

Kalamboukis, T.Z. (1995), "Suffix stripping with moderm Greek", *Program*, Vol. 29 No. 3, pp. 313-21.

Kaplan, R.M. and Kay, M. (1994), "Regular models of phonological rule systems", *Computational Linguistics*, Vol. 20 No. 3, pp. 331-78.

Karp, D., Schabes, Y., Zaidel, M. and Egedi, D. (1992), "A freely available wide coverage morphological analyser for English", *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*, Nantes, pp. 950-4.

Karttunen, L. (1983), "KIMMO: a general morphological processor", *Texas Linguistics Forum*, Vol. 22, pp. 217-28.

Karttunen, L., Kaplan, R.M. and Zaenen, A. (1992), "Two-level morphology with composition", *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*, Nantes, pp. 141-8.

Kazakov, D. (1997), "Unsupervised learning of naïve morphology with genetic algorithms", in Daelemans, W., Bosch, A. and Weijters, A. (Eds), *Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks*, Prague, pp. 105-12.

Kazakov, D. and Manandhar, S. (2001), "Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming", *Machine Learning*, Vol. 43 Nos 1/2, pp. 121-62.

Kemeny, J.G. and Snell, J.L. (1976), *Finite Markov Chains*, Springer-Velarg, New York, NY.

Klenee, S.C. (1956), "Representation of events in nerve nets and finite automata", *Automata Studies*, Princeton University Press, Princeton, NJ.

Knuth, D. (1973), *The Art of Computer Programming: Sorting and Searching*, 3, Addison-Wesley, Reading, MA.

Kosinov, S. (2001), "Evaluation of n-grams conflation approach in text-based information retrieval", *Proceedings of International Workshop on Information Retrieval*, Oulu.

Koskenniemi, K. (1983), *Two-level Morphology: A General Computational Model for Word-form Recognition and Production*, Department of General Linguistics, University of Helsinki.

Koskenniemi, K. (1996), "Finite-state morphology and information retrieval", *Proceedings of ECAI-96 Workshop on Extended Finite State Models of Language*, Budapest, pp. 42-5.

Kraaij, W. and Pohlmann, R. (1994), "Porter's stemming algorithm for Dutch", in Noordman, L.G.M. and de Vroomen, W.A.M. (Eds), *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, Tilburg, pp. 167-80.

Kraaij, W. and Pohlmann, R. (1995), "Evaluation of a Dutch stemming algorithm", in Rowley, R. (Ed.), *The New Review of Document and Text Management*, Vol. 1, Taylor Graham, London.

Krovetz, R. (1993), "Viewing morphology as an inference process", in Korfhage, R. (Ed.), *Proceedings of the 16th ACM/SIGIR Conference*, Association for Computing Machinery, New York, NY, pp. 191-202.

Kupiec, J. (1992), "Robust part-of-speech tagging using a Hidden Markov Model", *Computer Speech and Language*, Vol. 6, pp. 225-42.

Kupiec, J. (1993), "Murax: a robust linguistic approach for question answer using an on-line encyclopedia", in Korfhage, R., Rasmussen, E. and Willett, P. (Eds), *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburg, PA, pp. 160-9.

Lennon, M., Pierce, D.S., Tarry, B.D. and Willett, P. (1981), "An evaluation of some conflation algorithms for information retrieval", *Journal of Information Science*, Vol. 3 No. 4, pp. 177-83.

Lovins, J.B. (1968), "Development of a stemming algorithm", *Mechanical Translation and Computational Linguistics*, Vol. 11, pp. 22-31.

Mitchell, T.M. (1997), *Machine Learning*, McGraw-Hill, New York, NY.

Mohri, M. and Sproat, R. (1996), "An efficient compiler for weighted rewrite rules", *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, ACL-96, Santa Cruz, California, pp. 231-8.

Paice, C.D. (1990), "Another stemmer", *ACM SIGIR Forum*, Vol. 24 No. 3, pp. 56-61.

Paice, C.D. (1996), "A method for evaluation of stemming algorithms based on error counting", *Journal of the American Society for Information Science*, Vol. 47 No. 8, pp. 632-49.

Pirkola, A. (2001), "Morphological typology of languages for IR", *Journal of Documentation*, Vol. 57 No. 3, pp. 330-48.

Popovic, M. and Willett, P. (1992), "The effectiveness of stemming for natural-language access to slovene textual data", *Journal of the American Society for Information Science*, Vol. 43 No. 5, pp. 384-90.

Porter, M.F. (1980), "An algorithm for suffix stripping", *Program*, Vol. 14, pp. 130-7.

Robertson, A.M. and Willett, P. (1998), "Applications of n-grams in textual information systems", *Journal of Documentation*, Vol. 54 No. 1, pp. 48-69.

Roche, E. (1999), "Finite state transducers: parsing free and frozen sentences", in Kornai, A. (Ed.), *Extended Finite State Models of Language*, Cambridge University Press, Cambridge.

Roche, E. and Schabes, Y. (1997), *Finite State Language Processing*, MIT Press, Cambridge, MA.

Salton, G. (1980), "The SMART system 1961-1976: experiments in dynamic document processing", *Encyclopedia of Library and Information Science*, Vol. 28, pp. 1-36.

Salton, G. (1989), *Automatic Text Processing the Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, Reading, MA.

Salton, G. and McGill, M.J. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, NY.

Savary, A. and Jacquemin, C. (2003), "Reducing information variation in text", *Lecture Notes in Computer Science*, Vol. 2705, pp. 145-81.

Savoy, J. (1993), "Stemming of French words based on grammatical categories", *Journal of the American Society for Information Science*, Vol. 44 No. 1, pp. 1-9.

Savoy, J. (1999), "A stemming procedure and stopword list for general French corpora", *Journal of the American Society for Information Science*, Vol. 50 No. 10, pp. 944-52.

Schinke, R., Greengrass, M., Robertson, A.M. and Wilett, P. (1996), "A stemming algorithm for Latin text database", *Journal of Documentation*, Vol. 52 No. 2, pp. 172-8.

Schwarz, C. (1990), "Automatic syntactic analysis of free text", *Journal of the American Society for Information Science*, Vol. 41 No. 6, pp. 408-17.

Smeaton, A.F. and van Rijsbergen, C.J. (1983), "The retrieval effects of query expansion on a feedback document retrieval system", *The Computer Journal*, Vol. 26 No. 3, pp. 239-46.

Shannon, C.E. and Weaver, W. (1949), *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL.

Sheridan, P. and Smeaton, A.F. (1992), "The application of morpho-syntactic language processing to effective phrase matching", *Information Processing and Management*, Vol. 28 No. 3, pp. 349-69.

Silberztein, M. (1993), *Dictionnaires Électroniques et Analyse Automatique de Textes: le Système INTEX*, Masson, Paris.

Silberztein, M. (2000), "INTEX: an FST toolbox", *Theorical Computer Science*, Vol. 231 No. 1, pp. 33-46.

Smadja, F. (1993), "Retrieving collocations from text: XTRACT", *Computational Linguistics*, Vol. 19 No. 1.

Sparck Jones, K. and Tait, J.I. (1984), "Automatic search term variant generation", *Journal of Documentation*, Vol. 40 No. 1, pp. 50-66.

Strzalkowski, T. (1996), "Natural language information retrieval", *Information Processing and Management*, Vol. 31 No. 3, pp. 397-417.

Strzalkowski, T., Lin, L., Wang, J. and Pérez-Carballo, J. (1999), "Evaluating natural language processing techniques in information retrieval: a TREC perspective", in Strzalkowski, T. (Ed.), *Natural Language Information Retrieval*, Kluwer Academic Publishers, Dordrecht, pp. 113-45.

Tolle, K.M. and Chen, H. (2000), "Comparing noun phrasing techniques for use with medical digital library tools", *Journal of the American Society for Information Science*, Vol. 51 No. 4, pp. 352-70.

Turing, A. (1936), "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, Vol. 42 No. 2, pp. 230-65.

Van Rijsbergen, C.J. (1977), "A theoretical basis for the use of co-occurrence data in information retrieval", *Journal of Documentation*, Vol. 32 No. 2, pp. 106-19.

Voutilainen, A. (1997), "A short introduction to NPtool", available at: www.lingsoft.fi/doc/nptool/intro/

Xu, J. and Croft, B. (1998), "Corpus-based stemming using co-occurrence of word variants", *ACM Transactions on Information Systems*, Vol. 16 No. 1, pp. 61-81.