KIMMO KETTUNEN

Reductive and Generative Approaches to Morphological Variation of Keywords in Monolingual Information Retrieval

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Information Sciences of the University of Tampere,
for public discussion in the Auditorium Pinni B 1097,
Kanslerinrinne 1, Tampere, on October 27th, 2007, at 12 o'clock.

UNIVERSITY OF TAMPERE

ACADEMIC DISSERTATION
University of Tampere
Department of Information Studies
Finland

Distribution
Bookshop TAJU
P.O. Box 617
33014 University of Tampere
Finland

Tel. +358 3 3551 6055
Fax +358 3 3551 7685
taju@uta.fi
www.uta.fi/taju
http://granum.uta.fi

Cover design by
Juha Siro

Acta Universitatis Tamperensis 1261
ISBN 978-951-44-7087-5 (print)
ISSN 1455-1616

Acta Electronica Universitatis Tamperensis 654
ISBN 978-951-44-7088-2 (pdf)
ISSN 1456-954X
http://acta.uta.fi

Tampereen Yliopistopaino Oy – Juvenes Print
Tampere 2007

Gud slog mig, jag smällde
tillbaka

Gunnar Björling

Kiitokset

Väitöskirjani juuret ovat ajallisesti kaukana, 1980-luvun alkupuolen ja puolivälin yleisen kielitieteen opinnoissani Helsingin yliopistossa. Silloin luotiin pohjaa suomalaiselle tietokonelingvistiikalle, alalle joka nyttemmin tunnetaan nimellä kieliteknologia. Minulla oli ilo osallistua silloin erityisesti professori Fred Karlssonin kursseille, joilla käsiteltiin suomen kielen morfologian automaattista analyysia. Niistä jäi itämään kiinnostus tehdä jotain omaa alalla. Tuo kiinnostus johti ensin suomen kielen vartalo-ohjelmaan, sittemmin ajatuksiin sen hyötykäytöstä. Olenkin voinut hyödyntää ohjelmaani väitöskirjan tekemisessä, jos kohta kokonaisuudesta tuli perin toisenlainen kuin alkuun suunnittelin. Mutta suunnitelmien muuttuminen tekemisen aikana on kaiken luovan tekemisen, myös tutkimisen, edellytys.

Ryhtymiseni väitöskirjan tekoon oli onnekas sattuma, johon vaikutti merkitsevästi työni ohjaaja, akatemiaprofessori Kalervo Järvelin. Kalle nykäisi ensin hihastani keväällä 2002. Hieman yllättäen hihasta vetämisestä seurasi jotain konkreettistakin: loppusyksystä 2002 varmistui, että voisin aloittaa jatko-opintoni Tampereen yliopiston informaatiotutkimuksen laitokselle palkattuna vuoden 2003 alusta. Siitä kaikki lähti ja näiden reilun neljän vuoden aikana Kallen tarjoama ohjaus, kannustus, tietämys ja kritiikki ovat olleet erinomaisen tärkeitä työni edistymisen kannalta. Ilman Kallea työni olisi todennäköisesti kärsinyt suuremmista puutteista. Vain oma jääräpäisyyteni on estänyt minua hyödyntämästä kaikkia Kallen hyviä neuvoja, mutta vastuuhan jää kuulijalle tässä maailmassa myös Savon ulkopuolella.

Tampereen yliopiston informaatiotutkimuksen laitoksen muilta tutkijoilta, erityisesti tiedonhaun FIRE-ryhmältä, olen saanut työn mittaan monia hyviä kommentteja. Erikseen haluan kiittää laboratorioinsinööri Eija Airiota, joka on vastannut kaikesta työssä tarvitusta Unix-taustatyöstä tekstitietokantojen ja tarvittavien ohjelmien kanssa. Työn johdanto-osan englannin kielen on tarkastanut M. A. Virginia Mattila Tampereen yliopiston kielikeskuksesta.

Satunnaista kirjastoapua on antanut ystäväni Olli Louhimo. Tekniikan lisensiaatti Aarno Lehtolalta olen voinut kysyä neuvoja tietoteknisissä asioissa tarpeen mukaan. Suomen Kulttuurirahasto on tukenut työtä apurahalla.

Pietarin kaduilla ja Nikolai Gogolilla on myös osansa työn taustalla. Pietarin kaduilla oli vähäinen, joskin merkittävä osuus siinä tapahtumasarjassa, joka alkoi loppukesästä 2001 ja johti minut aloittamaan informaatiotutkimuksen jatko-opinnot Tampereella vuoden 2003 alussa. Nikolai Gogol on puolestaan kuvannut erinomaisesti absurdeja tapahtumia, jotka heittelevät ihmisiä pitkin katuja ja maailmaa.

Helsingissä syykuussa 2007

Kimmo Kettunen

Abstract

This thesis concerns use of reductive and generative methods in management of keyword variation in information retrieval with best-match retrieval systems. The main results of the thesis are related to Finnish language IR, but we present also results of Swedish, German and Russian IR.

The main contributions of this study can be summed up as follows.

Our main contribution was to show that generative methods are also appropriate for information retrieval (IR) in morphologically complex languages in a best-match retrieval environment. For Finnish we evaluated inflectional stem generation and its enhancements. We also created a new method, Frequent Case Generation, FCG, for inflectionally at least moderately complex languages and evaluated the method with four languages. The main idea of the method is to use only the most frequent nominal word forms of keywords as search terms. For three of the languages (Finnish, Swedish and German) the method was shown to yield good retrieval results when lemmatization was used as comparison. For Russian the results were inconclusive and the method should be re-evaluated with a better Russian collection. The method is based on skewness of word form distributions, and thus it is also expected to be applicable to other morphologically complex languages.

For Finnish best-match IR we have shown that besides lemmatization, also stemming, inflectional stem generation and its enhancements and most frequent case form generation of keywords yield good retrieval results when compared to the state-of-the-art, lemmatization. This broadens the spectrum of possible morphological tools for the handling of morphological variation of Finnish, which has been considered challenging in IR. As Finnish can be seen as a “worst case” language with respect to morphological variation, our results should also show the way to other languages having a fair degree of morphological variation.

Most of the methods evaluated in the study are shown to work for both long laboratory type queries and more realistic very short queries, which resemble user queries at least in the number of the keywords, although the research setting was a typical laboratory IR environment.

List of figures

Original research publications

This thesis consists of a summary and the following original research publications, reproduced here by permission.

I. Kettunen, Kimmo, Kunttu, Tuomas & Järvelin, Kalervo 2005. To stem or lemmatize a highly inflectional language in a probabilistic IR environment? Journal of Documentation 61 (4), 476–496.

II. Kettunen, Kimmo 2006. Developing an automatic linguistic truncation operator for best-match retrieval of Finnish in inflected word form text database indexes. Journal of Information Science, 32(5), 465–479.

III. Kettunen, Kimmo & Airio, Eija 2006. Is a morphologically complex language really that complex in full-text retrieval? In T. Salakoski et al. (Eds.) Advances in Natural Language Processing, LNAI 4139. Berlin Heidelberg: Springer-Verlag, 411–422.

IV. Kettunen, Kimmo, Airio, Eija & Järvelin Kalervo 2007. Restricted inflectional form generation in management of morphological keyword variation. Accepted for publication in Information Retrieval.

The studies will be referred to as Study I – IV in the introductory part of the thesis.

1. Introduction

Text-based information retrieval (IR) focuses on matches between text-based representations of human information needs and textual representations of documents. The match between the query and documents is seldom perfect, because both representations are expressed in natural language and have different origins and characteristics. IR has thus to deal with several problems: First, the query that represents the information need of the human is often indeterminate and short and thus provides little evidence for the IR system about the desired document features, which are related to document relevance. Secondly, the words used in the query may be different from those of relevant documents due to many features of natural language, e.g., synonymy and inflection. Thirdly, even useless documents may contain vocabulary similar to the query; this affects the overall effectiveness of retrieval. (cf. Belew 2000; Ingwersen & Järvelin 2005; Baeza-Yates & Ribeiro-Neto 1999).

In the present study we shall focus on monolingual textual representations of documents and queries, and their matching. Our specific focus will be on the morphological processing of documents and queries in order to derive representations that better support document-query matching. Our study is motivated especially by the morphological variability of natural languages. While much of IR research has dealt with English, English is morphologically fairly simple and findings in the English IR context may not necessarily apply to IR in other languages with different morphological characteristics. Therefore we shall contrast findings in English IR (and other languages) with findings especially in Finnish, but also in Swedish, German and Russian IR.

Multiple approaches have been utilized in IR in the management of morphological variation of text words. A common baseline is *token-based* indexing and retrieval – i.e. plain text words are used as such for the representation of both documents and queries. This approach has obvious problems when the word tokens of the queries do not match the document word tokens. A simple and traditional way to resolve the problems is to leave the document representation intact, but use a *truncation operation* on the query words to match in the index all document words having the same initial characters. In large text databases truncation tends to match too many words, making queries unmanageably long and producing too many unwanted matches. Linguistic morphological processing also can be alleviated by approximate string matching techniques, such as *n-gramming* of all word forms (McNamee & Mayfield 2004). Here words are represented as strings of varying length (the value of n is usually 2−6 characters), which makes the method language independent and generic. The main disadvantage of n-gramming is huge indexes.

Among linguistically informed approaches, one option is to apply *stemming* on both query and document words, thereby removing much of the inflectional variation (Porter 1980). Stemming may, however, conflate fairly remote words to common stems making them unspecific or failing to identify the common stem of some words in complex cases. A more elaborated and linguistically oriented approach combining stemming and truncation is *inflectional stem generation* (Kettunen, Kunttu & Järvelin 2005; Koskenniemi 1985a). Here several distinct inflectional stems are generated for one lemma before matching the token-based index. The benefit of this approach is that matching in the index is more accurate than with ad hoc stems. A further option consists of the production of all inflected word forms for query words (Arppe 1996). However, in morphologically complex languages this tends to lead to excessively long queries. The final approach is *lemmatization*, where the lemma of each document and query word is automatically identified and query word lemmas are compared to the lemma-based index. The benefit of lemmatization is that is uses full word forms and is searcher-friendly.

Lemmatization is usually based on the use of morphological rules and a large dictionary that contains the base forms or lemmas for words to be recognized. Lemmatization would be the ideal approach for handling morphology in IR except for two problems, word form ambiguity and out-of-vocabulary (OOV) words. Words out of context are frequently ambiguous but may be disambiguated. Most IR studies using disambiguation, however, have reported no or minor improvements in retrieval performance (Krovetz & Croft 1992; Sanderson 1994). Lemmatizers often cannot handle OOV words correctly, and often such words are different kinds of proper names (person names, geographical names, company names etc.), which tend to be significant words in queries. Their incorrect treatment may thus lead to severe impairment of IR performance. The problem of OOV words can be handled by relaxing of the morphological rules of lemmatizers (Alegria et al. 2002; Koskenniemi 1996; Oflazer 1996), but this, in turn, will exacerbate ambiguity.

Lemmatization has been found usually to produce the best IR results with morphologically complex languages, such as Finnish (e.g. Alkula 2000, 2001). In the present study we employ lemmatization as the gold standard for morphological processing in IR and compare the plain words' baseline and the morphologically simpler approaches to lemmatization with respect to IR performance. A well-known approach in lemmatization is Two-level Morphology (TWOL) developed by Koskenniemi (1983) and implemented in several lemmatization programs for several languages – e.g., FINTWOL, GERTWOL, ENGTWOL, and SWETWOL. The first experiments with Finnish morphological processing in IR and the TWOL software (among others) were conducted by Riitta Alkula (2000, 2001[1]). Her studies were performed in the Boolean exact-match retrieval environment. In the present study we shall focus solely on experiments in best-match IR environments.

Our main research problem in the present study is to evaluate the suitability of reductive and generative morphological methods to IR of highly inflected languages in a best-match IR environment. The main idea behind reductive methods is that varying word forms are *reduced* somehow so that relationships between keywords and index words can be detected. We refer to methods that generate inflectional stems or full word forms from a given input form as generative. Some of the reductive methods (stemming) have earlier been thought to be unsuitable for this purpose (Koskenniemi 1983, 13) and some of the generative methods used in the present study have not been evaluated at all in this context. We shall look at the following research questions under monolingual IR test condition:

1. On the general level, how do generative morphological methods compare with reductive methods? Specifically, are generative morphological methods feasible with morphologically complex languages, such as Finnish, Swedish, German and Russian in best-match IR?
2. More specifically, what is the relative retrieval performance of generative and reductive morphological methods? Can generative methods reach the IR performance of well established reductive methods, stemming and especially lemmatization, which is considered as the gold standard in the present study?
3. If generative and reductive morphological methods reach reasonably equal IR performance, what other merits should be taken into account when choosing a morphological method for managing keyword variation?
4. Laboratory type IR usually uses long queries that are made out of the topics of the collection and contain 10–20 words. This kind of queries gives some insight for the performance of different morphological methods, but the results may not be applicable to short queries. Therefore, we study both long queries and the performance of very short queries resembling queries performed by users in the web at least in the number of query words. Short queries are assumed to give insight into the suitability of different keyword variation management methods for web use.

The present study is a typical laboratory IR study that uses well established IR collections with predefined topic sets and predefined relevance assessments and many times lengthy queries (cf. Ingwersen & Järvelin 2005, 4–6). Although no real users and information needs are associated with our tests, it is assumed that

[1] *The first studies, viz. Nurminen (1986) appeared as early as in 1986.*

the results of our studies may be applicable for practical use. Such use could be development of morphological tools for web search engines that still frequently lack coverage of even basic word level variation for less used languages (Bar-Ilan & Gutman 2005). The generative morphological methods evaluated in the present study are promising candidates for this kind of use. As they are more easily implemented than lemmatizers, they could be a simpler answer to the basic level management of morphological variation of keywords for languages that do not already have a well developed set of natural language processing tools.

The rest of this study is organized as follows. Chapter 2 introduces a few basic concepts and looks briefly at retrieval models and document and query representation. Chapter 3 discusses natural language features affecting IR and management of morphological variation in IR. In Chapter 4 we discuss performance measurement of IR. In Chapter 5 we present summaries of our studies. Chapter 6 discusses results and draws conclusions.

2. Retrieval model, document and query representation

A few basic concepts related to information retrieval systems are introduced first. Next the retrieval model, document and query representation are discussed in greater detail.

2.1. Information retrieval systems

By an information retrieval system we mean a textual database system consisting of text documents and means to manage the database. Documents in the database can be searched for, and new documents can be added to the database if needed. In our case, the textual database is a full-text database containing all the original texts in full. A schematic view of an IR system is given in Figure 1.

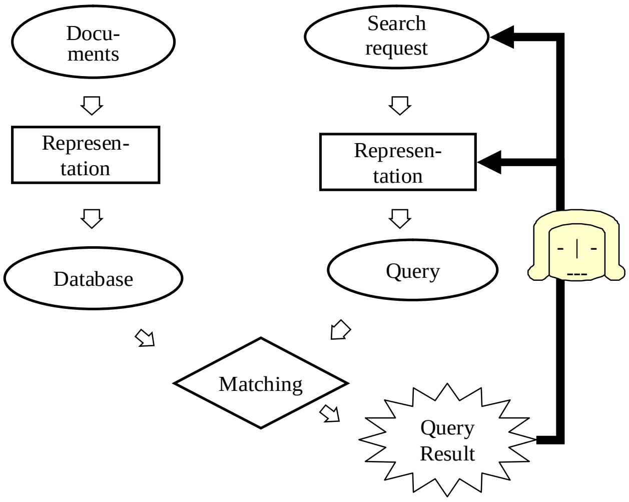


Figure 1. Schematic picture of an IR system (adapted from Ingwersen & Järvelin 2005, 115)

2.2. Retrieval models

Belkin and Croft (1987, 112) introduced a classification of retrieval models. Since the introduction of the Belkin and Croft classification, further developments of retrieval models – and also new retrieval models - have been introduced (Ingwersen & Järvelin 2005, 116−118). The main demarcation line between different models of retrieval is between *exact* and *partial matching*. Exact match retrieval systems are based on Boolean logic: the keywords of a query are joined by Boolean operators and the truth value of the query and document match are computed as a function of the operators and index text. When the query is matched against the documents, an exact match is needed; no partial results are given by the Boolean model. It is also typical for a Boolean IR system that queries need to be constructed in greater detail and care in use of connectives. A Boolean IR system does not rank the retrieved documents according to their expected relevance.[2] **(**Baeza-Yates & Ribeiro-Neto 1999, 25−27; Losee 1998, 57−60.)

A partial match (or best-match) IR system does not require an exact match of the query and documents, and thus it is able to return documents that match the query only partially. Another important feature of partial match IR systems is *ranking*: returned documents are given as an ordered list where the documents expected to be the most relevant are at the top and less relevant in decreasing order of relevance. (Baeza-Yates & Ribeiro-Neto 1999, 27−34; Ingwersen & Järvelin 2005, 119.)

The most prominent partial match retrieval models have been the vector space model (Salton & McGill 1983) and the probabilistic retrieval model (Crestani et al. 1998). A typical partial or best-match IR system allows natural language queries, where keywords can be used quite freely. It also arranges the result set according to their estimated relevance. Some best-match IR systems, e.g. InQuery, also allow strict structuring of the queries, which has been found beneficial (Kekäläinen 1999, 126**)**. Recently, probabilistic language models have been applied to best-match query systems (cf. Grossman & Frieder 2004; Metzler & Croft 2004).

2.3. Document representation

There are three main dimensions in documents: textual content, explicit structure and layout (e.g., text styles, number of columns). These document features depend mainly on domain, media, and social discourse community. IR research sees documents as collections of independent indexing features, which means

[2] *These are characteristics of the classic Boolean retrieval model. Boolean retrieval systems can also be extended to allow partial matching and relevance ranking, cf. Grossman and Frieder 2004, 67–69; Baeza-Yates and Ribeiro-Neto 1999, 38–41.*

that, in principle, each word in a document is considered as an indexing feature and stored as an access point in an inverted index – disregarding stopwords (cf. Ingwersen & Järvelin 2005).

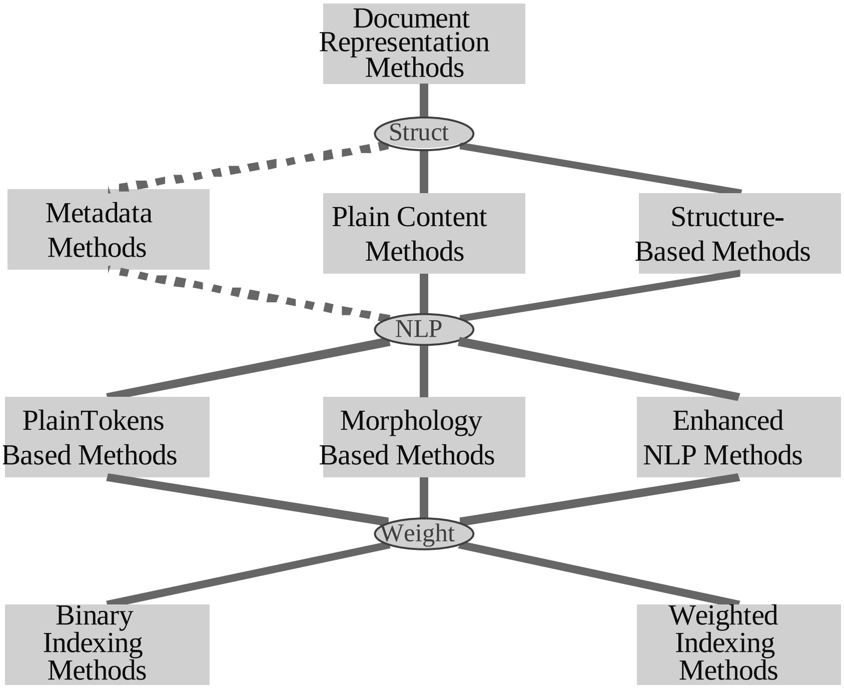


Figure 2. Classification of feature-based document representation methods (Ingwersen & Järvelin 2005, 125, figure 4.3.). Reprinted here with kind permission of Springer Science and Business Media.

Many methods have been used in document representation. Figure 2 classifies methods of feature-based document representation. Here three essential decisions need to be made: whether document structure is represented, whether natural language processing (NLP) techniques are used for manipulating document text before indexing, and whether binary or weighted indexing is used. Metadata-based methods are only interested in the structure, which means that only the metadata, e.g., bibliographic elements and keywords are represented as indexing features. This holds for the most traditional online databases and for the indexing of non-text media collections. The document may also be processed as plain content, with just the running positions of indexing features retained as in traditional full-text indexing. In more recent efforts, the hosting structural element, such as an XML path of the document, may be indexed with each indexing feature. When NLP is considered, the most traditional way is *plain token indexing*, which means that text-words as such are used as indexing features without any manipulation. Morphology-based methods cover traditional stemming and lemmatization of text words to turn them into indexing features. Enhanced NLP methods may also include processing of phrases and anaphor resolution. In traditional online systems weighting is binary, whereas best-match systems employ real non-binary weights.

In the present study we shall focus solely on NLP issues in document representation, especially the effectiveness of morphological processing of keywords and index words. We shall ignore all issues related to document structure and consider only plain texts. We will not focus on feature weighting but employ a standard probability weighting approach in all experiments. Our main focus will be on the morphological processing of features.

2.4. Indexing of texts and index types

Text indexing creates a description of the content of the original text(s) and results in a representation of the text(s). Indexes can be made manually (intellectually) or automatically. Manual indexes are usually short representations of text. Index keys can be derived from document texts, which is usually the case in automatic indexing, or from a controlled vocabulary source as in manual indexing. In automatic indexes all words of the text may function as indexing features. (Anderson & Pérez-Carballo 2001; Belew 2000, 26−29; Meadow, Boyce & Craft 2004, 93−94; more technically oriented in Baeza-Yates & Ribeiro-Neto 1999, 191−; Grossman & Frieder 2004, 182−.)

For our purposes we need to distinguish between two kinds of automatic text indexes: inflected and reduced. In an *inflected index* all the words of the text are put into the index as plain tokens, without any linguistic processing, such as stemming or lemmatization[3]. In a *reduced index* the index words are stored as either lemmatized (base forms, cf. 3.3.2.) or stemmed (stems, cf. 3.3.1.).

Compound words present a special problem for all types of indexes. A compound word (or a *compound*) is a word formed from two or more component (or *constituent*) words (Matthews 1991; Trost 2004). Often no difference is made between the compounds in which the components are written together and the compounds in which the components are written separately. In IR this distinction, however, is essential. Therefore by *compound word* we refer to the case in which the components are written together.

A reduced index may contain compound words of the text only as whole words or as whole words and also as split into component words. Splitting of compounds into components would mean that a Finnish compound, such as *kivi|talo* (components separated by |), would be represented in the index both as a

[3] *It is common to do character level normalization, such as lower casing etc., for words when they are put into an index. This is not considered linguistic processing in this context, merely common practice in index preparation.*

whole word (*kivitalo, 'stone house'*) and also as components (*kivi, 'stone'*) and (*talo, 'house')*, the components pointing to the document position that contained the compound. This kind of compound splitting has been found beneficial in the IR index creation of compounding languages, such as Finnish (Alkula 2001; Airio 2006), Swedish (Ahlgren 2004; Hedlund 2003) and German (Braschler & Ripplinger 2004). Compound splitting has not been used with inflected indexes and its suitability to these indexes is also questionable, although technically possible.

2.5. Query representation

IR research deals with queries as collections of searching features. In the laboratory research, these features have been either used as such, or often after some morphological processing such as stemming (e.g., Salton & McGill 1983) and phrase recognition (e.g., Croft, Turtle & Lewis 1991), as bags of search keys without further structural relationships. This has fostered automatic query construction from topics and enabled natural language queries.

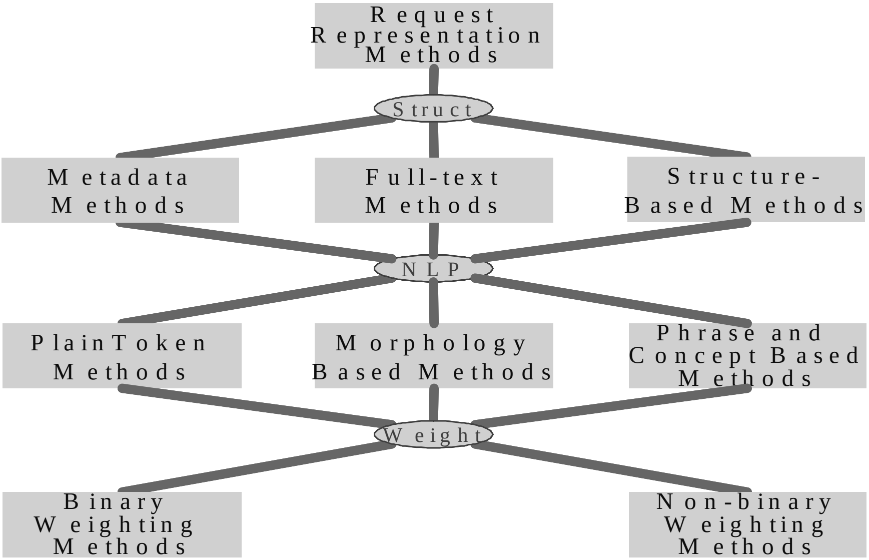


Figure 3. Classification of query representation methods for text retrieval methods (Ingwersen & Järvelin 2005, 127, figure 4.4.) Reprinted here with kind permission of Springer Science and Business Media.

Various methods have been used in query representation. Analogously to the classification of document representation methods, Figure 3 classifies methods of query representation for feature-based retrieval. As earlier, but now considering queries, one needs to make three essential decisions: first, whether structural search criteria are represented; second, whether NLP techniques are to be used for manipulating query words before searching, and third, whether binary or weighted search keys are to be used. In metadata-based methods, only the metadata, e.g., bibliographic elements and keywords are used as search keys. Alternatively, the query may be represented as plain content, as full-text keys. In more recent efforts, the required structural element, such as an XML element, of a document may be indicated for each search key. The most traditional way with regard to linguistic processing has been the use of plain word form tokens as search keys, i.e. no processing at all. Morphology-based methods cover traditional stemming and lemmatization of query words to turn them into search keys. Phrase and concept-based methods may also include phrase marking in queries and synonym set marking for keys representing the same query concept or aspect. Weighting in exact matching online systems is binary, whereas in best-match systems it is non-binary. Obviously, query representation must be compatible with document representation.

In the present study we shall focus on NLP issues in query representation, and on the effectiveness of morphological processing in particular. We shall bypass issues of weighted query keys or structural query conditions.

3. Managing morphological variation of keywords

3.1. Natural language features as problems in IR

Natural language is one of the most used means of information encoding. At present written documents are produced and stored mainly digitally worldwide. The emergence of the World Wide Web during the last 10–15 years has both greatly increased the number of digited documents and the variety of languages occuring in the web (Bar-Ilan & Gutman 2005; Grefenstette & Nioche 2000). Retrieval of digital documents has become more an everyday practice than an expert activity among information specialists. This emphasizes the need for general purpose, simple and robust linguistic tools for IR.

Table 1, *Word Level Features of Natural Language as IR Problems,* lists some of the word level natural language features that cause problems in IR, no matter whether retrieval is done in an IR laboratory setting or by a real user. Techniques to handle spelling variation, inflection, affixes, derivations, compound words and phrases already became available in the 1980s and were fairly easy to apply. Algorithmic handling of ambiguity and synonymy, for example, is still difficult, in some instances impossible. Many times the use of the more sophisticated linguistic techniques for IR has not been as beneficial as the use of simpler word or character level techniques. Table 1 is adapted with some modifications from Ingwersen and Järvelin (2005, 151). Articles on the basic language features mentioned in the text box can be found, e.g., in subchapters of Part I, Fundamentals, in Mitkov (2004). Cf. also (Daille, Fabre & Sébillot 2002; Chowdhury 2003.)

Table 1. Word Level Features of Natural Language as IR Problems

Word Level Features of Natural Language as IR Problems	
Spelling variation	Variation in spelling of words of a language may cause problems for IR. Examples of this kind of variation are historical text collections (Robertson & Willet 1993) or change of orthographic rules for the language when there have been orthographic reforms in the language.
Inflection	In most languages singular and plural nominal forms differ and there may be several grammatical cases (nominative, genitive, accusative etc.) which cause inflection of word forms. Different *affixes* – prefixes, infixes, suffixes and circumfixes – modify the meaning of the root and may hide it in retrieval.
Derivations	A root may produce several derivations, which should sometimes be conflated in IR but which sometimes have been lexicalized to the degree that the semantic connection to the root is only formal
Compound words and phrases	When compounds are written together, their headwords may be inaccessible in retrieval. Many times compounds and phrases carry meaning that is more than the product of the meaning of their constituents, i.e. they are lexicalized in their meaning. There is often instability in surface expression – ”seatbelt” vs. ”seat-belt” vs. ”seat belt”. This kind of variation may impair search results.
Ambiguity	Natural language is ambiguous due to homonymy (homography) and polysemy. With these language allows a large number of expressions through a smaller number of words, which is economical. This is advantageous to human language processing, but disadvantageous to computerized language processing, because the detection and resolution of ambiguity is laborious.
Synonymy	There are many synonymous expressions for many concepts. Acronyms, abbreviations and antonyms can also be considered special cases of synonymy. *Paraphrasing* may also be used in the absence of a specific word. This leads to situations where queries and documents may use different words for the same concept.

In the present study we focus on issues arising from inflectional morphology in document and query processing for IR. We shall also briefly consider the effects of compounding. Our main emphasis will be the Finnish language, but we shall also present results of Swedish, German and Russian retrieval.

3.2. Morphological differences between Finnish and Indo-European languages

Morphology studies word structure and formation and consists of *inflectional morphology* and *derivational morphology* (e.g., Matthews 1991; Trost 2004). The former focuses on the formation of inflected word forms from lexemes, the base elements of vocabulary. The latter is concerned with the derivation of new words from other words or root forms. Inflection is one way to express the grammatical relations between words. English and Chinese have a simple morphology, whereas many other languages, e.g., Finno-Ugric, Slavic, and Turkic languages are morphologically more complex. As IR research expanded into other languages than English in the 1990s, an expansion in morphological studies in IR was also seen in the same period (some examples of languages are given in Section 3.3.1).

The *Finnish language* is highly inflectional[4] and its vocabulary is rich in compounds. Its *inflectional* and *derivational morphology* is considerably more complex than that of the Indo-European languages like English, French or Italian. Storing Finnish text words in their inflected forms would necessitate clearly greater space for Finnish text than for English texts of corresponding length. For example, Finnish has more case endings than is usual in Indo-European languages. Finnish case endings serve the function of prepositions or postpositions in other languages (cf. Finnish *auto/ssa*, *auto/sta*, *auto/on*, *auto/lla* and English *in* the car, *out* of the car, *into* the car, *by* car). Thus Finnish is a synthetic language, while Indo-European languages are analytic (Korhonen 1994, 55; more on differences between language groups, cf. e.g., Comrie 1990).

There are 14 morphological cases in Finnish, while English has only two. As English nouns, for example, have singular and plural and two cases, altogether four distinct forms, Finnish nouns may in principle have over 2000 distinct inflected forms (Karlsson 1983, 1987). In Finnish, several layers of endings may be affixed to word stems, indicating number, case, possession, modality, tense, person and other morphological characteristics. This results in a vast number of possible distinct word forms: a noun may have some 2,000 forms[5], an adjective

[4] *In this context* highly inflectional *means mainly that the basic elements of lexicon of the language (lemmas or base forms) occur as various distinct word form variants in texts. The intricacies of morphology do not concern us here.*

[5] *The figure is a combinational calculation, and the number of forms depends on the details. A minimum number of 1872 noun forms is achieved with 2*13*6*12, where 2 denotes number (singular and plural), 13 is the number of cases, 6 the number of possessive endings and 12 the*

6,000, and a verb 12,000 forms[6]. These figures do not include the effect of derivation, which increases the figures by roughly a factor of ten (Koskenniemi 1985c).

Other Indo-European languages, such as Swedish, German and Russian are morphologically more complex than English or the Romance languages, but the number of distinct inflected forms for nouns is far less than in Finnish. Other Finno-Ugric languages, such as Estonian and Hungarian, also show a high degree of morphological complexity.

Several languages, Germanic and Finno-Ugric languages included, are rich in compounds in contrast to English, which is phrase-oriented, i.e. compounds are written apart like *motor vehicle*. For example, The Dictionary of Modern Standard Finnish contains some 200,000 entries, of which two-thirds are compound words (Koskenniemi 1983). For example the English phrase *Turnover Tax Bureau* is *liike|vaihto|vero|toimisto* in Finnish (word boundaries here marked by '|'). Compounding results in a problem of retrieving the second or later components of compounds, for example *verotoimisto* (tax bureau), if the searcher is not able to recall all possible first components.

3.3. Management of morphological variation in IR

One of the main effects of inflectional morphology is that forms of words may vary. The degree of variation may be very limited, as in English (*cat, cats, cat's, cats'*) or quite elaborate (*kissa, kissan, kissaa… kissoja… kissoissa* etc.) - altogether 26−28 forms with singular and plural forms without clitics and possessives, as in Finnish. The main problem of morphological variation for IR is that the simple principle "one keyword – one concept - one match" in the textual index does not hold due to morphology alone (we are not concerned here with other types of variation). Therefore something has to be done with morphological variation so that the performance of IR systems will not suffer.[7]

The first answers to morphological variation of keywords in IR have been (manual) term truncation and stemming. Later, lemmatization has been added to the repertoire. Generation of inflectional stems and generation of full word forms

number of clitics. If one marginal case and the variant forms for some cases are added, the figure is slightly over 2000. (Karlsson 1983, 356–357.)

[6] *This is typical for Finno-Ugric languages. Tordai & de Rijke (2005) report Hungarian as having 1400 forms for nouns, 2700 for adjectives and 59 for verbs.*

[7] *Variation may be on different linguistic levels, of which at least morphological, lexical, semantic and syntactical are of interest to IR (cf. Arampatzis et al. 2000). In the present study we shall restrict ourselves to (inflectional) morphological variation, which manifests as distinct word forms belonging to the same lexeme.*

have been used less, although they also offer a suitable solution to the problem. Galvez, de Moya-Anégon & Solana (2005, 524), for example, do not mention the possibility of word form or inflectional stem generation in their classification of term conflation methods in IR. The same goes for Frakes's (1992, 132) classification, which, however, concerns mostly stemming approaches.

In the following subchapters we shall introduce the methods that have been used for the management of keyword variation in IR either generally or specifically in the present study.[8] We shall divide these methods into two groups: reductive and generative. The main idea behind reductive methods is that varying word forms are somehow *reduced* so that relationships between keywords and index words can be detected. What we shall here call reductive methods have generally been named as conflation in the IR literature (Frakes 1992), and they include stemming and lemmatization, which are introduced in Sections 3.3.1. and 3.3.2. Methods that generate inflectional stems or full word forms from a given input form may be called *generative*. These methods are introduced in Sections 3.3.3 and 3.3.4.

[8] It should be noted that our list is not exhaustive, we do not, e.g., discuss n-gramming in any detail. Nor are different variants of stemming sub-classified as in Frakes (1992). Statistically based stemmer generation (cf. e. g., Bacchin, Ferro & Melucci 2004) is also omitted.

The classification of the keyword variation management methods used in the present study is presented in Figure 4.

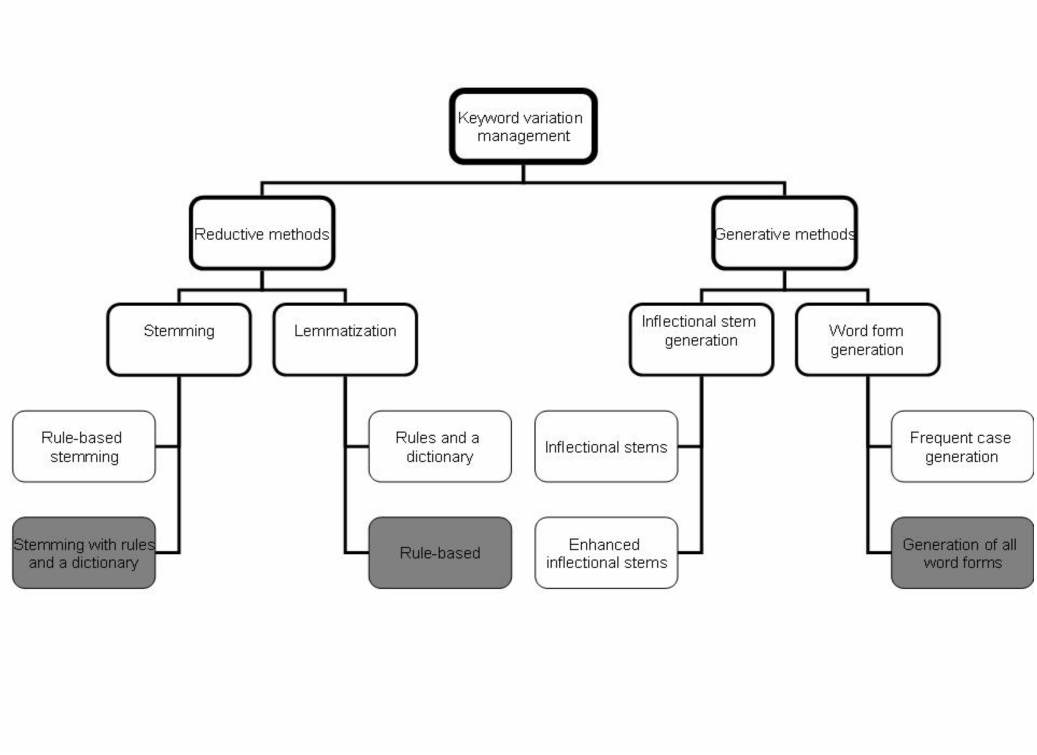
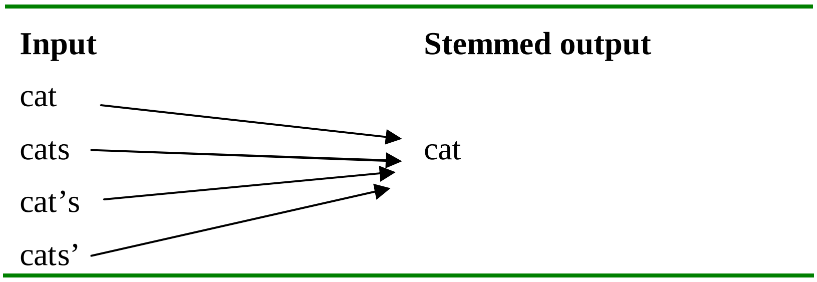


Figure 4. Classification of keyword variation management methods. Methods **not** used in the present study are marked with shaded background.

3.3.1. Stemming

Stemming has been the most widely applied reductive morphological technique in IR. In stemming distinct variants of word forms are conflated (optimally) to one form that may be a base form or just a technical stem. This many-to-one mapping is usually achieved by simple affix stripping techniques, where inflectional (many times also derivational) endings are pruned from the word forms. A simple example would be an ideal stemmer’s handling of all the variants of lexeme *cat:*



With stemming, the searcher does not need to worry about the correct truncation point of search keys. Stemming also reduces the total number of distinct index entries. Further, stemming causes query expansion by bringing word variants, derivations especially included, together.[9] The problems of stemming are under- and overstemming: too cautious or too greedy stem production. In these cases keyword variants are either not conflated or unrelated words are conflated, which leads to either recall or precision problems. (e.g., Alkula 2001; Frakes 1992; Krovetz 1993; Pirkola 2001.)

The first stemmer for English was implemented by Lovins (1968, here Frakes 1992). It was an iterative longest match stemmer that removed affixes from words according to rules. Later Porter (1980) implemented another stemmer for English. Porter’s stemmer version became one of the most used and versioned stemmers for English IR, and thus it is appropriate to briefly introduce its working principles.

Porter’s stemmer is a popular affix removal stemmer (Frakes 1992). It consists of a list of affixes and a set of rules that specify needed actions when listed affixes are encountered in input words fulfilling the conditions. Table 2 shows two examples of rules of the Porter stemmer (Porter 1980).

[9] *The term **query expansion** denotes in IR literature adding of search keys to the original query. This can be done in many ways, e.g. intellectually, automatically or interactively. Automatic query expansion can be based either on search results or different knowledge structures at hand (i.e. dictionaries, thesauri etc.) (cf. Kekäkäinen 1999, 40-). As e.g. English version of Porter stemmer handles both inflections and derivations, it actually does query expansion by bringing semantically related **derivational** forms together with the original keyword. Also the term **morphological query expansion** has been sometimes used when variant **inflectional** forms of a key are added to a query, but we do not consider this as query expansion proper. Lemmatization and generation of variant morphological full forms do not cause query expansion, but use of inflectional stems may do this, while matching of the index with the inflectional stems may also hit derivatives of the key.*

Table 2. Examples of Porter's stemmer

Rule condition (associated with the length of the word)	**Rule**	**Input word**	**Result (stem) after stemming**
	SSES à SS	caresses	caress
(m >0)	**ATIONAL à ATE**	relational	relate

Besides removing affixes, if needed, the stemmer also does some fine-tuning of the word forms (recoding) before output.

Some early IR research results with English collections questioned the effectiveness of stemming (Harman 1991). Later the results of Krovetz (1993) and Hull (1996) found stemming useful especially when long enough retrieved sets of documents were analysed. Hull also found that stemming is always useful with short queries. With short queries and short documents, a derivational stemmer was most useful, but with longer queries the derivational stemmer brought in more non-relevant documents.

In languages other than English, stemmers have been even more successful than in English text retrieval. Popovič & Willet (1992) showed that both stemming and manual truncation work well for Slovene in a best-match environment (INSTRUCT). Mayfield & McNamee (2003) evaluated stemming and different *n-gram* methods for a variety of languages including Swedish, German and Finnish. Stemming was found useful, although n-gramming was usually more effective. Sever and Bitirim (2003) evaluated three different types of stemmers for Turkish in a vector space model (SMART) and found stemming a suitable method for Turkish, which is a highly inflected language. Stemming increased search precision for Turkish by approximately 25 % when compared to no stemming at all. Tomlinson (2002, 2003) describes the results for 8−9 European languages using lexical and algorithmic stemming. Almost the same languages are covered in the study by Hollink et al. (2004). In Braschler and Ripplinger (2003) several different types of stemmers are introduced for German. Other morphologically interesting languages studied recently include Amharic (Alemayehu & Willet 2003), Arabic (e.g. Abu-Salem et al. 1999), Latin (Schinke et al. 1996), Modern Greek (Kalamboukis 1995). The morphological complexity of the languages in these studies varies, but all the studies include at least one language that is morphologically somewhat complex.

When one browses the IR literature of the 1990s and early 2000, it becomes evident that stemming has become a de facto standard method for conflation in IR. It is easy to find stemmers for over 20 languages in the IR literature. Porter stemmer implementations already exist for 14 languages (Snowball web site). Arabic, Amharic, Bulgarian, Dutch, French, German, Greek, Hebrew, Italian, Russian, Spanish, Swedish and Turkish are among the languages for which different types of stemmers have been developed for IR use.

3.3.2. Lemmatization

Lemmatization is another reductive technique: for each inflected word form in a text, its basic form, the lemma, is identified. Lemmatization is usually based on the use of morphological rules and a large dictionary giving the lemmas and information about the words. The benefits of lemmatization are the same as in stemming. In addition, when basic word forms are used, the searcher may match an exact search key to an exact index key. Such accuracy is not possible with often ambiguous stems. Homographic word forms cause ambiguity (and precision) problems – this may also occur with inflected word forms (Alkula 2001). Another problem is words that cannot be lemmatized, e.g., foreign proper names, because the lemmatizer’s dictionary does not contain them. Such problem words need special handling.

Tools for handling compounds depend on the respective languages: in compound-rich languages the morphological problem of compound splitting corresponds to the syntactical problem of phrase recognition in non-compounding languages. Morphological NLP tools for stemming, lemmatization and compound splitting are an aid to the searcher: the searcher needs not consider all word form variations or compounding and may use simple words or plain natural language text in query formulation. In best-match IR systems, which usually lack the search key truncation operator, the lemmatization of index word forms is essential for users if the collection language is morphologically complex. However, a query in a basic word form index has to be constructed with care in order not to lose derivatives, which one may cover by truncation in a conventional index – if truncation is available in the search system. While lemmatization with compound splitting seems to slightly improve retrieval performance in Boolean (Alkula 2001) and best-match retrieval (Kunttu 2003), the most important effects may be the simplification of query formulation. As Galvez et al. (2005) emphasizes, the use of whole words is also beneficial for further processing, where stems or other word parts may not be sufficiently informative.

3.3.3. Inflectional stem generation

Inflectional stem generation is a method, in which from the given basic form of a word one or several inflectional stems are generated (Koskenniemi 1985a; Alkula 2000; Study I). For example, from the Finnish base form *nainen* ('a woman'*)*, the following stems can be generated: *nainen, naise*, *naisi*, *naist,* each of the stems matching several full inflected forms. These can then be used as search keys in an inflected text index. As the stems are longer and more specific than one reductive stem (in this case hypothetically *nai*), the search is now more precise.

The method suits a language, in which alterations in word stems are many times so numerous that manual truncation by a lay user would be error prone (cf. Alkula 2000). Users may be assumed to be able to give base forms for keywords, because they are able to use a normal printed dictionary, which is based on the listing of lexemes in alphabetical order.

3.3.4. Inflectional form generation

By inflectional form generation we mean the generation of all possible inflected forms of a word when its base form is known. This method has not been much evaluated or used as a method of keyword variation management in IR, although experience in computational linguistics from the 1980s already showed that word form generators for different languages are relatively easily implemented (cf. Holman 1998; Lassila 1989; Koskenniemi 1985b for Finnish generators). The following example shows the full case inflection of the Finnish noun *kissa* without possessive endings and clitics.

Input	**Output**
kissa ('a cat')	kissa, kissan, kissaa, kissana, kissaksi, kissassa, kissasta, kissaan, kissalla, kissalta, kissalle, kissatta, kissat, kissojen, kissoja, kissoina, kissoiksi, kissoissa, kissoista, kissoihin, kissoilla, kissoilta, kissoille, kissoitta, kissoineen, kissoin kissain (an alternative plural genitive, infrequent)

The main obstacle to using all the inflected word forms of a keyword, at least for morphologically very rich languages, is that index searching with many keyword forms may be slow. But as we show in Study IV, for many European languages the number of the variant forms is not very great. For languages with large number of variant forms, a restricted version of generation, FCG (Frequent Case Generation), is the alternative, as shown in Studies III and IV.

As an example of word form generators, we discuss shortly programs that generate Finnish word forms. Generators for Finnish have been implemented since early 1980s (cf. Koskenniemi 1985b; Holman 1988; Lassila 1988). WGEN reported in Koskenniemi (1985b) was the first implementation of a Finnish word form generator – implemented already "during the winter 1981−82". It covered all the parts of speech for Finnish words. It used no explicit dictionary and the generation of word forms was based on the forms of the words and rules. As input it needed the part of speech of the word, its base form and morphosyntactic codes for the desired output form. Given e.g. as input the string *N:kamputselainen.PTV.PL* it would produce the plural partitive *kamputselaisia (Cambodians)* (Koskenniemi 1985b, 64).

Holman's Finnmorf (1988) is a similar kind of program that produces full inflected paradigms of words for language learning purposes. It also covers all the major parts of speech, nouns, verbs and adjectives, and uses no dictionaries in generation. Lassila (1988) reported of a generator named Formo that was modularized to make the generation as efficient as possible.

Common to all of these programs is that they use no lexicons in generation and are still able to produce accurately all the variant forms for nouns, adjectives and verbs. This is based on the regularity of Finnish word forms: when certain problems in the inflectional stem formation of the word have been taken care of – e.g. grade alteration and different types of plural formation for nouns – full inflected forms can be produced by just concatenating right affixes to the stems.

3.3.5. Differences and similarities between reductive and generative methods

In the present study stemming, lemmatization, inflectional stem generation and word form generation are considered different means for keyword variation management. They all have their merits, although the procedure of unifying variant forms may be reversed and the degree of unification varies.

In generative methods, inflectional stem generation and inflectional form generation, the point of departure is one given form, usually the base form. Out of this, several forms, either inflectional stems or full word forms are generated. These are then matched to the inflected textual index. The process is shown in Figure 5.

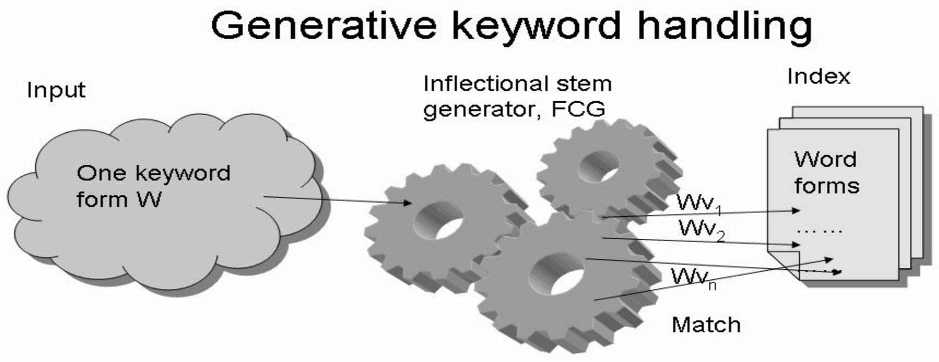


Figure 5. Generative keyword handling. FCG = Frequent Case (form) Generation

In reductive methods, lemmatization and stemming, the procedure is the reverse: given multiple forms that morphologically belong together, the forms are reduced (optimally) to a single form, which is either a lemma or a stem. These forms are then matched to a textual index that has been processed in the same way, i.e. either stemmed or lemmatized. The process is shown in Figure 6.

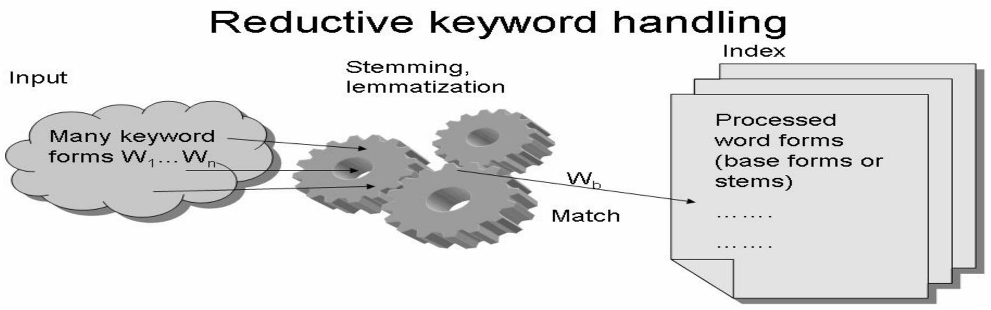

Reductive keyword handling
Input
Many keyword forms W$_1$…W$_n$
Stemming, lemmatization
W$_b$
Match
Index
Processed word forms (base forms or stems)
…….
…….

Figure 6. Reductive keyword handling[10]

[10] *The figure is simplified such that it excludes ambiguous word forms that will produce multiple analyses.*

The first stemmers (e.g. Lovins 1968; Porter 1980) were rule-based. Typically they had a small set of rules (from tens of rules to a few hundred) and used simple affix lists to detect suitable strings to be pruned from keywords. Simple reformulations for remaining stems were also performed (Frakes 1992). The process of stemming frequently resulted in inflectional and derivational unification: an example of this is Snowball's handling of the words *generalizations*, *generalize* and *general* which all are unified to the same stem, *general.*

Because of the known problems of stemming, mainly over- and under-stemming, Krovetz (1993) added a dictionary to his K-stemmer of English. The reason for the dictionary was to ensure that the suggested stems were indeed existing words, not just truncated strings. Krovetz (2000) reported that the retrieval effectiveness of the lexical stemmer was same as that of a Porter stemmer, but found the use of full forms in stemming beneficial.

After Krovetz (1993), stemmers with large lexicons have also become common. Lemmatizers without lexicons have been implemented (e.g. for Swedish, Hellberg 1972), but the use of a lexicon has been the norm with lemmatizers. The deduction of a base form for an inflected word is in principle possible without a lexicon (cf. Jäppinen & Ylilammi 1986), but leads to increased ambiguity in possible base forms. The use of large lexicons in stemmers in 1990s has made lemmatizers and stemmers more akin (Jacquemin & Tzoukerman 1999).

Thus Figure 6 can be enriched with respect to the use of a lexicon, which is optional as in Figure 7.

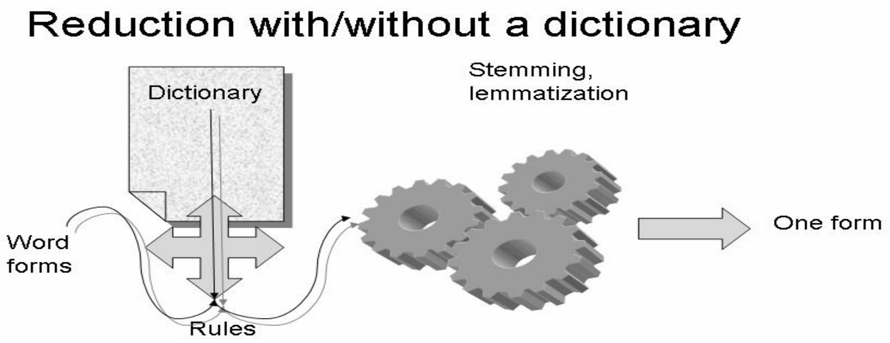


Figure 7. Reduction with/without dictionary

Word form generators can be implemented more easily without large lexicons (for Finnish e.g., Koskenniemi 1985b; Holman 1988; Lassila 1988). As rule-based programs they are more robust and not affected by out-of-vocabulary words.

3.4. Conclusions

All the methods for handling of keyword variation can also be compared on a more general level. Three kinds of benefits are usually associated with different types of morphological processing in IR (Harman 1991). They are briefly as follows:

- ease of use (the morphology of query words is taken care of by the retrieval system),
- storage savings (smaller indexes when lemmatization or stemming is used), and
- improved retrieval performance.

With these criteria reductive methods do well. The user’s burden in choosing the form of the input keyword is reduced to a minimum and storage savings and improved retrieval performance are clearly achieved. Generative methods lack storage savings, but improve retrieval performance, as has been shown in the separate studies of this thesis. Ease of use with the types of generative methods suggested in the thesis should not be a great problem, supposed that the user is able to give the key words in their base forms.[11]

Besides these criteria, there are, however, others that should be taken into consideration. Lexical coverage of the morphological method used is also important. This is an issue that may affect lemmatizers using dictionaries.[12] Their dictionaries will lack words for many reasons, and one of the main classes of lacking words will be proper names, which are usually an important subclass of query words (Pirkola & Järvelin 2001). Thus lemmatizers will need some augmentation for the handling of OOV words. Such augmentations include fuzzy string matching techniques, which have been used successfully in Cross Language IR (Hedlund 2003).

[11] *In a real interactive search system lemmatization could also be used to make sure that keywords given by the user are in base form before generation of search variants.*

[12] *On the basis of FINTWOL analysis of the 32 million word token HUT Corpus (Creutz & Linden 2004; Creutz 2005), the approximate percentage of unknown words on type level is about 17.6 % and on token level 5.4 %. It is evident that some of the unknown words are misspellings, but the percentages clearly show the existence of the problem.*

Other criteria can also be used for comparison, for example, the retrieval time and indexing time of the database (cf. Järvelin 1995, 52−53). When all these considerations are taken into account, the strengths of reductive and generative methods seem more equal. Although lemmatization usually achieves the best mean average precision, its usefulness is impaired by the following factors:

- use of a large dictionary that needs updating,
- problems caused by words missing from the dictionary of the lemmatizer (Alkula 2000; Koskenniemi 1996),
- longer implementation time for the lemmatizer if the language does not have such already,
- base form runs needed for the database indexes (Galvez, de Moya-Anegón & Solana 2005).

With the use of generative methods these biases are avoided. A rule-based stemmer is weakened only by the fourth point on the list; a stemmer with a dictionary will suffer from all the same biases as a lemmatizer. Generative methods, in turn, have problems of their own. One of the most serious of these is the matching of unwanted words, which results in too long and slowly processed queries when inflectional stems are used as search terms. These problems are discussed in Studies I and II.

4. Evaluation of IR performance

4.1. Performance measures

Evaluation of IR performance has been studied widely in the laboratory IR setting. IR performance can be measured in many ways, but the most common measures for retrieval performance, however, are precision and recall values and especially mean average precision. All of these are based on the concept of relevance, and imply that documents can be judged as relevant or non-relevant with respect to the query. (Hull 1996; Losee 1998; Baeza-Yates & Ribeiro-Neto 1999.) In this part we introduce the performance measures used in the present study and notions of relevance. Other, less popular performance measures, are introduced by Korfhage (1997), Losee (1998) and Ingwersen & Järvelin (2005).

4.2. Relevance

When textual queries are formulated, users expect to get somehow useful documents as a response to their query from the retrieval system. This usefulness is frequently called *relevance*, which can me measured and partitioned in different ways. Relevance has been characterized in many ways, but no simple all-purpose definition of relevance has been found. The most used relevance notions have been the following two:

- System or algorithmic relevance. This is the relation between a query and documents in a given IR system with a given procedure or algorithm
- Topical or subject relevance. This is a relation between the subject expressed in a query and the subject covered by the documents in the system (aboutness).

Cognitive, situational and motivational relevance have also been widely used in the IR literature (Saracevic 1975; Mizzarro 1997; Cosijn & Ingwersen 2000). In our study the notion of relevance used is that of topical relevance, which is often considered a prerequisite for other kinds of relevance, too, and is typical in laboratory oriented IR studies.

While relevance, in principle, has multiple degrees, one may partition relevance according to a graded scale or a binary scale, where documents are either relevant or non-relevant.

In the present study we have been using collections that have both graded relevance (the Finnish collection TUTK) and binary relevance (CLEF 2003 and 2004 collections for Finnish, German, Russian and Swedish).

4.3. Recall and precision

The *recall* of an IR system is a measure of the ability of the system to present all relevant items. The *precision* of an IR system is a measure of the ability of the IR system to present only the relevant items. In the IR literature recall and precision of retrieval are defined in Losee (1998, 81−82), Baeza-Yates & Ribeiro-Neto (1999, 74−75), Meadow, Boyce and Kraft (2000, 322−323) and Hull (1993).

Recall and precision are computed as follows:

Recall = r/R

Precision = r/n

where r = number of relevant documents retrieved,
R = total number of relevant documents and
n = number of documents retrieved.

If, for example, 100 documents were retrieved and were relevant and there were 500 relevant documents in the database, we would say that recall was 0.20 (or using percentages, 20 %). When all the retrieved documents are relevant, precision is 1.0 (or 100 %); when no relevant documents are among retrieved documents, precision is 0.0 (or 0 %). We will use real numbers in the range 0.0 – 1.0 and their equivalent percentages (0% - 100%) interchangeably for expressing precision.

Precision and recall are known to have an inverse relationship between them. If the precision of retrieval is high, its recall is usually low and vice versa. It should, however, be mentioned that neither precision nor recall depend on the other, but are jointly dependent on how the retrieval was carried out and the relevance values of the documents assigned. (Meadow, Boyce & Kraft 2000, 324.)

4.3.1. Precision at 10 documents

A specific case of precision is *precision at 10 documents* (marked with *P@10* or *P(10).* This measure “counts the number of relevant documents in the top 10 documents in the ranked list returned for a topic” (Buckley & Voorhees 2004, 26). It is a measure that correlates closely with user satisfaction in tasks such as web searching and it is also easy to interpret. Its weakness is that it is not a powerful discriminator between retrieval methods. For these reasons it has a much larger margin of error than mean average precision for instance (MAP, cf. 4.3.3.). (Buckley & Voorhees 2000; Buckley &Voorhees 2004.)

4.3.2. Precision at standard recall levels

Precision-recall graphs have been a frequent form of presentation for retrieval performance data. These graphs visualize the average progress of a group of searches “by studying the precision and recall at a number of different points” (Losee 1998, 83; cf. also Voorhees 2004a, 2004b). Recall is usually plotted on the x axis and precision on the y axis of the graph. Figure 8 shows a typical precision-recall graph where performances of different methods are shown together.

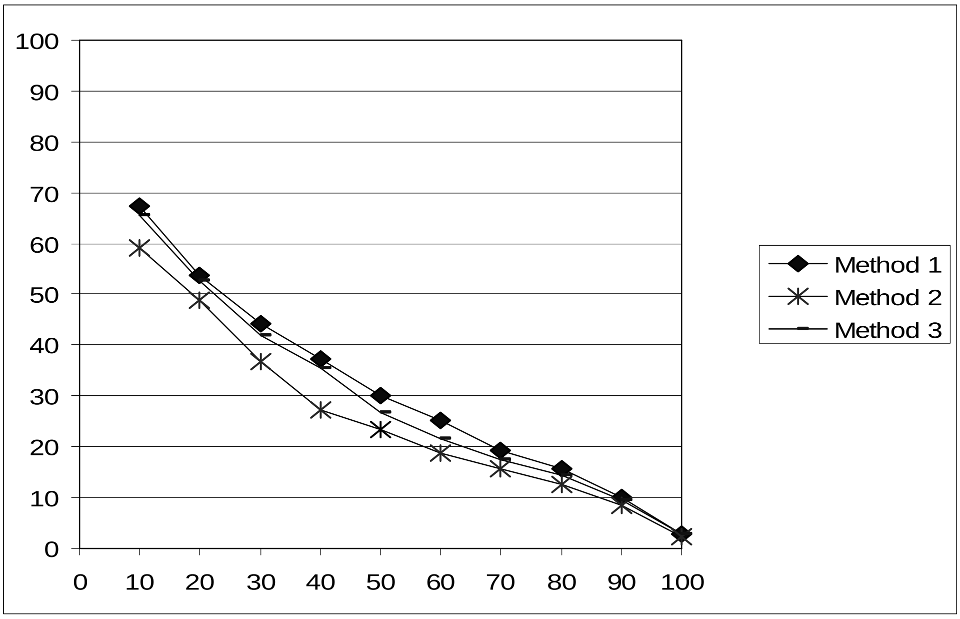


Figure 8. Precision-recall graphs

In Studies I – III we used statistics given by *deval* program of Inquery. Deval gives the 10 recall points that are plotted on the sample graph (Figure 8). As a summary measure we used the interpolated average precision over 10 recall levels, averaged across queries. We call this summary measure mean average precision (interpolated). In Studies I and II this was called 'average precision over recall levels', which should be obvious in the context.

4.3.3. Mean average precision

Mean average precision (MAP) is the average of average precision values over several topics. It is commonly used as an overall summary measure for IR tests. Kraaij (2004, 87−88) defines MAP as follows, giving first the definition of *average precision* (AP) for an individual query and defining MAP across queries with it:

> “The average precision for a certain query and a certain query system version can be computed by identifying the rank number n of each relevant document in a retrieval run. The corresponding precision is defined as the number of relevant documents found in the ranks equal or higher than the respective rank r divided by n. Relevant documents which are not retrieved receive a precision of zero. The average precision for a certain query is defined as the average value of the precision over all relevant documents. The mean average precision can be calculated by averaging the average precision over all queries (macro-average).”

A mathematical equation for MAP is given in Kraaij (2004, 88) as follows:

$$\text{MAP} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{Nj}\sum_{i=1}^{Nj} pr(d_{ij}) \quad \text{where pr}(d_{ij}) = \begin{cases} \dfrac{r_{n\,i}}{n_{\,i}} & \text{if } d_{ij} \text{ retrieved and } n_i \leq C \\ 0 \text{ in other cases} & \end{cases}$$

Where

n_i denotes the rank of document d_{ij} which has been retrieved and is relevant for query j

r_{ni} is the number of relevant documents found at ranks 1-i

N_j is the total number of relevant documents of query j

M is the total number of queries

C is the cut-off rank (usually 1000).

In Study IV we used data given by the *trec.eval* program, which gives 11 recall points for drawing the graphs and mean average precisions as the summary measure for comparisons. We call this measure *mean average precision (non-interpolated)*, or briefly, *MAP (non-interpolated)* in the tables.

The main disadvantage of MAP is that it is sensitive to topics with only a few relevant documents. If a document’s rank position changes even slightly, this may have a great effect on the average precision of the query and thus also indirectly affect the mean average precision of all the queries (Kraaij 2004, 88).

In any case, it has been shown that MAP gives reliable results in different evaluation measurements (Tague-Sutcliffe & Blustein 1995; Buckley & Voorhees 2000; Buckley & Voorhees 2004b), and thus it has been used as a standard measure in IR.

4.4. Statistical validation of results

The main purpose of a laboratory IR test setting is to find significant differences between the methods or systems evaluated. The research setting is arranged so that associations between the dependent variables (such as MAP) and independent variables (such as certain linguistic handling of keywords) of the test setting can be identified. When the results of the comparisons between different methods are obtained, statistical tests are used to show whether the differences between the methods are significant.

Statistical testing of the results of IR experiments is known to be problematic for several reasons. The main causes for the problems are sampling and sample sizes, the number of queries especially is often problematic. Another factor is that the sampling of the queries is not random; queries form merely a selection, not a random sample. (Robertson 1981.)

For these reasons the choice of standard statistical tests with IR experiments is somehow problematic. Parametric basic tests, such as the paired t-test, compare two methods against each other and have stronger assumptions about the data. In IR experiments usually more methods are evaluated, making the use of parametric tests not very suitable in many cases. Therefore non-parametric tests are often recommended. (Robertson 1981.)

The statistical testing of the differences between methods used in the present study was done using the Friedman test (original Friedman test, cf. Siegel and Castellan 1988, modifications used here in Conover 1980). The main reason for this was that in all studies of the thesis, multiple methods were compared to each other. Kekäläinen (1999, 100−101) found that in such a context, the Friedman test is appropriate.

The Friedman two-way analysis of variance by ranks is a generalization of the parametric sign test. Thus it offers a non-parametric alternative for comparing more than two related samples (Hull 1993).The basic principle of the Friedman test is to first calculate whether there are significant differences overall between the methods evaluated. If such differences are found, a pair-wise comparison between different methods is done to show which methods differ significantly from each other.

More specifically, the Friedman test verifies whether *k* related samples or repeated measures come from the same populations or populations with the same median. To test this, the data are cast in *b* rows and *k* columns, where rows represent queries (units) and columns represent respective treatments. The test itself is based on ranks. The scores of each row are ranked from 1 to *k*, and the Friedman test determines the probability that the rank totals for each variable (or column *k*) differ significantly from the values that would be expected by chance. (Conover 1980, 299−.)

5. Summary of the studies

This section presents a summary of the empirical studies of this thesis. The research problems, methods and results will be briefly summarized. Section 5.1 first introduces the retrieval systems, collections and morphological programs used. Section 5.2 introduces Studies I and II, which examined inflectional stem generation based retrieval versus reductive conflation methods for Finnish. Section 5.3 introduces Studies III and IV, which develop and evaluate frequency based inflectional form generation method (FCG, frequent case generation) for IR.

5.1. Test setting: search systems and collections

5.1.1. The probabilistic retrieval model - InQuery

Probabilistic retrieval systems have been used in IR since the introduction of the probabilistic retrieval model in the 1970s (Crestani et al. 1998). Concisely put, a probabilistic retrieval model computes the similarity coefficient between a query and a document as the probability that the document will be relevant to the query. Keyword weights and their estimation are the essential components of a probabilistic retrieval system.

In our studies we have mostly been using (with one exception) the InQuery retrieval system (Callan, Croft & Harding 1992; Broglio, Callan, Croft & Nachbar 1995). InQuery is a specific type of probabilistic retrieval system and it is characterized, among other things, by the following properties:

- It is based on the Bayesian inference network retrieval model, which has been widely used in IR since the early 1990s (cf. de Campos, Fernández-Luna & Huete 2004).
- It allows the use of highly structured queries.
- It uses a variant of the tf*idf (term frequency * inverse document frequency) formula for estimating the probability that a document is about a concept (cf. Robertson 2004 on inverse document frequency generally; InQuery’s variant is described in detail in Allan et al. 1997; cf. also Kekäläinen 1999, 27−28).

The Bayesian inference net retrieval model is based on the Bayesian probability theory. The main properties of a Bayesian inference network are as follows. The network of the model is a directed acyclic graph (DAG). The nodes of the graph represent propositional variables and the arcs represent dependencies or causal relationships between nodes. The value of a network's node is a function of the values of the nodes it depends upon (parents of the node). The bottom level nodes (leaves) of the DAG usually represent propositions whose values can be determined by observation. Other nodes typically represent propositions whose values must be determined by inference. These values are not necessarily absolute, and their certainty or probability can be represented by weights on the arcs. (cf. Callan, Croft & Harding 1992; Crestani et al. 1998: Grossman & Frieder 2004.)

The inference network model of InQuery can be shown schematically as in Figure 9; the leaves of the DAG are at the top of the figure (adapted from Grossman & Frieder 2004, 63; cf. also Callan, Croft & Harding 1992; Crestani et al. 1998).

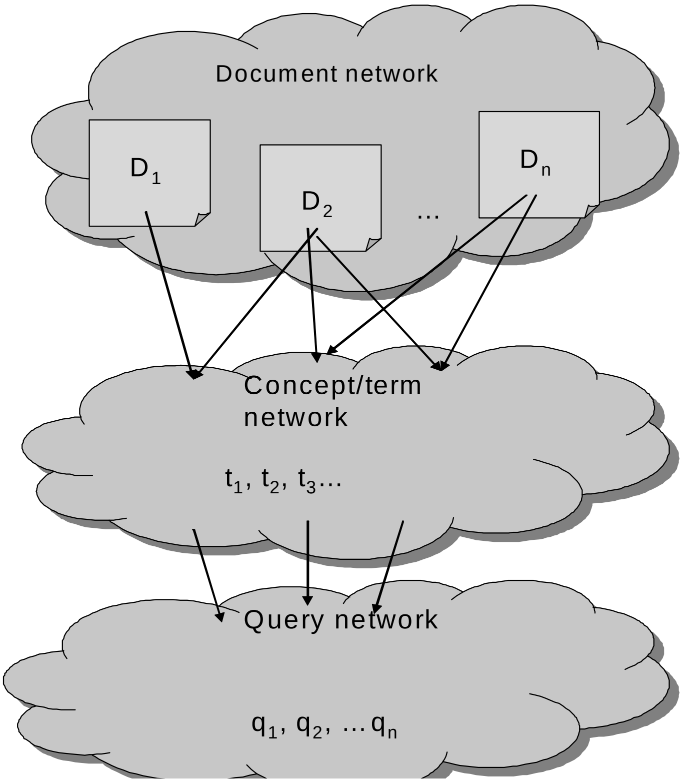


Figure 9. Inference network retrieval model (InQuery)

In Study IV we used the Lemur retrieval system for the Russian collection. Lemur combines an inference network retrieval model with language models, which are claimed to give more reliable estimates for word probabilities in documents (Metzler & Croft 2004; Grossman & Frieder 2004, 45–). One of the key benefits of the approach is, that “the resulting model allows structured queries to be evaluated using natural language estimates” (Metzler & Croft 2004).

5.1.2. Finnish test collections

Studies I and II used the same Finnish test collection, TUTK (Sormunen 2000) and the same retrieval system (InQuery). TUTK contains 53, 893 Finnish articles from three newspapers 1988–1992. The articles of the database are on average fairly short. Typical text paragraphs are two or three sentences in length and the average length of the articles is circa 218 words. Relevant documents of the collection, however, are longer, over 300 words depending on the relevance level (Sormunen 2000). The topic set used consisted of 30 topics[13]. Topics are long: the mean length of the original topics is 17.4 words.

Besides the TUTK collection we also used the Finnish CLEF 2003 collection in Study III and also briefly in Study IV. The Finnish CLEF 2003 collection contains 55, 344 articles from the Finnish newspaper *Aamulehti* 1994−1995. The collection has 45 topics with relevant documents.

In the first study we used the following morphological programs: FINTWOL by Lingsoft Ltd. (for lemmatization), MaxStemma (for inflectional stem generation), and the Finnish Snowball stemmer which is freely available on the web (Snowball web site; Porter 2001). MaxStemma inflectional stem generator was implemented by the present author in the early 1990s. Its original version is described in more detail in (Kettunen 1991). Other generational methods were simulated, as explained in Studies II and III. FINTWOL and Snowball were used in all of the studies.

5.1.3. Relevance thresholds in TUTK

The relevance assessments of TUTK have been described in detail in Sormunen (2000). We do not replicate them here, but only present the four-point scale used and its interpretation in Table 3 (Sormunen 2000, 63).

[13] *This is a subset of the whole set of TUTK topics, which amounts to 35 topics (Sormunen, 2000). The topic set used is the same as in Kekäläinen (1999).*

Table 3. Relevance levels of TUTK interpreted.

Relevance level	Document is
0	totally off target
1	marginally relevant, refers to the topic but does not convey more information than the topic description itself
2	relevant, contains some new facts about the topic
3	highly relevant, contains valuable information, the article's main focus is on the topic

This relevance scale has been partitioned or combined in our studies in more than one way. In the first testing environment of Study I query performance was evaluated on two binary relevance scales. The first one was created from the original four levels by combining levels 2 and 3 as relevant; levels 0 and 1 were considered irrelevant. This was called the normal scale. The second scale was created by considering the relevance level 3 as relevant and the levels 0−2 as non-relevant. This binary scale was called the stringent scale.

In the second testing environment of Study I, the relevance scales of the TUTK collection were used in a slightly different way. In this environment the liberal scale included all the relevance levels 1–3 as relevant and the normal and stringent scales were the same as in the first setting. In other studies the relevance scales of the TUTK collection were used in the same manner as in the second testing environment of Study I.

By combining multiple relevance levels differently, different aims can be achieved. Users are usually mostly interested in highly relevant documents (in TUTK's relevance level 3). If some retrieval method is able to pick more highly relevant documents than other methods, it has an advantage.

5.1.4. Swedish, German and Russian collections

In Study IV for Swedish and German we used the CLEF 2003 collections and for Russian the CLEF 2004 collection. InQuery was used as a retrieval system for Finnish, Swedish and German. For Russian we used Lemur retrieval system (Metzler & Croft 2004), because it provided better handling of the UTF-8 characters used for encoding the Russian texts. Table 4 describes the collections, their sizes, the number of topics and the retrieval systems.

Table 4. Swedish, German and Russian collections used in the study

Language	Collection	Collection size (docs)	Topics with relevant documents	Retrieval system in tests
Swedish	CLEF 2003	142 819	54	InQuery
German	CLEF 2003	294 809	56	InQuery
Russian	CLEF 2004	16 716	34	Lemur

For Swedish and German we used lemmatizers from Lingsoft Ltd. (SWETWOL and GERTWOL). Snowball stemmers for Swedish, German and Russian were also utilized in the study. The generation of keyword forms was simulated as explained in Study IV.

5.2. Summary of inflectional stem generation studies for Finnish

5.2.1. Research problems

Two studies were performed to evaluate the effects of different types of inflectional stem generation for Finnish text retrieval. The research problems of Study I were:

1. How do lemmatization and inflectional stem generation compare in a probabilistic environment?
2. Is a stemmer a realistic alternative for handling of the morphology of a highly inflected language, such as Finnish, for IR?
3. Is simulation of truncation feasible in a best-match system?

As sub-problems compound splitting, derivational queries, and different types of topics were also studied.

In Study I we found some problems with the use of inflectional stems. The main problems of the inflectional stems were long queries that were slow to run. In order to remedy these problems we developed the inflectional stem generation method further in Study II by enhancing the stems with possible continuations of the stems, i.e. with parts of case endings or whole case endings. In principle, enhanced stems offer better precision in the matching of the words of the inflected index. The research problems of Study II were:

1. Do enhanced inflectional stems make the IR performance of the inflectional stems better in an inflected full text index in best-match retrieval?
2. Do long and short queries behave differently with different keyword normalization methods?
3. Do enhanced inflectional stems produce clearly shorter queries that are easier to run without compromising the P/R performance?

5.2.2. Methods

The methods and basic concepts relevant to these studies have been presented in the previous sections. Section 4 discusses the evaluation of retrieval effectiveness in general, and the notions of relevance and performance measures used in the study are explained there. The morphological methods used in IR experiments of the study are discussed in Chapter 3: stemming is discussed in 3.3.1, lemmatization in 3.3.2., inflectional stem generation in 3.3.3., and inflectional form generation in 3.3.4. Best-match retrieval in a probabilistic retrieval system is introduced in Section 5.1.1.

5.2.3. Results

The main results of the monolingual Finnish IR experiments of Study I are presented in Table 5. It can be seen that lemmatization with FINTWOL performs slightly better than inflectional stem generation with MaxStemma. The performance of the Snowball stemmer is clearly inferior to the first two. The poorest performance was achieved for Plain words, i.e. words that were taken straight out of the topics to queries as keywords. On the average Plain words achieved 54.0 % of FINTWOL’s performance.

Table 5. Performance of monolingual Finnish runs on normal relevance scale

Morphological tool	**Mean average precision over recall levels % - interpolated**	**Change % w.r.t FINTWOL**
1. FINTWOL	35.0	--
2. MaxStemma	34.2	-2.3
3. Snowball	27.7	-20.9
4. Plain words	18.9	-46.0

In the present study we focused on the effectiveness of reductive morphological processing and generative morphological processing in monolingual IR. In our tests we found that, in Finnish IR, lemmatization by FINTWOL outperforms other approaches, in particular plain words and stemming, while inflectional stem generation approaches the performance of lemmatization. Their difference in performance was not statistically significant. However, in the latter approach, the index must be harvested for full words matching the generated stems. Thus queries tend to become unmanageably long. Final queries in our tests had 934−24, 443 query words after index harvesting with inflectional stems.

In Study II we developed our stem generation method further. We found that by extending the inflectional stems by regular expressions, query length can be dramatically reduced with only a minor loss in performance. Inflectional stems of Finnish were enhanced with regular expressions containing either one character from each possible case ending after the stem or full case endings. We evaluated long queries (average length of the query 14.6 words) and very short queries (average length of the query 2.9 words) on three relevance levels to see whether the proposed stem enhancement methods improve the effectiveness and efficiency of the queries.

The results showed that enhanced inflectional stem methods produced slightly lower mean average precisions than inflectional stems or lemmatization. All the enhanced inflectional stem method variants performed better than stemming with the Snowball stemmer, although the differences were not always statistically significant (cf. average precisions in Table 6). With long queries differences between different systems were slightly greater than with very short queries, and there were more statistically significant differences between systems especially on liberal and normal relevance scales. Although the proposed inflectional stem enhancing method did not outperform inflectional stem generation or lemmatization, the performance of the four variant systems was consistent and reliable with both long and very short queries. The main beneficial factor with the proposed method with respect to inflectional stems was that it resulted at best in much shorter queries than the usage of inflectional stems. With RegStemma+ the queries were about 53 % of the length of the full inflectional queries, with FullRegStemma* about 33 % and with FullRegStemma+ only about 18 %. Thus the queries were more manageable and easier to run than with inflectional stem generation in Study I. As this was achieved with a relatively small loss of average precision, especially on the stringent relevance level, the method is of interest, e.g., in web search environment, where the capability for picking the most relevant documents is needed most. Table 6 (cf. Table 5 in Study II) condenses the results of Study II for long queries.

Table 6. Performances of different methods on three relevance levels.

	Liberal relevance	**Normal relevance**	**Stringent relevance**
	Mean average precision over recall levels (%) - interpolated	Mean average precision over recall levels (%) - interpolated	Mean average precision over recall levels (%) - interpolated
FINTWOL	37.8	35.0	24.1
MaxStemma	37.3 (-0.5 %)	34.2 (-0.8 %)	22.6 (-1.5 %)
RegStemma*	35.6 (-2.2 %)	33.4 (-1.6 %)	23.1 (-1.0%)
RegStemma+	34.8 (-3.0%)	32.5 (-2.5 %)	22.9 (-1.2 %)
FullRegStemma*	33.6 (-4.2 %)	31.5 (-3.5 %)	22.7 (-1.4 %)
FullRegStemma+	32.4 (-5.4 %)	30.0 (-5.0 %)	21.4 (-2.7 %)
Snowball	29.8 (-8.0 %)	27.7 (-7.3 %)	20.0 (-4.1 %)

Here MaxStemma is the original inflectional stem generator, and RegStemmas and FullRegStemmas simulated versions of enhanced stem generation. The RegStemma procedures use only one character of the possible case endings as enhancement, whereas FullRegStemmas use whole case endings. Plus signs and asterisks denote to different types of matching, strict or relaxed, according to the semantics of regular expression notation. Use of the restrictive operator (+ in the name) makes the matching more limited and the non-restrictive operator (* in the name) more relaxed. FullRegStemma+ and FullRegStemma* are the respective fully enhanced versions, where restrictiveness of the match is due to the word final operator $ (or its absence) (see Study IV for further details or Friedl 1997, 18).

5.3. Summary of the FCG method studies

5.3.1. Research problems

After Studies I and II we were confident that generative methods of keyword variation management worked for Finnish best-match retrieval. In Study II we had already tried full-form generation of keywords with all the case forms (26–28 distinct forms). However, believing that not all the case forms would be needed, we sought for an intermediate solution where only a subset of the full forms would be generated. From this and statistical corpus analyses we developed our Frequent Case Generation method (FCG).

Two empirical studies on FCG method were carried out. Study III first introduced the method and showed that it worked for Finnish, which in principle has a plethora of distinct grammatical word forms. The research problems of the Study III were as follows:

1) What kind of distributions do Finnish nominal case forms have? Are all circa 2000 grammatical noun forms equally probable in texts? If the distributions are skewed, what are the most frequent case forms and are there clearly distinctive case forms that would be suitable as keyword forms?
2) Does frequent case form generation of keywords work in the IR of a morphologically complex language?
3) If it works, what is the best balance between the number of generated keyword form variants and achieved mean average precision in retrieval?

Study IV evaluated the FCG method using three Indo-European languages, Swedish, German and Russian that are morphologically complex enough to be of interest. The research problems of the second FCG study were as follows:

1) Is the FCG approach viable across languages of varying morphological complexity?

   1a) In order of increasing complexity, what is the performance of FCG in Swedish, German, Russian and Finnish as observed in generally available test collections?

   1b) How many morphological surface forms are needed to achieve reasonable performance?

   1c) How does this performance compare to doing nothing at all, stemming and lemmatization (reductive morphological methods)?

2) What is the effect of topic length on the performance of FCG as compared to doing nothing at all, stemming or lemmatization?

5.3.2. Methods

Our main purpose in Study III was to show that generating only a fraction of the grammatical nominal forms for Finnish keywords produces IR results that are comparable to other methods, especially lemmatization.

The forms to be used in retrieval were first analysed from several independent text corpora of varying sizes. Corpus analysis showed that only six cases (out of 14) constituted about 84–88 % of the token level occurrences of case forms for nouns – thus covering 84–88 % of the possible variation of about 2000 distinct inflected forms of nouns. Based on this finding, four different simulated frequent case form generation procedures (FCGs) were evaluated in two different full-text collections, TUTK and CLEF 2003.

This time TUTK's relevance scale was utilised in the following manner: we used three relevance scales, liberal, normal and stringent. In the liberal relevance class all TUTK's relevant documents from levels 1–3 were included. In the normal relevance class only relevant and highly relevant documents (levels 2–3) were included. In the stringent relevance class only highly relevant documents (level 3) were included. The CLEF 2003 collection uses a binary relevance scale. In both collections we ran long queries made straight out from the title and description fields of the topics.

In Study IV we applied our FCG method to three new languages, Swedish, German and Russian, which are morphologically fairly complex and thus suitable for further evaluation of the FCG method. We also evaluated the behaviour of Finnish very short queries. For Swedish we analysed the most frequent forms to be used as keywords on the basis of a SWETWOL analysis of newspaper material. For German word form frequency analysis we used an existing Tiger corpus. For Russian we obtained the case distribution information from the Russian national corpus.

5.3.3. Results

The results of Study III showed that frequent case form generation works in full-text retrieval in a best-match query system and at best competes well with the gold standard, lemmatization, for Finnish. Our best FCG procedures, FCG_9 and FCG_12 – with the number of variant keyword forms shown in the name of the procedure - achieved about 86 % of the best average precision of FINTWOL in TUTK and about 90 % in CLEF 2003. The runtimes of the FCG queries were also shown to be comparable to those of the other methods.

Results from the TUTK collection of Study III are shown in Table 7.

Table 7. Results of test runs in the TUTK collection on three relevance scales.

Mean average precision (per cent) – interpolated			
Method	**Liberal relevance**	**Normal relevance**	**Stringent relevance**
FINTWOL – lemmatized index, compounds split in the index	37.8	35.0	24.1
FCG_12, inflected index	32.7 (-5.1)	30.0 (-5.0)	21.4 (-2.7)
FCG _9, inflected index	32.4 (-5.4)	29.6 (-5.4)	21.3 (-2.8)
FCG _6, inflected index	30.9 (-6.9)	28.0 (-7.0)	21.0 (-3.1)
Snowball, stemmed index	29.8 (-8.0)	27.7 (-7.3)	20.0 (-4.1)
FCG _3, inflected index	26.4 (-11.4)	23.9 (-11.1)	18.9 (-5.2)
Plain, inflected index	19.6 (-18.2)	18.9 (-16.1)	12.4 (-11.7)

In Study IV we continued to evaluate the FCG method. We showed first that very short Finnish queries are also suitable for FCG style retrieval. After that we analysed Swedish, German and Russian long and short queries in an FCG setting. For Swedish and German we had both a lemmatizer and a stemmer available for comparison. For Russian we had only access to a Russian stemmer.

Our Swedish results showed quite clearly that the FCG method works well for Swedish in both long and short queries. In short queries the differences between all methods were smallest, but the margin between plain keywords and the best method also increased, which emphasises the importance of some sort of keyword processing. Lemmatization with compound splitting was the best method in both long and short queries. However, the best Sv-FGC methods achieved about 91 and 94 per cent of the best lemmatization results with long and short queries respectively.

Our German results showed that the method works for German as well, although the overlap of inflected noun forms[14] slightly disturbed the results. The margin between plain keywords and the best method was smaller than in Swedish, which is probably due to inflectional homography. However, the differences of FCGs from the gold standards were statistically insignificant.

Our Russian results were partly counterintuitive. With both long and very short queries recall rose steadily in number of retrieved documents when more case forms were put into the query. However, the mean precision of long queries did not get any better when forms were added, but rather decreased. The best mean average precision with long queries was gained with the process Ru-FCG_3 generating the nominal singular forms only. But the inflected queries, where query words were taken as such from the topics, were the third best method in terms of mean average precision with long queries. Overall it seemed that short Russian queries showed some advantage for FCGs, but as the collection is small and has very few relevant documents, the interpretation of the Russian results remained inconclusive.

In very short Russian queries the difference between doing nothing (inflected queries) and the use of a different number of case forms was quite clear. The differences between different case form procedures in terms of mean average precision were very small, and only recall clearly improved when more forms were used.

[14] *By overlap of inflected forms we mean homography, where distinct case forms can only be identified from the preceding article, as for example* Männer, *which can be plural nominative, accusative and genitive.*

6. Discussion and conclusions

The main objective of the present study was to evaluate the suitability of reductive and generative methods to IR of highly inflected languages in a laboratory IR setting with best-match retrieval system. The main contributions of this study can be summed up as follows:

- Our main contribution was to show that generative methods are also appropriate for IR in morphologically complex languages in a best-match retrieval environment. For Finnish we evaluated inflectional stem generation and its enhancements. We also created a new method, FCG, for inflectionally at least moderately complex languages and evaluated the method with four languages. For three of the languages (Finnish, Swedish and German) the method was shown to yield good retrieval results when lemmatization was used as a point of reference. For Russian the results were inconclusive and the method should be re-evaluated with a better Russian collection. As the method itself is based on skewness of word form distributions, it is also expected to be applicable to other morphologically complex languages.
- For Finnish best-match IR we have shown that besides lemmatization, also stemming, inflectional stem generation and its enhancements and most frequent case form generation of keywords give good retrieval results when compared to the state-of-the-art, lemmatization. This broadens the spectrum of possible morphological tools for the handling of morphological variation of Finnish, which has been considered challenging in IR. As Finnish can be seen as a "worst case" language with respect to morphological variation, our results should also show the way to other languages having a fair degree of morphological variation.
- Most of the methods evaluated in the study are shown to work for both long laboratory type (unnatural) queries and more realistic very short queries, which resemble natural user queries at least in the number of the keywords, although the research setting was a typical laboratory environment.

When we began this study, there were not very many large scale best-match retrieval studies on the effects of different morphological methods available for Finnish.[15] Kunttu's thesis (2003) was one of the first studies on different morphological methods for Finnish IR in a best-match retrieval environment. Kunttu's work was mostly fashioned along the guidelines given by the work of

[15] *Kekäläinen (1999) and Sormunen (2000) are prominent examples of earlier best-match retrieval studies on Finnish. Their focus, however, was not on morphological tools for keyword variation management.*

Alkula (2000), which was done in a Boolean retrieval system. After or at the same time as Kunttu, the first two studies of this thesis and the studies by Airio (2006), Mayfield and McNamee (2003), Hollink et al. (2004) and Tomlinson (2002, 2003) showed, among other things, that even a simple and linguistically naive stemmer works well for IR in Finnish. This had not been evaluated earlier, and it had been thought that simple stemming would not be suitable for Finnish IR (Koskenniemi 1983, 13). Our studies have also shown that the coverage of keyword variation of highly inflected languages may be achieved without index lemmatization or stemming, at least in a typical IR laboratory environment of medium scale.

We started this study by applying an inflectional stem generator of our own for inflected Finnish indexes. It had been shown earlier (Koskenniemi 1985a, Alkula 2000) that inflectional stem generation works well for Finnish retrieval in a Boolean retrieval environment. Kunttu (2003) replicated the work of Alkula in a best-match environment with output of Finstems (Koskenniemi 1985a), and thus it was reasonable to evaluate inflectional stem generation further. In Study I, however, we found that even if inflectional stem generation yields in good retrieval performance when compared to lemmatization, the queries tend to become unmanageably long when the index is harvested with all the stems for possible matches. This effect was later restricted in Study II by enhancing the stems with regular expressions that gave a part or the whole of the inflectional endings after stems. This resulted in more manageable query lengths, but it also made enhanced stem queries slightly less effective. Nevertheless, the results of Study II showed that this kind of approach is a feasible way to do inflectional stem-based retrieval for Finnish.

After these studies, we took a fresh start, and introduced a new method, the generation of only the most frequent case forms of keywords, to manage keyword variation of Finnish. In linguistics it is well known that although the number of grammatical forms for the words of a language may in principle be huge, realizations of the forms are far scarcer and the distributions of the different word forms in corpuses are skewed (Karlsson 1986, 2000; Kostić, Marković & Baucal 2003; for statistical language analysis cf. Baayen 2001). This simple idea was then utilised in a new method, Frequent Case Generation (FCG). The method was first established and evaluated with Finnish and after promising results it was tried out with Swedish, German and Russian.

As our empirical results in Studies III and IV show, the FCG method is a promising alternative to account for morphological keyword variation in languages with at least moderately complex morphology. In our tests Finnish represents a language that shows very rich morphological variation in number of grammatical word forms. Swedish, German and Russian are more typical examples of languages that have enough morphological variation to deserve attention in IR.

It may be anticipated that a major application area for the FCG approach is web searching. The present state of language specific search capabilities of general search engines, such as Google, Alltheweb or Altavista, does not seem satisfying from the user point of view. Very few search engines seem to offer e.g. stemming, and search term truncation has been omitted almost totally (Search Engine Showdown 2006). The status of language specific search capabilities of general search engines thus seems poor. Bar-Ilan and Gutman (2005) report their findings for four different languages (French, Hebrew, Hungarian and Russian, tests made in November 2002) with national and general search engines. From their results it can be seen that national web services (such as Yandex in Russian, Origo-Vizsla in Hungarian and Morfix for Hebrew) take into account the requirements of each particular language and their search results are far better than those of general search engines with the language in question. As the web is constantly becoming more multilingual (Bar-Ilan and Gutman 2005; Greffenstette and Nioche 2000), it would also be desirable, if the most popular search tools of the web were more sensitive to the language specific requirements. Otherwise the huge information potential of the non-English web cannot be effectively utilised.

The method we have presented in Studies III and IV provides one effective solution to the problem of web searches in various languages. So far we have shown that it competes well with other morphological programs in languages of varying morphological complexity. The basic idea of the method is easily adaptable to other languages and evaluation of the effects of FCG style of search can be implemented relatively easily with the present state of language technology tools and search engines.

Why, then, use this kind of approach when full morphological analysis programs are available? There are several reasons for this.

Firstly, the generation approach works with inflected indexes of search systems and no base form processing is needed for the index. This is mainly of practical value, but an important issue: as web indexes especially are very large, separate base form runs for them would take a great deal of time. As indexes also need constant updating, making base form indexes does not sound like a very good option. Searching the index with a few most frequent inflected forms of keywords should not take too much time, when the usual web search consists of only one to three keywords regardless of the language (Jansen, Spink & Saracevic 2000; Jansen & Spink 2005).

Secondly, our approach, generation of only the most frequent word forms, is simple and could be easier to implement, if (usually) commercial morphological analyzers are not available for IR in a specific language.

Thirdly, while word based IR is quite effective, it requires management of word form variation in some way. Although full-scale morphological programs perform well, as Galvez et al. (2005) and Galvez and de Moya-Anegón (2006) argue, they may be unnecessarily complex and resource consuming for simple keyword variation management purposes.

Fourthly, the generation approach is not as dependent on large lexicons as are full-scale morphological analyzers, because for many languages use of the lexicon in generation is not necessary. The main advantage of not using large lexicons is that out-of-vocabulary words do not adversely affect retrieval results, as they evidently do with lemmatizers.

For the languages so far evaluated with the FCG approach, realistically long web-style searches would mean longer searches than with one form. In Table 8 we present the mean number of word forms per lexeme that are maximally generated for our short queries for each language's best FCG procedure in Study IV. From this the number of required search forms can be realistically approximated.

Table 8. Mean number of generated word forms per lexeme in short queries, stop words not included

Language	Forms/lexeme
Finnish	12.27 (FCG_12)
	9.35 (FCG_9)
German	2.98 (De-FCG_4)
Russian	5.34 (Ru-FCG_8)
	3.80 (Ru-FCG_6)
Swedish	3.29 (Sv-FCG_4)

As seen in the table, the figures for German and Swedish are not prohibitively high. In a typical one to three word web search, these figures would mean about 3–10 keyword forms for German and Swedish. For Russian searches the number of generations for Ru-FCG_8 would mean already about 5–16 keyword forms, which is rather high. Ru-FCG_6 would generate 4–12 keyword forms. For Finnish the number of keyword forms, 9–36, might border on the impractical, but good index packaging and retrieval algorithms might make even this possible. A smaller number of generated keyword forms, six, (cf. Study III) could also be enough for Finnish.

The findings of this study show that on the general level both reductive and generative management methods of morphological variation give at least satisfactory IR results. Differences between the gold standards and the newly developed methods are often small and not always statistically significant. Further practical conclusions would need user oriented evaluation in real web search environments with large collections.

7. Future work

In this thesis we have been discussing the usage of reductive and generative morphological methods in handling of keyword variation in IR. We have shown empirical results, which establish that both kinds of methods are effective with languages that are morphologically at least somehow complex. It would also be interesting to see, what are the reasons for performance differences of these methods. This would need further studying of the search results and comparison of the found index words with different methods.

The collections we used in the present thesis were established test collections, TUTK and CLEF for Finnish, and the CLEF collections for Swedish, German and Russian. They are based on relatively small corpora, when compared to many experimental and operational corpora being used currently in IR research. The collections are derived from newspaper archives, and on average, the documents are not particularly long. This complicates the assessment of the generalizability of the findings, and further studies using, e.g., larger web and other collections are needed. It should, however, be noted, that the availability of very large collections in other languages than English, is still not self-evident. Thus results achieved with well-known standard test collections with smaller languages are of value when developing search systems for those languages.

As Swedish, German and Russian are only modestly inflected languages, it would have been possible to use all of the inflected noun and adjective forms as keywords to get a comparative baseline. This was not done, but such comparisons can be made in future work. We believe that this type of full generation would be most suitable for less inflected languages, such as English or Romance languages. With modestly inflected languages that have lots of documents available, such an approach might result in efficiency problems in a real large scale multi-user retrieval system and thus our FCG style with an optimized number of keywords for each language might be a better compromise of effectiveness and efficiency.

8. References

Abu-Salem, H., Al-Omari, M. & Evens, M. W. 1999. Stemming methodologies over individual query words for an Arabic information retrieval system. *Journal of the American Society for Information Science* 50(6), 524−529.

Ahlgren, P. 2004. *The effects of indexing strategy-query term combination on retrieval effectiveness in a Swedish full text database*. Department of Library and Information Science/Swedish School of Library and Information Science. University College of Borås/Göteborg University.

Airio, E. 2006. Word normalization and decompounding in mono- and cross-lingual IR. *Information Retrieval 9,* 249–271.

Alegria, I., Aranbaze M., Ezeiza, A. & Urizar. R. 2002. Robustness and customisation in an analyzer/lemmatiser for Basque. Available at <http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1019570039/publikoak/robust2.pdf>. Accessed 15.8.2005.

Alkula, R. 2000. *Merkkijonoista suomen kielen sanoiksi.* Acta Universitatis Tamperensis 763, Available at <http://acta.uta.fi/pdf/951-44-4886-3.pdf>. Accessed 15.5.2005.

Alkula, R. 2001. From plain character strings to meaningful words: producing better full text databases for inflectional and compounding languages with morphological analysis software. *Information Retrieval* 4, 195−208.

Allan, J., Callan, J., Croft, W. B., Ballesteros, L., Byrd, D. Swan, R. & Xu 1997. J. Inquery does battle with TREC-6. Available at < trec.nist.gov/pubs/trec6/papers/umass-trec6.ps.gz >. Accessed 1st March, 2007.

Anderson, J. D. & Pérez-Carballo, J. 2001. The nature of indexing: how humans and machines analyze messages and texts for retrieval. Part II: Machine indexing, and the allocation of human versus machine effort. *Information Processing and Management* 37, 255−277.

Arampatzis, A., van der Weide, Th., van Bommel, P. & Koster, C. H. A. 2000. Linguistically motivated information retrieval. In Kent, A. & Hall, K. M. (eds.*), Encyclopedia of Library and Information Science,* vol. 69. New York, Basel: Marcel Dekker, Inc., 201−222.

Arppe, A. 1996. Information explosion and the use of linguistic tools in Finland. *Kieli ja Tietokone, AFinLAn Vuosikirja 1996.* Suomen Soveltavan Kielitieteen Yhdistyksen Julkaisuja, 54 (= AFinLA Series, 54), 7−32.

Baayen, R. H. 2001. *Word Frequency Distributions*. Dordrecht Boston London: Kluwer Academic Publishers.

Bacchin, M., Ferro, N. & Melucci, M. 2004. A probabilistic model for stemmer generation. *Information Processing and Management* 41(1), 121 – 137.

Bar-Ilan, J. & Gutman, T. 2005. How do search engines respond to some non-English queries? *Journal of Information Science* 31, 13–28.

Belew, R. K. 2000. *Finding out about A Cognitive Perspective on Search Engine Technology and WWW.* Cambridge: Cambridge University Press.

Belkin, N. J. & Croft, W. B. 1987. Retrieval techniques. In Williams, M. E. (ed.), *Annual Review of Information Science and Technology,* vol. 22. New York, NY: Elsevier, 109–145.

Braschler, M. & Ripplinger, B. 2004. How Effective is Stemming and Decompounding for German Text Retrieval? *Information Retrieval* 7, 291–316.

Broglio, J., Callan, J., Croft, B. & Nachbar, D. 1995. Document Retrieval and Routing Using the INQUERY System. In *Proceedings of the Third Text Retrieval Conference (TREC-3*), Gaithesburg, MD: National Institute of Standards and Technology, special publication 500-225, pp. 29−38.

Buckley, C. & Voorhees, E. M. 2000. Evaluating evaluation measure stability. In Belkin, N. J., Ingwersen P. & Leong, M. (eds.) *Proceedings of the 23rd AnnualInternational ACM SIGIR Conference on Research and Development in Information Retrieval,* 33−40.

Buckley, C. & Voorhees, E. M. 2004. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '04,* ACM Press, 25−32.

Callan, J., Croft, B. & Harding, S. 1992. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Databases and Expert Systems Applications,* Berlin: Springer Verlag, 78−84.

De Campos, L. M., Fernández-Luna, J. M. & Huete, J. F. 2004. Bayesian Networks and information retrieval: an introduction to the special issue. *Information Processing & Management* 40 (5), 727−733.

Chowdhury, G. G. 2003. Natural Language Processing. In Cronin, B. (ed.) *Annual Review of Information Science and Technology,* vol. 37. New York, NY: Elsevier, 51−89.

Comrie, B. (ed.) 1990. *The World’s Major Languages*. New York, Oxford: Oxford University Press.

Conover, W. J. 1980. *Practical Nonparametric Statistics.* 2nd edition. New York: John Wiley and Sons.

Cosijn, E. & Ingwersen, P. 2000. Dimensions of relevance. *Information Processing and Management* 36, 533−550.

Crestani, F., Lalmas, M., van Rijsbergen, C. J. & Campbell, I. 1998. “Is This Document relevant... Probably”: A Survey of Probabilistic Models in Information Retrieval. *ACM Computing Surveys* 30 (4), 528−552.

Creutz, M. 2005. E-mailed information, 17. 5. 2005. Personal correspondence.

Creutz, M. & Linden, K. 2004. *Morpheme Segmentation Gold Standards for Finnish and English*. Publications in Computer and Information Science. Report A77. Espoo: Helsinki University of Technology.

Daille, B., Fabre, C. & Sébillot, P. 2002. Applications of Computational Morphology. In Boucher, P. & Plénat, M. (eds.), *Many Morphologies.* Somerville, MA: Cascadilla Press, 210−234.

Frakes, W. B.1992. Stemming algorithms. In Frakes, W. B. & Baeza-Yates, R. (eds.), *Information Retrieval. Data Structures and Algorithms.* Upper Saddle River, NJ, USA: Prentice Hall, 131−160.

Friedl, J. E. F. 1997. *Mastering Regular Expressions.* Sebastobol, CA: O'Reilly & Associates, Inc.

Galvez, C., de Moya-Anegón, F. & Solana, V. H. 2005. Term conflation methods in information retrieval. Non-linguistic and linguistic approaches. *Journal of Documentation* 61(4), 520–547.

Galvez, C. & de Moya-Anegón, F. 2006. An evaluation of conflation accuracy using finite-state transducers. *Journal of Documentation* 62, 328–349.

Grefenstette, G. & Nioche, J. 2000. Estimation of English and non-English language Use on the Web. Available at <http://arxiv.org/ftp/cs/papers/0006/0006032.pdf>. Accessed 28th October, 2006.

Grossman, D. A. & Frieder, O. 2004. *Information Retrieval.* Algorithms and Heuristics. Second edition. Netherlands: Springer.

Harman, D. 1991. How effective is suffixing? *Journal of the American Society for Information Science* 42(1), 7–15.

Hedlund, T. 2003. *Dictionary-Based Cross-Language Information Retrieval.* Acta Universitatis Tamperensis 962.

Hellberg, S. 1972. Computerized lemmatization without the use of a dictionary: a case study from Swedish lexicology. *Computers and the Humanities* 6, 209–212.

Hollink, V., Kamps, J., Monz, C. & de Rijke, M. 2004. Monolingual Document Retrieval for European Languages. *Information Retrieval* 7, 33–52.

Holman, E. 1988. Finnmorf: A Computerized Research Tool for Students of Finnish Morphology. *Computers and the Humanities* 22, 165–172.

Hull, D. 1993. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval.* New York: ACM, 329–338.

Hull, D. 1996. Stemming algorithms: a case study for detailed evaluation. *Journal of the American Society for Information Science* 47(1), 70–84.

Ingwersen P. & Järvelin, K. 2005. *The Turn. Integration of Information Seeking and Retrieval in Context.* Dordrecht, The Netherlands: Springer.

Jacquemin, C. & Tzoukerman, E. 1999. NLP for term variant extraction: synergy between morphology, lexicon, and syntax. In T. Strzalkowski (ed.), *Natural Language Information Retrieval.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 25–74.

Jansen B. & Spink, A. 2005. An analysis of Web searching by European Alltheweb.com users. *Information Processing and Management* 41, 361−381.

Jansen, B., Spink, A. & Sarasevic, T. 2000. Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web. *Information Processing & Management* 36, 207–227.

Jäppinen, H. & Ylilammi, M. 1986. Associative Model of Morphological Analysis: an Empirical Inquiry. *Computational Linguistics* 12 (4), 257−272

Järvelin, K. 1995. *Tekstitiedonhaku tietokannoista* [Text Retrieval in Databases].Espoo, Finland: Suomen ATK-kustannus.

Kalamboukis, T. Z. 1995. Suffix stripping with modern Greek. *Program* 29(3), 313–321.

Karlsson, F. 1987. *A Finnish grammar*. Porvoo: WSOY.

Karlsson, F. 1986. Frequency Considerations in Morphology. *Zeitsschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung* 39, 19–28.

Karlsson, F. 2000. Defectivity. In Booij G. et al. (eds.), *Morphology. An International Handbook on Inflection and Word-Formation.* Volume 1. Berlin: Walter de Gruyter, 647–654.

Kekäläinen, J. 1999. *The effects of query complexity, expansion and structure on retrieval performance in probabilistic text retrieval*. Acta Universitatis Tamperensis 678.

Kettunen, K. 1991. Doing the stem generation with Stemma. In Niemi, J. (ed.) *Papers from the Eighteenth Finnish Conference of Linguistics*. Kielitieteellisiä tutkimuksia, Joensuun yliopisto, N:o 24, 80–97.

Kettunen, K. 2006. Developing an automatic linguistic truncation operator for best-match retrieval in inflected word form text database indexes. *Journal of Information Science* 32(5), 465–479.

Kettunen, K. & Airio, E. 2006. Is a morphologically complex language really that complex in full-text retrieval? In T. Salakoski et al. (eds.), *Advances in Natural Language Processing*, LNAI 4139. Berlin Heidelberg: Springer-Verlag, 411–422.

Kettunen, K., Airio, E. & Järvelin K. 2007. Restricted Inflectional Form Generation in Management of Morphological Keyword Variation. Accepted for publication in *Information Retrieval*.

Kettunen, K., Kunttu, T. & Järvelin, K. 2005. To stem or lemmatize a highly inflectional language in probabilistic IR environment? *Journal of Documentation*: 61 (4), 476–496.

Korfhage, R. R. 1997. *Information Storage and Retrieval*. New York: John Wiley & Sons, Inc.

Korhonen, M. 1994. *Kielen synty* [Origins of Language]. Toinen painos. Juva: WSOY.

Koskenniemi, K. 1983. *Two-level morphology: a general computational model for word-form recognition and production*. Publications of the Department of General Linguistics, University of Helsinki. No. 11.

Koskenniemi, K. 1985a. FINSTEMS: a module for information retrieval. In Karlsson, F. (ed.), *Computational morphosyntax*. Report on research

1981–84. Publications of the Department of General Linguistics, University of Helsinki. No. 13, 81–92.

Koskenniemi, K. 1985b. A System for Generating Finnish Inflected Word Forms. In Karlsson, F. (ed.), *Computational Morphosyntax*. Report on research 1981–84. Publications of the Department of General Linguistics, University of Helsinki. No. 13, 63–80.

Koskenniemi, K. 1985c. An application of the two-level model to Finnish. In Karlsson, F. (ed.), *Computational morphosyntax*: Report on research 1981–84. Publications 13, University of Helsinki, Department of General Linguistics, Helsinki, 1985, 19–41.

Koskenniemi, K. 1996. Finite state morphology and information retrieval. *Natural Language Engineering* 2 (4), 331−336.

Kostić, A., Marković, T. & Baucal, A. 2003. Inflectional Morphology and Word Meaning: Orthogonal or Co-implicative Cognitive Domains. In Baayen, R. H. & Schreuder, R. (eds*.) Morphological Structure in Language Processing*. Trends in Linguistics, Studies and Monographs 151. Berlin: Mouton de Gruyter, 1–43.

Kraaij W. 2004. *Variations on Language Modeling for Information Retrieval.* Haag: CTIT Ph. D. series No. 04-62. Electronic version available at <http://dis.tpd.tno.nl/mmt/pubs/wkthesis-c.pdf >.

Krovetz, R. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Pittsburg, PA, 191–202.

Krovetz, R. 2000. Viewing morphology as an inference process. *Artificial intelligence* 118, 277−294.

Krovetz, R. & Croft, W. B. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems* 10(2), 115–141.

Kunttu, T. 2003. *Perus- ja taivutusmuotohakemiston tuloksellisuus todennäköisyyksiin perustuvassa tiedonhakujärjestelmässä.* Informaatiotutkimuksen pro gradu -tutkielma. Informaatiotutkimuksen laitos, Tampereen yliopisto. (M.Sc. Thesis, University of Tampere, Dept. of Information Studies).

Lassila, E. 1988. Suomen kielen sanamuodot taivuttava ohjelma FORMO. In Mäkelä, M. Linnainmaa, S. & Ukkonen, E. (eds.) *STeP-88. Invited Papers. Contributed Papers: Applications*. Helsinki: Finnish Artificial Intelligence Society, 118–126

Losee, R. M. 1998. *Text Retrieval and Filtering. Analytic Models of Performance.* Boston/Dordrecht/London: Kluwer Academic Publishers.

Lovins, J. B. 1968. Development of a Stemming Algorithm. *Mechanical Translation and Computation Linguistics* 11 (1), 23–31.

Matthews, P. H. 1991. *Morphology*. Second edition. Cambridge: Cambridge University Press.

Mayfield, J. & McNamee, P. 2003. Single N-gram Stemming. In *Proceedings of Sigir2003, The Twenty-Sixth Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval, 415−416.

McNamee, P. & Mayfield, J. 2004. Character n-gram tokenization for European language text retrieval. *Information Retrieval 7*(1−2), 73−97.

Meadow, C. T., Boyce, B. R. & Kraft, D. H. 2000. *Text Information Retrieval Systems*. Second edition. San Diego: Academic Press.

Metzler, D. & Croft, W. B. 2004. Combining the Language Model and Inference Network Approaches to Retrieval. *Information Processing and Management* Special Issue on Bayesian Networks and Information Retrieval 40, 735–750.

Mitkov, R. 2004. *The Oxford Handbook of Computational Linguistics*. Paperback edition. Oxford: Oxford University Press.

Mizzaro, S. 1997. Relevance: the whole history. *Journal of the American Society for Information Science*, 48(9), 810−832.

Nurminen, R. 1986. *Suomen kieltä analysoivien ohjelmien vaikutus dokumenttien tallennukseen ja hakuun suorakäyttöjärjestelmissä*. Tampereen yliopisto, kirjastotieteen ja informatiikan laitos, pro gradu. (Master’s thesis, University of Tampere, Dept. of Information Studies).

Oflazer, K. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics* 22 (1), 73−89.

Pirkola, A. 2001. Morphological typology of languages for IR. *Journal of Documentation* 57(3), 330−348.

Pirkola, A. & Järvelin, K. 2001. Employing the resolution power of search keys. *Journal of the American Society for Information Science and Technology,* 52 (7), 575−583.

Popovic, M. & Willett. P. 1992. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science* 43(5), 384−390.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14, 130−137.

Porter, M. 2001. Snowball: A language for stemming algorithms. Available at <http://snowball.tartarus.org/texts/introduction.html>. Accessed January 15th, 2004.

Robertson, S. 1981. The methodology of information retrieval experiment. In Sparck-Jones, K. (ed.), *Information Retrieval Experiment*. London: Butterworth, 9−31.

Robertson, S. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation* 60 (5), 503−520.

Robertson, A. M. & Willett, P. 1993. A Comparison of Spelling-correction Methods for the Identification of Word Forms in Historical Text Databases. *Literary and Linguistic Computing* 8(3), 143–152.

Salton, G. & McGill, J. M. 1983. *Introduction to modern information retrieval*. New York, NY: McGraw-Hill.

Sanderson, M. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval, Dublin, Ireland, 142−151.

Saracevic, T. 1975. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science* 26(6), 321−343.

Schinke, R., Greengrass, M., Robertson, A. M. & Willet, P. 1996. A Stemming algorithm for Latin text databases. *Journal of Documentation* 52(2), 172−187.

Search Engine Showdown. Search Engine Features Chart (Last updated Sep. 17, 2006). Available from <http://www.searchengineshowdown.com/features/>. Accessed October 28th, 2006.

Sever, H. & Bitirim, Y. 2003. FindStem: Analysis and Evaluation of a Turkish Stemming Algorithm. In Nascimento, M., Moura, E. & Oliveira, A. (eds.) *String Processing and Information Retrieval*, 10th International Symposium, SPIRE 2003, 238−251.

Siegel, S. & Castellan, N. J. Jr. 1988. *Nonparametric statistics for the behavioral sciences*. Second edition. New York: McGraw-Hill Book Company.

Snowball web site. <http://snowball.tartarus.org/ >. Accessed January 22nd, 2007.

Sormunen, E. 2000. *A method for measuring wide range performance of Boolean queries in full-text databases.* Acta Universitatis Tamperensis 748. Tampere: University of Tampere.

Tague-Sutcliffe, J. M. & Blustein, J. 1995. A Statistical Analysis of the TREC-3 Data. In Harman, D. (ed.), *The Third Text Retrieval Conference (TREC-3)*. NIST Special Publication 500-226, 385–398.

Tomlinson, S. 2002. Experiments in 8 European Languages with Hummingbird SearchServer™ at CLEF 2002. Available at <http://clef.iei.pi.cnr.it:2002/workshop2002/WN/26.pdf>. Accessed April 28, 2004.

Tomlinson, S. 2003. Lexical and Algorithmic Stemming Compared for 9 European Languages with Hummingbird SearchServer™ at CLEF 2003. Availabe at <http://clef.iei.pi.cnr.it/2003/WN_web/19.pdf >. Accessed April 28th, 2004.

Tordai, A. & de Rijke, M. 2005 Hungarian Monolingual Retrieval at CLEF 2005. Working Notes for the CLEF 2005 Workshop, 21-23 September, Vienna, Austria. <http://www.clef-campaign.org/2005/working_notes/workingnotes2005/tordai05.pdf>. Accessed September 6th, 2006.

Trost, H. 2004. Morphology. In Mitkov, R. (ed.) *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press, 25–48.

Voorhees, E. M. 2004a. Overview of TREC 2004. Available at <http://trec.nist.gov/pubs/trec13/papers/OVERVIEW13.pdf >. Accessed January 15th, 2007.

Voorhees, E. M. 2004b. Common Evaluation Measures. Available at <http://trec.nist.gov/pubs/trec13/appendices/CE.MEASURES.pdf>. Accessed January 15th, 2007.

Erratum for contributed articles

[I] In Table IV (p. 482 in the original publication) we state that compounds in the FINTWOLled index of TUTK are not split. They are split in the index.