Information Retrieval, 4, 195–208, 2001

From Plain Character Strings to Meaningful Words: Producing Better Full Text Databases for Inflectional and Compounding Languages with Morphological Analysis Software

RIITTA ALKULA
Tieto Enator Corporation, Finland

Received January 16, 2001; Revised May 7, 2001; Accepted May 24, 2001

Abstract. The paper deals with linguistic processing and retrieval techniques in fulltext databases. Special attention is focused on the characteristics of highly inflectional languages, and how morphological structure of a language should be taken into account, when designing and developing information retrieval systems. Finnish is used as an example of a language, which has a more complicated inflectional structure than the English language. In the FULLTEXT project, natural language analysis modules for Finnish were incorporated into the commercial BASIS information retrieval system, which is based on inverted files and Boolean searching. Several test databases were produced, each using one or two Finnish morphological analysis programs.

Keywords: natural language processing, full text retrieval, stemming, morphology

Introduction

As information storage and retrieval systems, especially full text retrieval systems deal with natural language texts, it makes sense to treat the text as a natural language phenomenon instead of plain character strings. Unfortunately, this is not the case in most information retrieval systems.

The problem is that although information retrieval is strongly based on textual information and thus dependent on the use of the written form of natural language, it has not been easy to utilize methods and techniques developed in natural language processing. Specific attempts to apply computational linguistics and/or to test its effectiveness in information retrieval have been the exception rather than the rule. One reason for this lack of interaction has been that natural language processing and information retrieval have concentrated on different levels of language analysis. In the 1990s, however, the two domains have been growing closer each other.

To be precise, it is not the structure of the natural language that causes problems in information retrieval. The actual problem is the loss of this structure, when a database is constructed and inverted file(s) compiled. The words are reduced to character strings, the syntax between words is omitted and must be replaced by other means while searching. This primitive view of a word is shared also by most electronic archives, electronic mail systems, etc. By adding linguistic knowledge to these systems, the accuracy of existing

text (word) processing facilities can be improved and new processing facilities can be added.

This paper presents the findings of the FULLTEXT project, where the Finnish language was used as an example of a language that strongly differs from English. For that reason, Finnish information retrieval systems cannot flawlessly utilize techniques developed for the English language texts.

In the project, different morphological analysis program were incorporated to a fulltext database and retrieval experiments were run to test, if morphological software helps to improve the performance of the system. Thus the research was carried out within the traditional laboratory environment paradigm, presenting various test parameters and their evaluation (Alkula 2000).

There have been some earlier studies (e.g. Karlsson 1985) in which potential methods for using morphological analysis software in Finnish information retrieval systems have been presented. In those previous studies, however, no empirical retrieval experiments were performed.

Definitions

In traditional information retrieval systems, the inverted file (index) consists of unmanipulated words, and all of the inflectional variants of a word are treated as separate entities (items, tokens). Each word form is an individual character string and, consequently, a separate index entry (for example, dog and dogs are separate entries).

Because of inflection, a searcher must take care to find all forms of a word. As the typing of every inflected form of a word would be awkward, information retrieval systems offer a so-called *truncation* facility. When the symbol "*" indicates truncation, the search term *dog** retrieves all documents containing any word that begins with this character string, for example dog, dogs—and, unfortunately, also so-called false drops that accidentally begin with the same character string, e.g. dogmatics.

Basically, truncation means the same as *stemming*, namely cutting off affixes in order to bring out the word stem. It is, however, a more robust operation because, for information retrieval purposes, a shorter form than the root stem common to the all inflectional variants may be quite adequate (e.g. democracy → *democ**) In information retrieval research, stemming is mostly used in the restricted sense of suffix removal.

Word *normalization* is a computational process which identifies word variants and reduces them to a single canonical form, i.e. *a citation form* or a *basic word form.* For example, from forms dog, dogs and dog's is generated the basic form dog. Normalization is usually based on dictionary lookup; in other words, the word forms generated by the process are compared to the citation forms in a machine-readable dictionary (lexicon).

Lemmatization is a term mostly used in computational linguistics. It is a process of defining or solving, which word variants belong under a common *lemma*. Lemma defines a group or a cluster and does not necessarily take a stand of the basic word form, as normalization does (Karlsson 1994).

Conflation is a general term for all processes of merging together nonidentical words which refer to the same principal concept. In the paper, conflation refers to morphological

conflation and morphological query expansion, where the related word forms resemble each other as character strings. Other methods, such as expanding query with synonyms, are not discussed here, as they deal with conceptual similarity, whereas we are interested in morphological similarity.

Morphological analysis software is a general term for all computational linguistics software that are used to analyze linguistic expressions (typically words) on the morphological level. They either conflate variants together under a common lemma or basic form (lemmatizers, basic word form generators), or from a specific item produce its variants (stem generators, inflected word-form generators), or otherwise process words (hyphenation algorithms, spelling checkers, etc.)

To stem or not to stem?

Benefits of stemming

With automatic conflation or stemming, several benefits are desired (Lennon et al. 1981, Krovetz 1993, Pirkola 2001):

First, when an automatic algorithm produces correct stems or conflates the word variants into the same form, a searcher does not need to worry about the correct truncation point of a search term. He or she just enters the search term and the algorithm takes care of the rest. This is useful especially when a word is inflectionally very complicated, as there is a risk that an inexperienced searcher may forget some of the inflected stems. If some stems are missed, the searcher may lose relevant information.

Second, conflation reduces the total number of distinct index entries and thus is likely to reduce the size of the index, too.

Third, increase in recall is gained, as conflation can be used as a query expansion method. With a single canonical form, all the word variants are brought together, although they as character strings differ from each other.

Fourth, conflation—or actually basic word form generation with dictionary lookup—can improve precision. When basic word forms are used, a searcher is able to match an exact search term to an exact index term. Such accuracy is not possible with truncated, ambiguous stems.

Stemming algorithms for the English language

Although it is commonly accepted that stemming is beneficial, there have been opposite views, too. One of the most quoted articles on stemming algorithms is Donna Harman's article "How effective is suffixing?" in 1991. She noted that there actually is no empirical evidence that stemming algorithms increase retrieval performance.

Harman examined this by testing three different stemming algorithms in the best match IRX retrieval system, namely Lovins, Porter and "S" algorithms. She concluded that the use of the three algorithms did not result in overall improvements in retrieval performance in her test collections. The number of queries with improved performance tended to equal the number of poorer performance, for which Harman did not find any reasonable explanation.

Before Harman's study, however, there had been a study by Lennon et al. (1981) where seven different stemming algorithms were compared. Lennon and his colleagues found out that stemming always performed better than no stemming.

Interestingly, Keen (1991) came to different conclusion than Harman when he analyzed Harman's findings. Both Harman and Keen noted that performance differences between stemming and no stemming were minute, when compared with traditional measures: statistically significant differences were hardly found out. Keen, however, noted that even in that case one should take the consistent trend into account. In more than half of the cases in Harman's study, stemming (any of the stemming algorithms) produced better results than no stemming. Only in ten per cent of cases, unstemmed words performed better than stemmed. In the rest of the cases, the stemmed and unstemmed words produced equal results. Keen's conclusion was that the results slightly favor stemming to non-stemming.

Later, Krovetz (1993) tested four stemming algorithms for English language: Porter algorithm, improved Porter algorithm and two algorithms by Krovetz himself: an inflectional stemmer and a derivational stemmer. All stems produced by the algorithms were refined with a dictionary-lookup: when an entry matching to the stemmed word was found in the dictionary, the process was finished—otherwise, a new round of stemming procedure was used. According to Krovetz, stemming proved clearly useful. He also remarked that morphologically tuned stemming producing words instead of truncated stems gives better performance than mere affix stripping.

Hull (1996) has so far performed the most thorough study. He compared five different algorithms, including two algorithms developed by Xerox. Among other things, he noted that the size of retrieval sets should be taken into account. When only small retrieval sets are used even in very large test collections, the differences between various methods may not be remarked. When measuring precision-recall score, one should use larger and more realistic retrieval sets in large collections (document levels like 10, 50, 100, and 500 instead of small, "traditionally" used document levels of 5, 10, 15, and 30 documents). Hull's finding would explain, why Harman (1991) did not find out any differences in her tests—the retrieval sets were simply too small to show differences between the algorithms investigated.

Hull also found out that stemming is always useful, when the queries are short. With short queries and short documents, the derivational stemmer is most useful, because it is most likely to conflate different word variants—in long documents, however, the derivational stemmer brings more non-relevant documents.

Stemming in other languages

Variation in morphological properties among the world's languages is high. In languages other than English, stemmers or basic word form generators have been found even more useful than in projects dealing with English texts.

Popovič and Willett (1992) had the same text material both in English and in Slovenian. They found out a substantial performance difference between the conflated and the non-conflated Slovenian text. Their difference was far greater than observed in stemming tests using English documents only. The found differences clearly were due to the properties of the language and not to the particular test data.

Savoy (1999) noted that removing of plural suffixes in French texts had positive effects on precision. He also noted that conflation was more useful with a collection of short documents than with longer documents.

Kalamboukis (1995) tested stemming with modern Greek texts and noted that stemming improved both recall and precision when compared to non-stemming.

Abu-Salem et al. (1999) documented that stems or roots are useful index terms for Arabic. Their benefits were clearer than for English, as the Arabic language is a root based language.

The characteristics of the Finnish language

Inflection

In Finnish, the inflectional and derivational morphology is considerably more complex than that of the Indo-European languages like English. When the words in a text are traditionally stored in their inflected form, this has resulted in uneconomical space requirements for Finnish text compared to those of English texts of corresponding length.

For example, Finnish has more case endings than is usual in Indo-European languages. Finnish case endings correspond to 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). Finnish has 15 cases where English has only two, the nominative and the genitive as in Bill**/'s** (Karlsson 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 enormous number of distinct word forms: a noun may have some 2,000 forms, an adjective 6,000 and a verb 12,000 forms. In addition, the derivational morphology is also rich. The figures mentioned above do not include the effect of derivation, which increases the figures roughly by a factor of 10 (Koskenniemi 1985).

A special phenomenon called consonant gradation makes the inflection even more complicated, as the stem of a word may alter when certain type of endings are attached to it. For example, the word laki 'law' has in practice four inflected stems: laki-, lake-, lai-, and lae-. The common root of the stems consists of only two (2) characters, which means it is impractical to use it as a search term.

Compounds

As documents are usually not retrieved by a single word, several words can be combined into a query. If Boolean logic is used, the idea is to simulate the sentence structure of original text by indicating the relative locations of the word occurrences. This is useful in English language texts, where concepts may be expressed as multi-word terms (phrases, open compounds).

In Finnish, it is typical to use closed compounds instead of separate multi-word terms. The Dictionary of Modern Standard Finnish contains some 200,000 entries, of which two-thirds are compound words (Koskenniemi 1983, p. 68). According to the orthography of Finnish, no spaces or hyphens are inserted between the component words; in the English

language, these compounds would mostly be multi-word terms (e.g. liikevaihtoverotoimisto 'Turnover Tax Bureau').

In English, adjacency operators are of essential importance when searching open compounds, as the constituent parts can be connected together by using an appropriate operator, for example *information*(*W*)*retrieval* where W specifies that the two terms must occur next to each other and in a specified order. In Finnish, the adopted orthographic convention results in a different kind of problem: how to retrieve the second or later elements of the closed compounds.

In traditional databases, only the first component of a closed compound word is easy to retrieve by using the standard right-hand truncation. The other components require both left- and right-hand truncation, which may result in poor precision, especially when using short words (e.g. *lintu* 'bird' → puhe**lintu**otanto 'telephone production'). Thus relevant information is hidden and remains irretrievable, if the searcher is not able to recall all possible first constituents. For example, when the searcher wants to retrieve information about power plants, he must enter in addition to the voimala (or voimalaitos) also all possible compound forms like: aurinkovoimala 'solar power plant,' ydinvoimala 'nuclear power plant,' vesivoimala 'hydroelectric power plant,' lämpövoimala 'thermal power station' and so on. Splitting compounds into their components would make retrieval of the second and later components easier.

The FULLTEXT project

Research problems

The characteristics of highly inflectional and compounding languages result in poor system performance when commercial information retrieval systems developed for English are used. It is reasonable to presume that normalizing the inflectional variants of a word to their basic form (standard form) and splitting closed compounds into their components would improve retrieval effectiveness.

In project FULLTEXT, the following research problems were presented:

1. If automatic stemming is used, will the result sets retrieved with stemmed words produce higher average recall and/or precision figures than those retrieved by words truncated by the user?
2. If the words in texts are reduced to their basic form and stored in index in their basic form, and also the search terms are entered in their basic form, will the result sets retrieved with basic word forms have higher average recall and/or precision than sets retrieved with truncated word forms?

The test collection and morphological analysis software

In the FULLTEXT project, natural language analysis modules for Finnish were incorporated into the BASIS information retrieval system, which is based on inverted indices and Boolean logic. The test material consisted of 23, 244 daily newspaper articles. For evaluation purposes, several test databases were produced.

The requests consisted of requests made in a traditional newspaper clipping archive as well as requests for an experimental newspaper database at the University of Tampere. From them, a basic group of 26 requests were selected.

From the basic group, also additional two subgroups were formed: the derivative subgroup with 8 requests and the compound subgroup with 9 requests. The former subgroup contained search terms having many derivational variants. The latter subgroup contained search terms that were compounds that could be broken down into their constituents. The idea of these subgroups was to focus our examination on the special properties of derivatives and compounds, and thus perhaps detect differences which would not be discerned in a general basic group.

When a query contained more than one search term, two test queries were formed: one where the terms were connected with AND operator and another where they were connected with sentence operator.

For the morphological analysis, programs from Lingsoft (*www.lingsoft.fi*) and from Kielikone (*www.kielikone.fi*) were used. There were two types of programs: stem generators and basic word form generators.

When using *stem generators*, the user enters the search word in its basic word form to the program (for example, a nominative singular for a noun). The program analyzes the word and produces a set of stems where all possible inflectional variants are taken into account. Thus the stems can be used as search terms which retrieve all the inflected word forms of the input term. For example, from the input rata 'track, line, orbit,' the following stems are generated (Koskenniemi 1985):

(1) rata → rata, rada, ratoi, radoi, ratoj

The *basic word form generators* and *compound splitters* analyze inflected word forms and generate their *basic word* form. Although these programs work like stemmers, they in strictest sense are not stemmers, because they produce words, not just strip suffixes off. If desired, they also break closed compound words down into their constituents and generate basic forms of the constituents, too. For example, tekstinkäsittelyohjelmalla 'with a word processor' would produce three entries:

(2) tekstinkäsittelyohjelmalla → teksti, käsittely, ohjelma

In a database index, the constituent parts could be marked with some specific tag, if there is a need to separate the constituents of a compound from independent words (e.g. teksti-, -käsittely-, and -ohjelma).

Test databases

There were five test databases or test systems:

Inflected word form database (T1). The inflected word forms were stored as such in the index and the searcher truncated search terms manually. Traditionally, text databases are produced like this.

The **automatic stemming and stem expansion** (T2) integrated a stem generator in a user interface. The index contained the words in their inflected form. A user enters the search term in its basic form. From it, the stems are generated and added to the query, connected to the original search term with OR operator.

Automatic stemming, stem expansion and sifting (T3). The setting was as above, but in addition, the index terms that matched to stemmed words were further analyzed. The idea of this sifting was to weed out accidental words that were not real instances of the search term but just happened to start with a similar character combination.

First, all the word forms matching with search stems were retrieved in the index. Second, the *basic word* forms of these matching inflected word forms were generated. Third, each generated basic word form was compared to the original search term given by the searcher. If it was identical to the search term, the index word was accepted, otherwise rejected.

Basic word form index and queries (T4). The inflected word forms were reduced to their basic form before storing the words into the index. In the retrieval phase, a searcher enters a basic word form as a search term and the index term fully matching to the search term is retrieved.

Basic word form index and queries with compound splitting (T5). The inflected word forms were reduced to their basic forms and, if they were compound words, also broken down into their components before storing the words into the index.

If a searcher wanted to specify that the term sought should appear as a constituent of a compound, he or she added an appropriate truncation symbol to the search term to indicate its position in the compound word. (For example, auto- for the first constituent, -auto for the last one, and, repectively, -auto- for the constituents in the middle of the word.)

Test queries

Benchmark test queries for the inflected word form database (T1) were produced by selecting search terms from requests and truncating the terms manually (symbol MT, "manual truncation"). The request "*autoverotus*" 'taxation of cars' is used here as an example: in this case, the truncated search term was autovero* (the word auto meaning 'car,' verotus 'taxation,' and vero 'tax'). The correctness of the benchmark queries was checked by three professional information specialists.

Test queries for other search environments (T2 – T5) were formulated as follows: The *basic queries* (symbol A) consisted of the same search terms used as the benchmark queries, but the terms were in their basic form, for example: autoverotus.

Using only basic word forms as search terms, however, would not produce optimal results in all cases. In an inflected word form database, the searcher would get only the terms appearing in the nominative case (basic form), whereas manually truncated terms would retrieve also the compounds and possibly also some derivatives starting with the same character string as the search term.

To examine the effect of derivatives, compounds, or both, on search results, the basic query was expanded in three ways:

– In the *derivative query* (AB), the basic query was expanded with the derivatives and/or root of the search terms in their basic form:

  autoverotus OR autovero. (The term auto was not expanded.)

– In the *compound query* (AC), the basic query was expanded with compound words that included the search term:

  autoverotus OR autoverotus- OR -autoverotus- OR -autoverotus.

– In the *combined query* (ABC), the basic query was expanded both with derivatives of the search terms and with compounds that included the search term or its derivatives:

  (autoverotus OR autoverotus- OR -autoverotus- OR -autoverotus) OR
  (autovero OR autovero- OR -autovero- OR -autovero).

When queries contained compounds, additional query types were formed by splitting the compounds into their constituents. The aim of this splitting was to get hold of the second and later components of closed compounds. Therefore, we obtain the additional four query types: The *split basic query* (symbol Aa), consisting of the original search terms and their constituent parts; for example, autoverotus OR (auto AND verotus). In the *split derivative query* (ABab), the derivative query was expanded with the constituent parts of the compound search terms, and in the *split compound query* (ACac), the compound query was expanded with the the constituent parts of the search terms. Finally, in the *split combined query* (ABCabc), the combined query was expanded with the constituent parts of the search terms, for example: (auto OR auto- OR -auto- OR -auto) AND [(verotus OR verotus- OR -verotus- OR -verotus) OR (vero OR vero- OR -vero- OR -vero)].

Analysis of the results

Relevance analysis was performed in a similar manner as in Tenopir and Ro (1990), where three experts analysed the results. The result sets were pooled, making a total of 1488 articles. They were presented to the experts, which judged each article as relevant, somewhat relevant, or non-relevant. According to the judgments, recall and precision figures for each query type were calculated. For the results, statistical test was performed according to the Friedman test (Hull 1996, Conover 1980).

Findings

The variables used in evaluations were as follows: the test database or test environment (from T1 to T5), the type of the operator used to connect search terms (AND or sentence operator), and the group (basic group of 26 queries, derivative subgroup or compound subgroup).

Table 1. Comparison of the search results retrieved with manually truncated terms (T1) to the search results retrieved with automatically generated stems and stem expansion (T2). In the comparisons, only the terms used in requests were used.

		Relative recall				Precision			
Operator	Group	T1/MT	T2/AC	Diff	α	T1/MT	T2/AC	Diff	α
AND	Basic	72.0	63.9	−8.1	0.01	68.0	71.1	+3.1	0.025
	Deriv	82.7	57.1	−25.6	0.01	46.9	54.7	+7.8	0.025
	Comp	88.1	83.1	−5.0	0.1	42.1	47.1	+5.0	0.05
Sent	Basic	60.1	54.2	−5.9	0.01	76.6	77.3	+0.7	–
	Deriv	56.2	37.3	−18.9	0.01	64.3	66.5	+2.2	–
	Comp	62.8	61.0	−1.8	–	81.1	78.1	−3.0	–

Basic: Basic group, $N = 26$; Deriv: Derivative subgroup, $N = 8$; Comp: Compound subgroup, $N = 9$; Diff: Difference of the search results compared to manual truncation in inflected word index (T1); α: Level of statistical significance.

Table 2. Comparison of the search results retrieved with manually truncated terms (T1) to the search results retrieved with sifting (T3). In the comparisons, only the terms used in requests were used.

		Relative recall				Precision			
Operator	Group	T1/MT	T3/AC	Diff	α	T1/MT	T3/AC	Diff	α
AND	Basic	73.8	63.5	−10.3	0.01	68.4	72.8	+4.4	0.05
	Deriv	82.7	53.4	−29.3	0.01	46.9	57.7	+10.8	0.05
	Comp	88.1	76.8	−11.3	0.01	42.1	53.8	+11.7	0.01
Sent	Basic	61.8	54.4	−7.8	0.01	77.0	80.7	+3.7	–
	Deriv	56.2	33.3	−22.5	0.01	64.3	75.2	+10.9	–
	Comp	62.8	59.3	−3.5	–	81.1	77.3	−3.8	–

(Symbol explanations the same as in Table 1, with the exception that $N = 25$ in the basic group.)

Our working assumption was that automatically stemmed search terms (T2) would produce more precise search results than manually truncated terms (T1), because they are longer as character strings. The aim was to provide clarification for the first research problem (presented in the beginning of the previous chapter). The next assumption was that sifting (T3) would increase the precision even more by weeding out words not being actual instances of the search terms.

We indeed found out that within all groups (basic, derivative and compound), automatic stemming (Table 1) and sifting (Table 2) performed better than manual truncation, when the search terms were connected with AND operator. The differences in precision were also statistically significant. But when the search terms were connected with the sentence operator, no significant differences were found. It seems that if the query already is narrow, additional narrowing with more precise stemming does not have much effect.

Unfortunately, automatic stemming (T2) as well sifting (T3) resulted in loss of the recall. In both environments the loss in the recall was higher than profit in the precision. The differences were also statistically significant. The result was due to derivatives—automatically generated stems matched with original search terms, but left useful derivatives out of the result set.

To study the effect of the derivatives on the search results, the queries were expanded with derivatives (e.g. the basic query A was expanded to the derivative query AB, or the compound query AC expanded to the combined query ABC). The derivatives of the search terms were fed in a stem generator and the new stems were added into the query. This expansion brought the recall on the same level as when searching with manually truncated terms. Unfortunately, the precision decreased to as low a level as it was when using manual truncation. It appeared that stems produced from verbs were as short as manually truncated terms, therefore producing similar precision and recall figures to them.

Sifting (T3) had an additional, technical drawback: it required enormous amounts of computer resources and increased the response time with seconds or tens of seconds. So sifting seems to be just a waste of resources, as the benefit of using it was modest.

The second working hypothesis was that when the words are stored into the index in their basic word form and also the search terms are entered in basic form (T4 and T5), the search results are more precise than with manually truncated stems in inflected word form database. It appeared that this hypothesis was correct (Table 3): the precision of a search set retrieved by the basic query was clearly higher than that retrieved with manually truncated terms.

Unfortunately, at the same time the recall collapsed. The differences were also statistically significant. The loss was due to the compounds and derivatives: as already said, the truncated stems retrieve the compounds and often also many derivatives "accidentally."

When the basic queries were enhanced with derivatives, the recall improved. The differences between the original basic queries and the enhanced derivative queries were statistically significant, too. At the same time, the precision decreased. The profit in the recall, however, was clearer than the loss in the precision. But the recall of the queries enhanced

Table 3. Comparison of the search results retrieved with manually truncated terms (T1) to the search results retrieved with basic word forms in basic word form index (T4; in basic word form index with split compounds, T5, the results were the same). In the comparisons, only the terms used in requests were used.

		Relative recall				Precision			
Operator	Group	T1/MT	T4/AC	Diff	α	T1/MT	T4/AC	Diff	α
AND	Basic	72.0	54.7	−17.3	0.01	68.0	75.8	+7.8	0.01
	Deriv	82.7	51.9	−30.8	0.01	46.9	67.3	+20.4	0.01
	Comp	88.1	67.4	−20.7	0.01	42.1	65.7	+23.6	0.01
Sent	Basic	60.1	46.3	−13.8	0.01	76.6	84.2	+7.6	–
	Deriv	56.2	34.3	−21.9	0.01	64.3	88.6	+24.3	0.05
	Comp	62.8	56.1	−6.7	0.05	81.1	89.9	+8.8	–

(Symbol explanations the same as in Table 1.)

with derivatives were still clearly poorer than that of manually truncated terms in inflected word form database.

A similar result was obtained when basic queries were enhanced with compounds. The recall of such an enhanced query was clearly better than that of the basic query, but did not reach the recall of a query with manually truncated terms in inflected word form database. Although the precision of a compound query was higher than when searching with manually truncated terms, the loss in recall was greater than benefit in precision.

So enhancing a query either with derivatives alone or compounds alone does not increase performance enough. When both derivatives and compounds were added to the query, the recall and precision figures improved. Interestingly, in T4 and T5 both the precision and the recall were slightly better than in T1. The differences, however, were not statistically significant.

Last, all the test environments were compared with each other (Tables 4 and 5). This was done by comparing the results retrieved by the most enhanced query type, namely combined query (in compound subgroup the comparisons were performed with split combined query).

The best average recall was achieved in the index containing the basic word forms and components of compound words (T5) (Table 4). The second best was the database containing basic word forms (T4). The database containing inflected word forms (T1) was the third, but did not differ much from the automatic stemming (T2). Sifting (T3) clearly obtained the poorest recall. Differences between T3 and the other databases were systematic, and in most cases also statistically significant.

The precision values had more variance than the recall values (Table 5). The queries had the highest precision in the index where the terms were sifted (T3). The index containing basic word forms (T4) was the second best, but the basic word forms index with split compounds (T5) was almost as good. There was not much difference between the inflected

Table 4. Comparison of average relative recall values in the test group and subgroups.

Group	Oper	T1	T2	T3	T4	T5	Statistical difference
Basic	AND	73.8	73.6	70.0	73.8	75.6	T1, T2, T4, T5 > T3 (0.025)
	Sent	61.8	61.6	59.1	61.9	62.8	T1, T2, T4, T5 > T3 (0.01)
Deriv	AND	82.7	82.7	73.7	83.4	88.8	T1, T2, T4, T5 > T3 (0.1)
	Sent	56.2	56.2	49.9	56.9	59.0	T1, T2, T4, T5 > T3 (0.025)
Comp	AND	88.1	88.1	83.5	88.6	98.4	T1, T2, T4, T5 > T3 (0.01); T5 > T1, T2 (0.1)
	Sent	62.8	62.8	62.2	63.4	71.7	T5 > T1, T2, T3 (0.01); T5 > T4 (0.05); T4 > T3 (0.1)

Basic: Basic group, $N = 25$; Deriv: Derivative subgroup, $N = 8$; Comp: Compound subgroup, $N = 9$; AND/Sent: Operator: AND operator/Sentence operator; T1: Inflected word form database, words truncated by the searcher; T2: Inflected word form database, automatic stemming and stem expansion; T3: Inflected word form database, automatic stemming as above and sifting; T4: Basic word form index and queries; T5: Basic word form index and queries with compounds splitting.

Table 5. Comparison of average precision values in the test group and subgroups (symbol explanations the same as in Table 4.)

Group	Oper	T1	T2	T3	T4	T5	Statistical difference
Basic	AND	68.4	69.1	69.8	70.7	70.7	No differences found
	Sent	77.0	77.0	78.4	77.8	77.7	No differences found
Derivative	AND	46.9	46.9	48.3	48.2	47.8	No differences found
	Sent	64.3	64.3	67.9	66.8	65.4	No differences found
Compound	AND	42.1	43.0	50.4	44.6	46.6	T3 > T1,T2 (0.01); T3 > T4 (0.05); T3 > T5 (0.1); T4,T5 > T1 (0.05);
	Sent	81.1	81.4	81.2	83.7	82.1	No differences found

word form database with manually truncated terms (T1) and automatic stemming (T2), although the latter can be regarded as somewhat better.

Discussion

Normally, the goal of the searcher is to retrieve an appropriate amount of documents at a high level of precision and recall. In practice, however, these two measures often are contradictory: high precision means low recall and vice versa. In the FULLTEXT project, when using basic word forms (with or without splitting compounds, that is, both in T4 and T5), both the recall and the precision were higher than those of the traditional inflected word form index (T1). This means that their search sets contained more relevant documents and less false drops than the sets retrieved via the inflected word form indices. This is in coherence with Krovetz (1993), who stated that algorithms generating words instead of stems gave better performance than mere suffixing.

When automatic stemming was used, the precision figures of result sets in the basic form index (T4) were higher than those obtained with similar search stems in the inflected word form index (T2). Also this indicates that searching in a basic word form index is inherently more precise than when searching in an inflected word form index.

Apparently, it is beneficial to use words instead of stems. Perhaps this is not the case in all languages (ref. Abu-Salem et al. 1999), but in general this is a reasonable conclusion, considering how different Finnish and English morphologically are.

One important point, when searching in the basic form index, is that one should not content oneself searching only with the original search term. With truncated word forms, the searcher often gets useful index terms accidentally, because of the inaccurate nature of truncated search terms.

According to the FULLTEXT project, the recall becomes better when such a basic query is expanded with the derivatives and/or compound words related to the search term. Although the precision simultaneously will decrease, the loss in precision will be smaller than the profit in recall.

So the query in the basic word form index has to be constructed more consciously than when searching with truncated word forms. On the other hand, the searcher has more options to choose between high recall and high precision, which is not possible with imprecise truncated search strings. And basic word forms are, in general, better suited to be used with searching thesauri.

Traditionally, the searcher has pondered over problems of very low abstraction level: how the search terms behave as character strings. With appropriate morphological software it, however, is possible to provide automatic morphological query expansion tools for the user. This frees the searcher to think problems of higher abstraction level: how to cope with the similarities and dissimilarities of conceptual level.

References

Abu-Salem H, Al-Omari M and Evens MW (1999) Stemming methodologies over individual query words for an Arabic IR system. Journal of the American Society for Information Science, 50:524–529.

Alkula R (2000) From plain character strings to meaningful words. Doctoral Thesis, University of Tampere, Tampere. Acta Electronica Universitatis Tamperensis, 51. URL: http://acta.uta.fi/pdf/951-44-4886-3.pdf [in Finnish].

Conover WJ (1980) Practical Nonparametric Statistics, 2nd ed. Wiley, New York, 493 pp.

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

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

Kalamboukis TZ (1995) Suffix stripping with modern Greek. Program, 29:313–321.

Karlsson F (1985), Ed. Computational morphosyntax: Report on research 1981–84: University of Helsinki, Department of General Linguistics, Helsinki. Publications, 13, 178 pp.

Karlsson F (1987) Finnish Grammar. WSOY, Porvoo, 222 pp.

Karlsson F (1994) Yleinen kielitiede [General Linguistics]. Yliopistopaino, Helsinki, 302 pp. [in Finnish].

Keen EM (1991) The effect of stemming strength on the effectiveness of output ranking. In: Jones KP, Ed., The Structuring of Information: Proceedings of Informatics 11 Conference, University of York, 20–22 March 1991. Aslib, London, pp. 37–50.

Koskenniemi K (1983) Two-level morphology: A general computational model for word-form recognition and production. University of Helsinki, Department of General Linguistics, Helsinki. Publications, 11, 160 pp.

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

Krovetz R (1993) Viewing morphology as an inference process. Proceedings of the Sixteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval. Pittsburg, PA: 27 June–1 July 1993. The Association for Computing Machinery, New York, pp. 191–202.

Lennon M, Peirce DS, Tarry BD and Willett P (1981) An evaluation of some conflation algorithms for information retrieval. Journal of Information Science, 3:177–183.

Pirkola A (2001) Morphological typology of languages for IR. Journal of Documentation, 57:330–348.

Popovič M and Willett P (1992) The effectiveness of stemming for natural-language access to Slovene textual data. Journal of the American Society for Information Science, 43:384–390.

Savoy J (1999) A stemming procedure and stopword list for general French corpora. Journal of the American Society for Information Science, 50:944–952.

Tenopir C and Ro JS (1990) Full Text Databases. Greenwood Press, Westport, 251 pp.