

# Automatic generation of frequent case forms of query keywords in text retrieval

Kimmo Kettunen

University of Tampere, Department of Information Studies, Kanslerinrinne 1, FIN-33014 Tampere, Finland  
Kimmo.kettunen@uta.fi

**Abstract.** This paper presents implementations of generative management method for morphological variation of query keywords. The method is called FCG, Frequent Case Generation. It is based on the skewed distributions of word forms in natural languages and is suitable for languages that either have fair amount of morphological variation or are morphologically very rich. The paper reports implementation and evaluation of automatic procedures of variant query keyword form generation with short and long queries of CLEF collections for English, Finnish, German and Swedish. The evaluated languages show varying degrees of morphological complexity.

## 1 Introduction

Morphological variation of textual words and keywords is a well known issue in information retrieval (IR) and needs some sort of management. Roughly put, the need for managing the variation of keywords increases as the morphological complexity of the language increases. For languages like English, it is not crucial, but for languages like Finnish, Turkish, Russian etc. it is much more important for better retrieval results. The first answers to management of 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 have been used less, although they also offer a suitable solution to the problem [1].

Kettunen [2] has divided the methods of keyword variation management into two groups: reductive and generative. The main idea behind reductive methods is that varying word forms are somehow reduced so that relationships between query keywords and index words can be detected. These methods demand both reductive analysis of textual data bases for index formation and reduction of query keywords. What is here called reductive methods have generally been named conflation in the IR literature [3], and the methods include stemming and lemmatization. Methods that generate inflectional stems or full word forms from a given input form may be called generative. With generative methods of keyword variation management textual indexes are left in their original form without any linguistic processing. Reductive methods have been used far more than generative so far, although also generative methods should be of interest e.g. in present web retrieval systems, where very large multilingual indexes may be impractical for reductive methods.

In this paper we shall report IR results of restricted automatic generation of varying query keyword forms for four languages. With English, Finnish, German, and Swedish we have used CLEF 2003 materials. Our purpose is to show the feasibility of the Frequent Case Generation method that has been simulated earlier with languages that are morphologically different: one of the languages, English, is morphologically simple; German and Swedish are somehow complex and Finnish morphologically quite complex in the sense, that it has lots of word form variation that needs to be taken account for. We shall show that by generating the most frequent forms of nominal query keywords quite good IR performance is achieved.

## 2 The FCG Method

The FCG method has been earlier presented for management of morphological variation of query words with Finnish, Swedish, German and Russian in Kettunen and Airio [4] and Kettunen and colleagues [5]. The FCG method and its language specific evaluation procedure are characterized as follows:

1) For a morphologically sufficiently complex language the distribution of nominal case/other word forms is first studied through corpus analysis. The used corpus can be quite small, because variation at this level of language can be detected even from smaller corpuses. Variation in textual styles may affect slightly the results, so a style neutral corpus is the best.

2) After the most frequent (case) forms for the language have been identified with corpus statistics, the IR results of using only these forms for noun and adjective keyword forms are tested in a well-known test collection. As a comparison the best available reductive keyword and index management method (lemmatization or stemming) is used, if such is available. The number of tested FCG retrieval procedures depends on the morphological complexity of the language: more procedures can be tested for a complex language, only a few for a simpler one.

3) After evaluation, the best FCG procedure with respect to morphological normalization is usually distinguished. The testing process will probably also show that more than one FCG procedure is giving quite good results, and thus a varying number of keyword forms can be used for different retrieval purposes, if necessary.

It should be noted, that the FCG method does not usually outperform golden standard, usage of a lemmatizer, for morphologically complex languages. It provides, however, a simple and usually easily implementable alternative for lemmatization for languages that might lack language technology tools for information retrieval.

Based on this method, Kettunen and Airio [4] first evaluated four different FCG procedures in two different full-text collections of Finnish, TUTK (with graded relevance assessments, Sormunen [6]) and CLEF 2003 (with binary relevance). The results of [4] showed that frequent case form generation works in full-text retrieval of inflected indexes in a best-match query system (Inquery) and competes at best well with the gold standard, lemmatization, for Finnish. The best FCG procedures in Kettunen and Airio. [4], FCG\_9 and FCG\_12<sup>1</sup>, achieved about 86 % of the best average precisions of FINTWOL lemmatizer in TUTK and about 90 % in CLEF 2003. Kettunen and colleagues [5] tested the method with three new languages, German, Russian and Swedish. With German and Swedish the results were positive, but Russian results were reported to be inconclusive most obviously due to the limits of the Russian collection used.<sup>2</sup>

So far the process of FCG query keyword generation has been simulated in tests, but we have now implemented fully automatic query generation using word form generators of four languages: English, Finnish, German and Swedish. Three of the languages are morphologically at least moderately rich and English has been included to see, how a morphologically simple language behaves with the same approach.

## 3 Materials and Methods

CLEF collections for English, Finnish, German and Swedish were utilized in this study. The used retrieval system was Lemur [8]. Lemur combines an inference network retrieval model with language models, which are thought to give more sound estimates for word probabilities in documents [9, 10]. In Table 1, the number of documents and topics with relevant documents in each collection is shown.

---

<sup>1</sup> Here 9 and 12 denote number of variant keyword forms used in the procedure. These figures are a fraction of all the possible grammatical noun forms (1872 -  $\approx$  2000) and 35-46 % of the productive noun forms (26).

<sup>2</sup> The limits of the Russian CLEF collection are most clearly expressed in Savoy [7].

**Table 1.** Collections used in the study

Language	Collection	Collection size (docs)	Topics	IR system
EN	CLEF 2003	169 477	54	Lemur
FI	CLEF 2003	55 344	45	Lemur
DE	CLEF 2003	294 809	56	Lemur
SV	CLEF 2003	142 819	54	Lemur

### 3.1 Query Formation and Linguistic Tools Used

Our query formation for all of the languages was based on application of a same type of routine: topics of the collection were first preprocessed and then lemmatized with the lemmatizers FINTWOL, SWETWOL, GERTWOL and ENGTWOL from Lingsoft Ltd. for Finnish, Swedish, German and English. Stop words were omitted. From the base forms of topical words we generated the actual queries with word form generators for each language followingly:

- generation was used only for nouns and adjectives in the topics (except for English, where only nouns are inflected in cases), all words of other parts of speech were left in the form they were in the topic
- only one base form interpretation for each word in the output of lemmatizer was used for generation (first nominal interpretation given by the lemmatizer)
- if the lemmatizer was not able to give a base form analysis for the topic word, it was anyhow given to the generator for generation: this would produce sometimes right generations and sometimes false generations depending on which form the word happened to be in the topic; Assumedly wrong generations will not harm queries because they usually match nothing but right generations might boost performance of the query. This tactic also makes the generation more independent of the lexicons of lemmatizers, which anyhow lack words.

Generators for the four languages were obtained from different sources, free and commercial. Generator for English was obtained from the University of Sussex [11]. For Swedish we used Grim generator [12] from Nada KTH with Java interface of Martin Hassel; the functionality of the generator can be seen also from the Grim web page<sup>3</sup>. For Finnish we obtained Teemapoint's generator FGEN<sup>4</sup> and for German Canoo's WMTRANS<sup>5</sup>. Three of the generators, English, Finnish and Swedish, are rule based and lexiconless, German generator uses large lexicons for generation. The generators were embedded in the query generation process of each language with Unix scripts. The way they were used emulates use in an interactive search system: a user would give the keywords in their base forms and inflected forms of these would be generated using the generator. In our case, the base forms of topical words are produced by lemmatizers of each language. The process of query generation is shown schematically in Figure 1.

<sup>3</sup> <http://skruten.nada.kth.se/grim/>

<sup>4</sup> [www.teemapoint.com](http://www.teemapoint.com)

<sup>5</sup> <http://www.canoo.com/wmtrans/home/index.html>

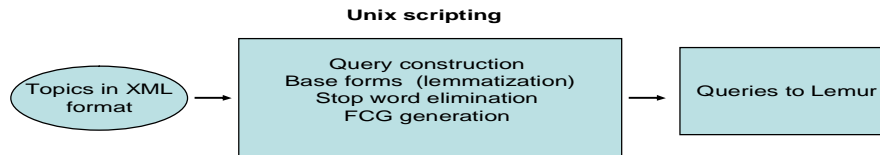


Fig. 1. Generation of queries using FCG generation<sup>6</sup>

### 3.2 Queries

CLEF topics have the following structure: each topic has three parts, title, description and narrative as in the topic #200 in German:

```

<top>
<num> C200 </num>
<DE-title> Hochwasser in Holland und Deutschland </DE-title>
<DE-desc> Finde statistische Angaben über die Hochwasserkatastrophe in
Holland und Deutschland im Jahre 1995. </DE-desc>
<DE-narr> Relevante Dokumente sollen das Ausmaß des Schadens
bezüffern, der durch die Überschwemmungen entstand, die 1995 in
Deutschland und den Niederlanden stattfanden. Die Dokumente sollen die
Wirkung des durch Überschwemmung verursachten Schadens hinsichtlich
der Anzahl von Menschen oder Tieren, die evakuiert wurden, und/oder
hinsichtlich der ökonomischen Verluste bezüffern.</DE-narr>
</top>
  
```

Out of these three parts a query can be formulated using either all the three parts or some combination of parts, usually title and description. We chose to use title and description parts to make long queries and titles only to make short queries that resemble Web queries in the number of words.

The FCG queries were structured with Lemur's #SYN operator so, that all the generated morphological variants of the base form were combined by the same #SYN operator. This way they are handled and weighted as instances of the same form by the query program. As an example a generated title query #200 for Swedish is shown:

```

<query> #combine(#syn(översvämning översvämningar översvämningarna
översvämningen) #syn(holland ) #syn(tyskland ))</query>
  
```

As can be seen, only the word *översvämning* ('flood') has got generated forms during query construction, whereas country names (*Holland* and *Tyskland*) have not been recognized either by the lemmatizer or generator and thus they have been left in the original form.

<sup>6</sup> Due to lack of space no example scripts are included here. Model scripts can be shown upon request.

### 3.3 Distributions

The prerequisite for applying the FCG method is that distributions of nominal case forms for the language need to be known so that only the most frequent nominal forms are generated for keywords in the FCG query construction process. For Finnish, German and Swedish, Kettunen and Airio [4] and Kettunen and colleagues. [5] had analyzed and published the distributions of nouns and adjectives, and we used the same forms in the FCG generation now. For English we needed distributional data.

We analyzed English word form distributions for nouns from three different samples: 228 084 nouns from the Brown corpus, which is morphologically tagged and disambiguated material [13], a sample of 42 064 words from NY Times [14] and 38 723 word forms from the CLEF collection's English material (from Glasgow Herald). The last two samples were run through ENGTWOL lemmatizer and all noun interpretations given by the lemmatizer were counted.

As expected, almost all of the nouns in different corpora of English are in singular or plural nominative. The majority of forms (72–76.5 %) were in singular nominative and 21.8–26.4 % in plural nominative depending on the corpus. The occurrences of genitive were very rare (0.9–1.7 % in plural and singular). Only proper nouns have a bigger share of genitive, as analyzed from the Brown corpus, which distinguishes proper nouns from common nouns. Out of 43 154 proper noun tokens in the Brown corpus 39 045 (90.5 %) were in singular nominative, and 1351 (3.1 %) were in plural nominative. 2716 forms (6.3 %) were in singular genitive and 42 forms (0.1 %) in plural genitive.

From this kind of distribution and scarcity of variation in word forms follows that only generation of English plural nominative besides singular nominative form should yield fairly good IR results. The situation is basically the same as with the so called s-stemmer, but in reverse: while s-stemmer removes plural and genitive *s*, we generate plural forms with the *s* [3]. This procedure is named En-FCG\_2 in the tests. To see the effect of full paradigm generation we also made an En-FCG procedure which includes genitive forms. This procedure is named En-FCG\_2G in the tests.

## 4 Results

### 4.1 Results of English Queries

It was to be expected that English would not benefit much from any of the variation management methods used. So far usually a stemmer that combines both inflectional and derivational stemming has achieved best results for English IR, but the difference between doing nothing and the best method is usually small [15, 16]. For English we compared lemmatization, Snowball stemmer [17], plain query words and the two En-FCG procedures. Table 2 shows results of our short English queries in mean average precisions (MAP, given by trec.eval program) for all the runs and Table 3 shows the results of long queries. Compared methods are coded in the tables as follows: Lemmas (ENGTWOL lemmatizer), Plain (plain query keywords), Stems (Snowball stemmer), EN-FCG\_2 (only nominative forms in singular and plural) and En-FCG\_2G (En-FCG\_2 + genitive).

**Table 2.** English results, title queries, MAP

Lemmas	Plain	Stems	En-FCG_2	En-FCG_2G
0.4102	0.4065	0.4287	0.4201	0.4256

**Table 3.** English results, title-description queries, MAP

Lemmas	Plain	Stems	En-FCG_2	En-FCG_2G
0.4671	0.4467	0.4809	0.4591	0.4472

Figures 2 and 3 show the P/R graphs of short and long English queries.

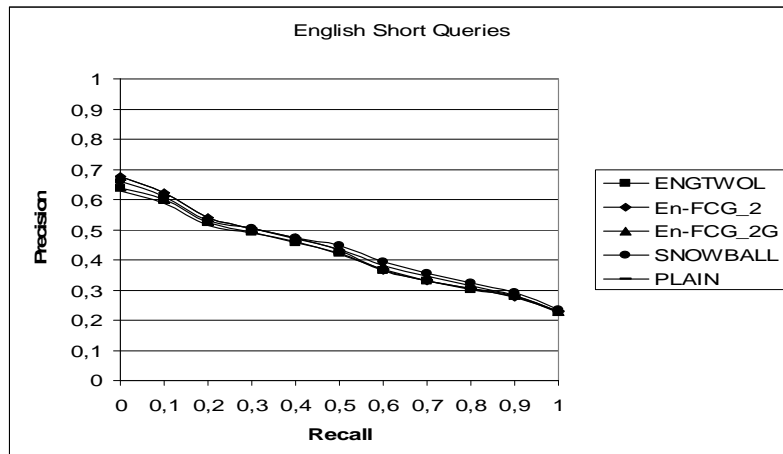


Fig. 2. P/R graphs of short English queries

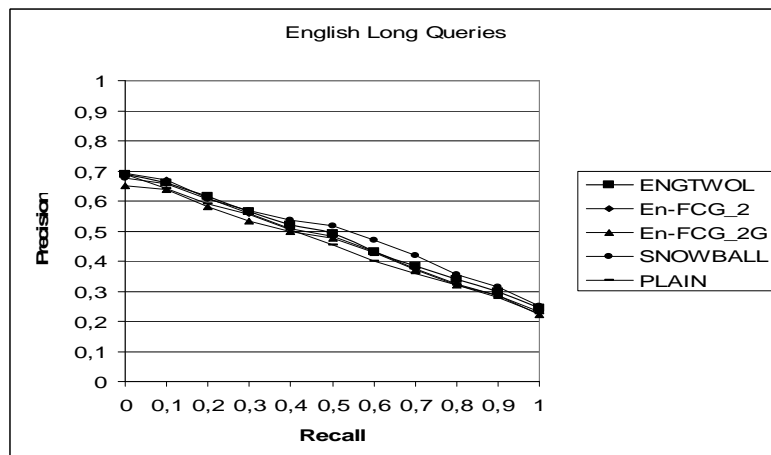


Fig. 3. P/R graphs of long English queries

All the methods perform almost at the same level, and the best mean average precision is achieved by Snowball stemmer both in short and long queries. Lemmatization does not perform very well with English short queries, and it is the second worst method there. With long queries it performs better, being the second best method. Overall the difference of doing nothing to query words and the best achieved results is small, only 2.22 % with short queries and 3.42 % with the long ones. Generation of plural nominative forms for English nouns in En-FCG\_2 with short queries increases MAP about 1.5 % compared to plain query words, and slightly less with long queries. En-FCG\_2G with added genitive forms performs only slightly better than En-FCG\_2 with short queries, but worse with long queries.

Statistical significance of the results was tested using the Friedman test, using the version in Conover [18]. None of the differences between different methods were statistically significant for English.

#### 4.2 Results of Finnish Queries

Finnish was morphologically the most complex language in our tests. Kettunen and Airio [5] had used four different FCG procedures in their tests, but two of the procedures with least word forms did not yield too good IR results. Thus for Finnish we compared two FCG procedures with 9 and 12 variant query word forms with lemmatization, Snowball stemmer and plain query words. Table 4 shows our Finnish results for short queries in mean average precisions for all the runs, and Table 5 shows the results of long queries. The methods compared are coded in the tables as follows: Lemmas (FINTWOL lemmatizer, compounds split in

the index), Plain (plain query keywords), Stems (Snowball stemmer), FCG\_12 (twelve forms of six cases in singular and plural) and FCG\_9 (three cases in singular and plural and three cases in singular only).

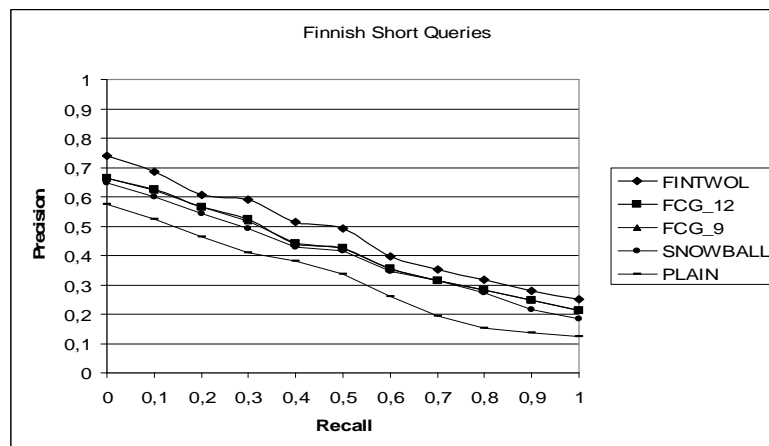
**Table 4.** Finnish results, title queries, MAP

Lemmas	Plain	Stems	FCG_12	FCG_9
0.4525	0.3041	0.3841	0.4028	0.4021

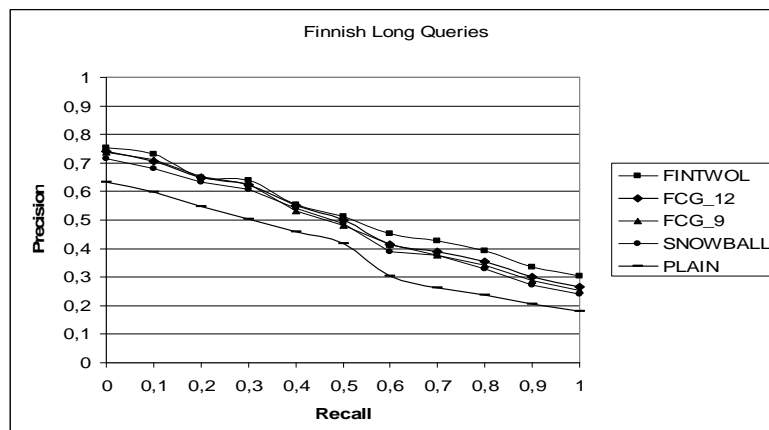
**Table 5.** Finnish results, title-description queries, MAP

Lemmas	Plain	Stems	FCG_12	FCG_9
0.5071	0.3753	0.4624	0.4804	0.4734

Figures 4 and 5 show the P/R graphs of short and long queries of Finnish.



**Fig. 4.** P/R graphs of short Finnish queries



**Fig. 5.** P/R graphs of long Finnish queries

As can be seen from the results, FINTWOL lemmatizer performs best with short queries and both of the FCG procedures perform about 5 % below it, but slightly better than Snowball stemmer. With long queries situation is similar: FINTWOL yields also best results, and both of the Finnish FCG procedures perform

very well being slightly better than Snowball stemmer. The difference of FGCs to lemmatizer is 2.7 – 3.4 per cent.

Comparing the statistical significance of the performance of the methods using the Friedman test gave significant differences ( $p < 0.01$ ) for the entire set of methods. Statistically significant pairwise differences ( $p \leq 0.01$ ) within short and long queries were found between all the variation management methods and plain queries using the Friedman test. There were no statistically significant pairwise differences between lemmatization, stemming and the FCG procedures.

### **4.3 Results of Swedish Queries**

For Swedish we compared lemmatization, the Snowball stemmer, plain query words and two FCG procedures. Table 6 shows results of short queries for Swedish in mean average precisions for all the runs, and Table 7 shows results for long queries. The methods compared are coded in the tables as follows: Lemmas (SWETWOL lemmatizer, compounds split in the index), Plain (plain query keywords), Stems (Snowball stemmer), Sv-FCG\_4 (four forms) and Sv-FCG\_2 (two forms). The  $\mu$  value below the MAP shows the used  $\mu$

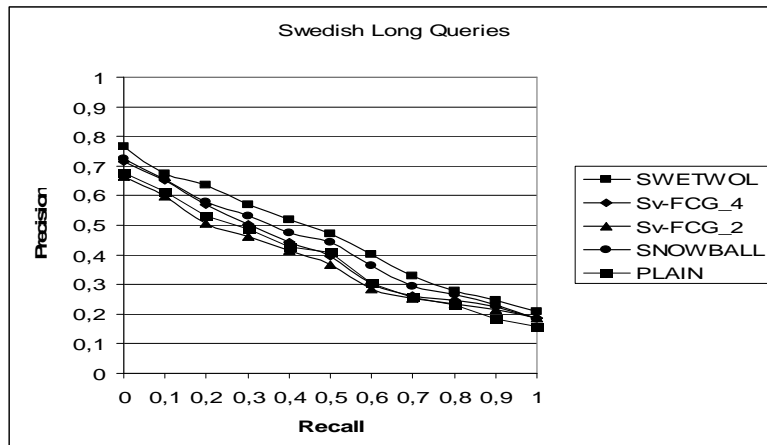


Fig. 7. P/R graphs of long Swedish queries

Our Swedish results for short queries are along the lines of earlier results of [5]. The baseline given by the plain query words with Lemur is 5 % higher than with Inquiry used in Kettunen and colleagues [5], and so are also other results. Lemmatization performs best with short queries, and the difference to the best Swedish FCG is about 2.7 %. Snowball stemmer performs at the same level as the best Sv-FCG procedure.

With long queries SWETWOL lemmatizer is the best method and Snowball stemmer performs second best. Sv-FCGs do not perform very well, and FCG\_2 performs worse than unprocessed query words with long queries.

Comparing the statistical significance of the performance of the methods using the Friedman test gave significant differences ( $p < 0.01$ ) for the entire set of methods. Statistically significant pairwise differences ( $p \leq 0.01$ ) for short queries were found between SWETWOL, Sv-FCG\_4, Sv-FCG\_2, Snowball and plain queries using the Friedman test. With long queries SWETWOL stemmer was significantly better than plain queries and both Sv\_FCGs. Snowball stemmer was also statistically significantly better than Sv\_FCG\_2.

#### 4.4 Results of German Queries

For German we compared also lemmatization, Snowball stemmer, plain query words and two German FCG procedures. Table 8 shows the results of German short queries in mean average precisions for all the runs, and Table 9 the results of long queries. The methods compared are coded in the tables as follows: Lemmas (GERTWOL lemmatizer, compounds split in the index), Plain (plain query keywords), Stems (Snowball stemmer), De-FCG\_4 (four forms) and De-FCG\_2 (two forms).

Table 8. German results, title queries, MAP

Lemmas	Plain	Stems	De-FCG_4	De-FCG_2
0.3524	0.2854	0.3354	0.2962	0.3029
( $\mu= 2500$ )	( $\mu= 2300$ )	( $\mu= 2000$ )	( $\mu= 1800$ )	( $\mu= 2800$ )

Table 9. German results, title-description queries, MAP

Lemmas	Plain	Stems	De-FCG_4	De-FCG_2
0.4456	0.3842	0.4332	0.4158	0.3937
( $\mu= 1500$ )	( $\mu= 2400$ )	( $\mu= 2000$ )	( $\mu= 700$ )	( $\mu= 700$ )

Figures 8 and 9 show the P/R graphs of short and long queries of German.

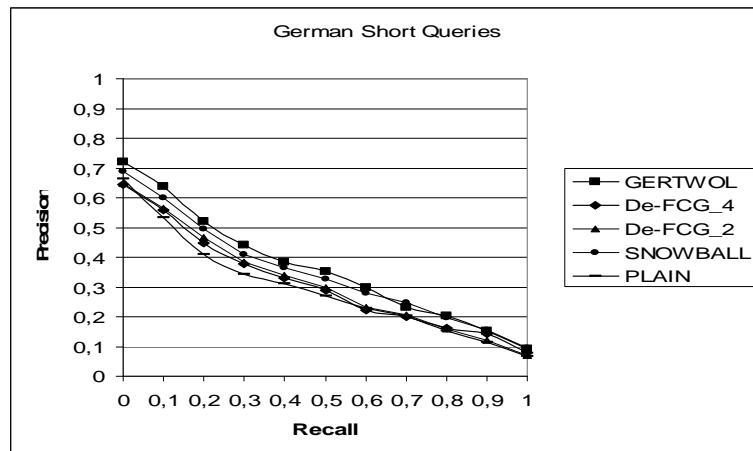


Fig. 8. P/R graphs of short German queries

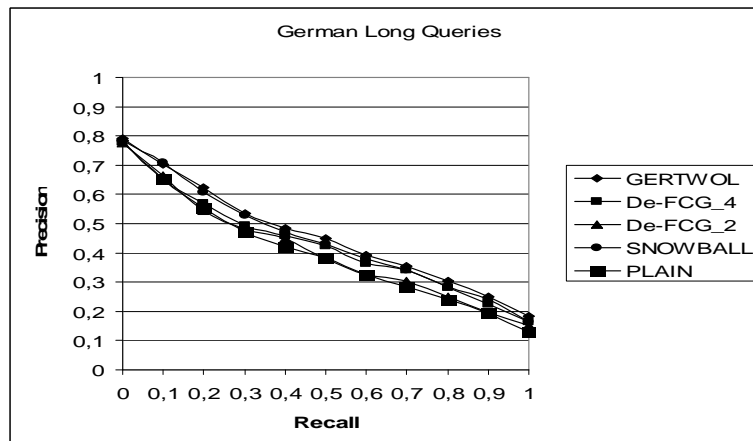


Fig. 9. P/R graphs of long German queries

The results of German short queries seem to be the worst of all the results. The difference between doing nothing and the best De-FCG procedure is only 1.83 % in short queries, opposed to over 4 % earlier with Inquiry in Kettunen and colleagues [5]. Lemmatization did not perform very well with Inquiry, but with Lemur it performs clearly best. Plain query words perform also quite well with Lemur.

Lemmatization was the best method also with long queries. Snowball stemmer is the second best method with long German queries, but De-FCG\_4 performs also quite well here, being only 2.98 % behind GERTWOL lemmatizer. The gap between De-FCG\_4 and plain query words is at best 3.13 % with long queries.

Comparing the statistical significance of the performance of the methods using the Friedman test gave significant differences ( $p < 0.01$ ) for the entire set of methods for short queries. Statistically significant pairwise differences ( $p \leq 0.01$ ) for short queries were found between lemmatization and plain queries and between lemmatization and both of the De-FCG procedures. Also Snowball stemmer was statistically significantly better than plain queries and both of the De-FCG procedures. With long queries statistical significance test of the methods using the Friedman test gave significant differences ( $p < 0.01$ ) for the entire set of methods. In pairwise comparisons GERTWOL and Snowball stemmer were significantly better than plain queries and De-FCG\_2 procedure.

## 5 Discussion and conclusion

We have now set up and evaluated *automatized* FCG query generation with four languages that are morphologically different. English is morphologically simple, Swedish and German moderately complex, and Finnish the most complex of all. English FCGs performed quite well: they gained better mean average precisions than lemmatization (with short queries) or plain query words and were only slightly beyond performance of Snowball stemmer in a setting, where the overall difference between the best and worst method will anyhow be small regardless of the morphological tools used. This shows that the usage of FCG style generation for languages with little morphological variation in words is a feasible alternative to lemmatization and stemming, if morphological tools need to be used with text retrieval.

Our “mid-level” languages with respect to morphological complexity, Swedish and German, got partly expected and partly worse than expected FCG results. Swedish results for short queries were reasonably good, and results for long queries slightly worse. German results were worse than expected, as the German FCGs were only slightly better than plain query words with short queries. In long queries the gap was slightly bigger. Lower than expected German results are most obviously due to the fact, that the German generator version from Canoo uses a 100 000 word lexicon<sup>7</sup> for generation and all the other generators are lexiconless and thus able to cope better with unknown words. Canoo’s generator was unable to generate any inflected forms for about 100 (19 %) of the nominal query words, because they were left either unanalyzed by GERTWOL or otherwise unknown to the generator. Most of these words are either proper names or compound nouns, which many times lack from dictionaries, but are important for the queries. This emphasizes the limits of the lexical lemmatizers and generators and also limits of simulated query procedures used in Kettunen and colleagues [5]: upper limits of performance achieved with simulation may not be achieved with real word generation programs, which have their restrictions. Also impact of the retrieval system needs consideration. As the results of [4, 5] were achieved with InQuery, change of retrieval system to Lemur changes some of the results. Overall plain query words fare better with Lemur than with InQuery, but also other methods yield better results, and thus the relative differences between doing nothing and morphological processing are almost the same with all the languages. This emphasizes the pragmatic or empirical nature of IR as Robertson states [19].

Our Finnish results were good as the results of FCG procedures were only about 2.5-5 % worse than results of lemmatization and slightly better than results of the Snowball stemmer. Finnish FCGs behaved consistently in both short and long queries, being always statistically significantly better than plain query words, and never worse than lemmatization or stemming. FCG procedures fared actually slightly better with Lemur than with Inquery, gaining about 2 % in MAP with both short and long queries.

Our aim in this paper has been setting up an automatized query generation system for several languages with purpose to show feasibility of the FCG method that has been simulated earlier. The results of four languages, although not totally compatible with earlier results, show that the method works well with word forms generators taken off-the-shelf from different sources in a new retrieval system. On the basis of these and earlier findings and common knowledge about word form distributions in texts of natural languages, it is to be expected, that the method will work for other languages of equal morphological complexity as well. Applications of the proposed restricted generation method include in particular Web IR for languages poor in morphological resources and with at least moderate amount of morphological variation that needs management in full-text retrieval. Also the multi-linguality of a web index [20] can be dealt with the approach. As the indexes consist of mixture of word forms in different languages with no linguistic processing (lemmatization or stemming), language specific FCG procedures will yield better retrieval results than usage of plain query words in random textual forms or especially base forms. A natural continuation for work done in this paper would thus be evaluation of web retrieval using the FCG method for a group of different languages showing different degrees of morphological complexity.

## Acknowledgements

This work was supported by the Academy of Finland grant number 124131. We wish to thank Ms. Eija Airio, Dept. of Information Studies, University of Tampere, for implementing all the Unix scripts for the query processes.

## References

1. Sparck-Jones, K., Tait, J.I.: Automatic Search Term Variant Generation. *Journal of Documentation* 40, 50–66 (1984)
2. Kettunen, K.: Reductive and Generative Approaches to Morphological Variation of Keywords in Monolingual Information Retrieval. *Acta Universitatis Tampensis* 1261. University of Tampere, Tampere (2007)
3. Frakes, W. B.: Stemming algorithms. In: Frakes, W. B. & Baeza-Yates, R. (eds.) *Information Retrieval. Data Structures and Algorithms*, pp. 131–160. Prentice Hall, Upper Saddle River, NJ, USA (1992)
4. Kettunen, K., Airio, E.: Is a Morphologically Complex Language Really that Complex in Full-text Retrieval? In: Salakoski, T. et al. (Eds.) *Advances in Natural Language Processing, LNAI 4139*, pp. 411–422. Springer-Verlag, Berlin Heidelberg (2006)
5. Kettunen, K., Airio, E., Järvelin, K.: Restricted Inflectional Form Generation in Management of Morphological Keyword Variation. *Information Retrieval*: 10, 415–444 (2007)
6. Sormunen, E.: A Method for Measuring Wide Range Performance of Boolean Queries in Full-text Databases. *Acta Universitatis Tampensis* 748. University of Tampere, Tampere (2000)
7. Savoy, J.: Searching Strategies for the Bulgarian Language. *Information Retrieval* 10, 509–529 (2007)
8. The Lemur Toolkit for Language Modeling and Information Retrieval, <http://www.lemurproject.org/>
9. Metzler, D., Croft, W. B.: 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 (2004)
10. Grossman, D. A., Frieder, O.: *Information Retrieval. Algorithms and Heuristics*. Second edition. Springer, Netherlands (2004)
11. Minnen, G., Carrol, J., Pearce, D.: Applied Morphological Processing of English. *Natural Language Engineering*, 7, 207–223 (2001)
12. Knutsson, O., Pargman, T.C., Eklundh, K.S., Westlund, S.: Designing and Developing a Language Environment for Second Language Writers. *Computers and Education, An International Journal*, 49, (2001)
13. Brown Corpus Manual, <http://khnt.hit.uib.no/icame/manuals/brown/INDEX.HTM>
14. TDT2 Multilanguage Text Version 4.0., <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T57>
15. Airio, E.: Word normalization and decompounding in mono- and bilingual IR. *Information Retrieval*, 9, 249–271 (2006)
16. Hollink, V., Kamps, J., Monz, C., de Rijke, M.: Monolingual Document Retrieval for European Languages. *Information Retrieval* 7, 33–52 (2004)
17. Snowball, <http://snowball.tartarus.org/>
18. Conover, W. J.: *Practical Nonparametric Statistics*. 3rd ed. Wiley, New York (1999)
19. Robertson, S.: Salton Award Lecture. On Theoretical Argument in Information Retrieval. *ACM Sigir Forum* 34, 1–10 (2000)
20. Rasmussen, E. M.: Indexing and Retrieval for the Web. In: Cronin, B. (ed.), *Annual Review of Information Science and Technology* Vol. 37, pp. 91–124 (2003)