Improving Precision in Information Retrieval for Swedish using Stemming

Johan Carlberger, Hercules Dalianis, Martin Hassel, Ola Knutsson
NADA-KTH
Royal Institute of Technology
100 44 Stockholm, Sweden
ph: +46 8 790 91 05
fax: +46 8 10 24 77
email: { jfc, hercules, xmartin, knutsson}@nada.kth.se

ABSTRACT

We will in this paper present an evaluation[1] of how much stemming improves precision in information retrieval for Swedish texts. To perform this, we built an information retrieval tool with optional stemming and created a tagged corpus in Swedish.

We know that stemming in information retrieval for English, Dutch and Slovenian gives better precision the more inflecting the language is, but precision depends also on query length and document length. Our final results were that stemming improved both precision and recall with 15 respectively 18 percent for Swedish texts having an average length of 181 words.

Keywords

Stemming, Swedish, Information Retrieval, Evaluation

1. INTRODUCTION

Stemming is a technique to transform different inflections and derivations of the same word to one common "stem". Stemming can mean both prefix and suffix removal. Stemming can, for example, be used to ensure that the greatest number of relevant matches is included in search results. A word's stem is its most basic form: for example, the stem of a plural noun is the singular; the stem of a past-tense verb is the present tense. The stem is, however, not to be confused with a word lemma, the stem does not have to be an actual word itself. Instead the stem can be said to be the least common denominator for the morphological variants. The motivation for using stemming instead of lemmatization, or indeed tagging of the text, is mainly a question of cost. It is considerably more expensive, in terms of time and effort, to develop a well performing lemmatizer than to develop a well performing stemmer. It is also more expensive in terms of computational power and run time to use a lemmatizer than to use a stemmer. The reason for this is that the stemmer can use ad-hoc suffix and prefix stripping rules and exception lists while the lemmatizer must do a complete morphological analysis (based on a actual grammatical rules and a dictionary). Another point of motivation is that a stemmer can deliberately "bring together" semantically related words belonging to different word classes to the same stem, which a lemmatizer cannot.

A problem concerning stemming is the issue of overstemming. If the stemmer removes too much in its quest for the stem the result is that, morphologically or semantically, unrelated words are conjoined under the same stem. For example, if both the words *tiden* ("time") and *tidning* ("newspaper") are stemmed to *tid*, a search for *tidning* also would return documents containing *tiden*. The graveness of this problem depends on both the set of stemming rules and the document collection

[1] This project is supported by NUTEK (Swedish board for Industrial and Technical Development) FavorIT programme in cooperation with Euroseek AB.

(more precisely; the index terms used to index the document collection). This is due to the fact that both the set of rules and the set of index terms used influence the amount of index terms conjoined under the same stem.

Here follows a number of algorithms previously used to find the stem of a word, (these are not using static lexicons which also can be used but are not so general).

The so-called Porter stemmer (Porter, 1980) for English, which removes around 60 different suffixes, uses rewriting rules in two steps. The Porter stemmer is quite aggressive when creating stems and does overstemming, but still the Porter stemmer performs well in precision/recall evaluations. KSTEM is another stemmer described in (Krovetz, 1993). KSTEM is not as aggressive as the Porter stemmer, and it does not create as many equivalence classes as the Porter stemmer does. KSTEM is also considered more accurate, but does not produce better results in evaluation experiments. A stemmer for Slovene is described in Popovic & Wilett (1992). Since Slovene is morphologically more complicated than English, the Slovene stemmer removes around 5 200 different suffixes. A Porter stemmer for Dutch is described in Kraaij and Pohlman (1994).

Based on the work in constructing a Swedish tagger (Carlberger & Kann 1999) we developed techniques to find the stems of Swedish words and we have used these techniques in our information retrieval work. Our stemming algorithm for Swedish uses about 150 stemming rules. We use a technique where we, with a small set of suffix rules, in a number of steps modify the original word into an appropriate stem. The stemming is done in (up to) four steps and in each step no more than one rule from a set of rules is applied. This means that 0-4 rules are applied to each word passing through the stemmer. Each rule consists of a lexical pattern to match with the suffix of the word being stemmed and a set of modifiers, or commands, see Figure 1.

The technique is quite general and can easily be adapted to inflectional languages other than Swedish.

*****	Don‘t remove or replace anything
–	Remove matched if a preceeding vowel is found
+	Remove matched
=	Remove matched if matching the whole word
.	Stop matching (break)
abc	Replace with *abc*

Figure 1. The set of commands applicable to words being stemmed.

In step 0 genitive-s and active-s are handled; these are basically –s stripping rules. Definite forms of nouns and adjectivies are handled in step 1, as well are preterite tense and past participle.

hals	***.**	Don‘t remove or replace anything and stop matching. ("neck")
abel	**–**	Remove matched if a preceding vowel is found
sköt	**+.skjut**	Remove **sköt**, insert **skjut** ("shoot" or "push") and break

Figure 2. Example of exception rules.

In step 2 mainly plural forms of nouns and adjectives are handled. Noun forms of verbs are handled in step 3. In step 3 there are also some fixes to cover exceptions to the above rules, see Figure 2.

A word's stem does not have to be of the same part of speech as the word; in whatever sense you can talk about part of speech for the stem. The rules are designed so that word classes can be ‘merged‘. This means that, for example, **cykel** ("bicycle") and **cyklade** ("rode a bicycle") are both stemmed to **cykl**.

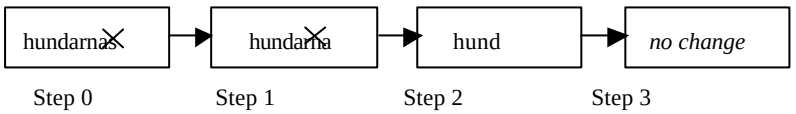

Figure 3. Example of stemming of the word *hundarnas* (the dogs' (genitive form)) to *hund* (dog).

This technique, see Figure 3, works well when the stem is to be used as a internal representation for a set of morphological variants and semantically related words. The stems themselves are, however, too cryptic to be presented to the user as bearing any information.

2. PRECISION AND RECALL[1] IN INFORMATION RETRIEVAL

Regarding information retrieval, there have been experiments using stemming of texts before indexing, or query expansion of the query before retrieving the text collections to investigate the improvement on precision. These experiments have been made for English but also for Slovene and Dutch.

Xu & Croft (1998) describe that stemming at document indexing time is more computational efficient than at query time (query expansion). Query expansion and stemming in information retrieval are regarded as equivalent, but most experiments have been carried out with stemming both on the document collection and on the query, i.e. normalization of both the query and text. (One can also just use query expansion on the query and no stemming on the document collection. Query expansion means that all possible inflections of a word are generated)

Popovic & Wilett (1992) found that there is no difference in precision using manual truncation of the query and automatic stemming; both methods gave the same results, at least for Slovene texts.

The first investigations by Harman (1991) indicated that there were no significant improvement in the retrieval using stemming, but in a later study by Krovetz (1993), an improvement of the retrieval (around 40 percent increase in precision) was proven specifically for shorter documents (average 45 words) with short queries (average 7 words). Longer texts (average 581 words) and with short queries (average 9 words) gave only 2 percent increase in precision.

According to Hull (1996), stemming is always beneficial in retrieving documents, around 1-3 percent improvement from no stemming, except on very small document collections.

Popovic & Wilett (1992) showed that stemming on a small collection of 400 abstracts in Slovene and queries of average length of 7 words increased precision in information retrieval with 40 percent.

In the above experiments the relation between the number of documents (500 to 180 000 documents) in the document collection and the number of unique questions range between 0.1 percent and 10 percent of the document collection.

3. THE KTH NEWS CORPUS

From the KTH News Corpus, described in detail in Hassel (2001), we selected 54 487 news articles from the period May 25, 2000 to November 4, 2000. From this sub-corpus we randomly selected 100 texts and manually tagged a question and answer pair central to each text; see Figure 4, for an example.

Question

<top>

<num> Number: 35

<desc> Description: (Natural Language question)

[1] Precision = number of found relevant documents / total number of found documents

Recall = number of found relevant documents / total number of relevant documents

Vem är koncernchef på Telenor? (Who is CEO at Telenor?)

</top>

Answer

<top>

<num> Number: 35

<answer> Answer: Tormod Hermansen

<file> File: KTH NewsCorpus/Aftonbladet/Ekonomi/0108238621340_EKO__00.html

<person> Person: Tormod Hermansen

<location> Location: Norden

<organization> Organization: Telenor

<time> Time: onsdagen

<keywords> Keywords: Telenor; koncernchef; teleföretag; mobilmarknaden; uppköp

</top>

Figure 4. Questioning and answering tagging scheme

4. EVALUATION

Our information retrieval system uses traditional information retrieval techniques extended with stemming techniques and normalization of both the query and text. (The system can also be executed without using the stemming module).

We used a rotating questioning-answering evaluation schema to avoid training effects of running the information retrieval system. Each of three users answered 33, 33 and 34 questions respectively with and without stemming functionality. The three users were not allowed to do more than five trials on each question to find the answer and were not allowed to use longer queries than five words. No background knowledge was allowed, which means that only the words used in the natural language question were allowed. Boolean expressions and phrase searches were allowed but rarely used.

After going through all of the 100 questions and finding answers to these, that is 33 questions each, we rotated the work and we became evaluators of the previous persons' answers assessing how many of the found top ten answers were correct and how many were wrong.

Of the 100 questions, the test persons found 96 answers, 2 questions did not give any answers at all and 2 other questions gave unreadable files. Each of the asked queries had an average length of 2.7 words. The texts containing the answer had an average length of 181 words.

We found a 15 percent increase on precision on the first 10 hits for stemming compared to no stemming (see Table 1). We also compared with weighting the first hits higher than the last ones and we found no significant difference: 14 percent better with stemming and weighting. (We gave the first hit a weighting factor of 10 and the second hit a weighting factor of 9, decreasing the weighting factor until the last tenth hit giving it 1 and then we normalized everything to 1).

Table 1. No stemming versus stemming

Precision/Recall at 10 first	**Wordform**	**Stemming**	**Weighted Wordform**	**Weighted Stemming**
Number of questions	96	96	96	96
Average precision	0.255	0.294	0.312	0.353
Increase of precision %		15.2%		13.1%
Average relative recall	0.665	0.784		
Increase of relative recall %		18.0%		

Regarding the recall, we calculated the relative recall. Maximum number of recalled texts per question is 21 (=10+10+1). This is calculated using the found unique or disjunctive texts when retrieving using both no stemming and stemming and also adding the tagged correct answer. We calculated the increase in recall taking the difference of the average relative recall, and we found an improvement of 18 percent on relative recall using stemming.

5. CONCLUSIONS

Stemming (and/or manual truncation) can give better precision (4-40 percent) in information retrieval for short queries (7-9 words) on short documents (500 words) than no stemming at all for languages as English, Dutch and Slovenian. Our experiments show that stemming for Swedish can give at least 15 percent increase in precision and 18 percent increase on relative recall depending on the set of rules and the document collection. We are convinced that the cost in creating a stemmer is proportional to the gain when using the stemmer. This indicates that using stemming on morphologically complicated languages will give great gain in precision.

ACKNOWLEDGMENTS

We would like to thank the search engine team and specifically Jesper Ekhall at Euroseek AB for their support with the integration of our stemming algorithms in their search engine and allowing us to use their search engine in our experiments.

6. REFERENCES

J. Carlberger and V. Kann. 1999. *Implementing an efficient part-of-speech tagger*, Software Practice and Experience, 29, 815-832, 1999. ftp://ftp.nada.kth.se/pub/documents/Theory/Viggo-Kann/tagger.pdf

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

M. Hassel. 2001. *Internet as Corpus – Automatic Construction of a Swedish News Corpus.* NODALIDA '01 - 13th Nordic Conference on Computational Linguistics, May 21-22 2001, Uppsala, Sweden.

D.A. Hull. 1996. *Stemming Algorithms - A Case Study for Detailed Evaluation.* Journal of the American Society for Information Science, 47(1): 70-84

W. Kraaij and R.Pohlmann. 1994. *Porter's stemming algorithm for Dutch.* In L.G.M. Noordman and W.A.M. de Vroomen, editors, Informatie wetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie, pp. 167-180.

R. Krovetz. 1993. *Viewing Morphology as an Inference Process.* In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, pp 191-202.

M. Popovic and P. Willett. 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.

M.F. Porter. 1980. *An algorithm for suffix stripping.* Program, vol 14, no 3, pp 130-130. (Se also http://open.muscat.com/developer/docs/porterstem.html)

J. Xu and W. B. Croft. 1998. *Corpus-based Stemming using Co-occurrence of Word Variants.* ACM Transactions on Information Systems, Volume 16, Number 1, pp 61-81, January 1998.