Hand-crafted versus Machine-learned Inflectional Rules: The Euroling-SiteSeeker Stemmer and CST's Lemmatiser

Hercules Dalianis and Bart Jongejan

DSV
KTH - Stockholm university
Forum 100
164 40 Kista, Sweden
E-mail: hercules@kth.se

CST
University of *Copenhagen*
Njalsgade 80
2300 København S, Denmark
bart@cst.dk

Abstract

The Euroling stemmer is developed for a commercial web site and intranet search engine called SiteSeeker. SiteSeeker is basically used in the Swedish domain but to some extent also for the English domain. CST's lemmatiser comes from the Center for Language Technology, University of Copenhagen and was originally developed as a research prototype to create lemmatisation rules from training data. In this paper we compare the performance of the stemmer that uses handcrafted rules for Swedish, Danish and Norwegian as well one stemmer for Greek with CST's lemmatiser that uses training data to extract lemmatisation rules for Swedish, Danish, Norwegian and Greek. The performance of the two approaches are about the same with around 10 percent errors. The handcrafted rule based stemmer techniques are easy to get started with if the programmer has the proper linguistic knowledge. The machine trained sets of lemmatisation rules are very easy to produce without having linguistic knowledge given that one has correct training data.

1. Introduction

It is well known that stemming or lemmatisation in a search context for Swedish and many other languages produces both better precision and recall, (Carlberger et al. 2001). The Swedish company Euroling AB has developed a suffix based stemmer, for Swedish, Danish, Norwegian and English, that is used in their commercial web site and intranet search engine SiteSeeker. CST has developed a lemmatiser that generates its own lemmatisation rules from a word list containing inflected forms and their corresponding lemma forms (Jongejan & Haltrup 2005). CST's lemmatiser can handle languages with suffix-based morphology, such as the Scandinavian languages, English and Greek. Euroling's stemmer contains around 1 000 hand made rules for each language to reduce inflected words that have the same lemma to a common stem.

2. The Stemmer

The Euroling stemmer was developed in 2001 for Swedish (Carlberger et al. 2001) for the search engine SiteSeeker. In 2003 and 2005 respectively was the Danish and Norwegian stemmer developed.

The stemmer is based on a suffix rule file for each language and a C++ program that executes the rule file.

There are about 1 000 rules for each language including exceptions (300 suffix rules and 700 exception rules).

The 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. For example the Swedish word *böckernas* (of the books) with umlaut will be stemmed to *bok* (book)

> *Böckernas* => (remove genitive *s*) *böckerna* => (match *rna*, remove *na*) *böcker* => (match *böcker* replace by *bok*) => *bok*

Of Euroling's stemmers the Swedish stemmer is the most elaborated since it is used in the commercial search engine SiteSeeker and hence continuously updated.

The Greek stemmer has been developed by a Greek master student at KTH, Georgios Ntais (2006). The stemmer is written in Javascript and based on the Porter algoritm. The Greek stemmer has 41 preprocessing rules to exclude a group of words from the stemming process. 158 suffix rules and 517 exception rules for various words or group of words. These exception rules "protect" the different group of words from possible suffix removal. The Greek stemmer operates only on capitalized Greek to avoid the diacritics that is used with Greek lower case letters.

3. The Lemmatiser

CST's lemmatiser was developed in 2002 to make life easier for the lexicographers who were collecting words for STO – a large computational lexicon for Danish, (STO 2006, Braasch & Olsen 2004). By 2002, STO already included most words in the general domain and now lexicographers had to sweep corpora in specialised domains to find new candidate words. Frequency information was needed for setting coding priorities for these candidate words, but the words had to be counted by their normalised lemma form, not their full form. There was little time to develop a lemmatiser, and it was therefore decided to develop a lemmatiser that could train its own lemmatisation rules from the words already present in the lexicon. In that way the costly process of hand-crafting the lemmatisation rules was completely avoided.

Basically the lemmatiser is a simple tool. In contrast to the stemmers discussed above, CST's lemmatiser does the transformation from full form to lemma by the application of one single rule. Each rule consists of a search pattern and a replacement string. The search pattern is compared with the word's ending and may comprise any number of characters, in some cases matching all of the word. If the search pattern is successfully matched, all matched characters are removed and the replacement string is inserted in its place. For example, the Danish word *bøgernes* (of the books) is lemmatised by application of the rule *øgernes* → *og* to the word *bog* (book). Only the rule with the longest matching search pattern is applied. In some cases, two or more rules have the same search pattern (but different replacement strings). In such cases, more than one lemma is produced. The lemmatiser handles hundreds of thousands words in a matter of seconds, the huge number of lemmatisation rules notwithstanding.

The training of the lemmatiser is a process with many iterations that let the lemmatisation rules converge to a near-optimum. This process can take many minutes, but it has to be done only once, because the rules are written to a (human readable) file. After training, the lemmatiser can produce all lemmas of all words that partake in the training set and produces no lemmas that were not in the training set. Only when applied to words that were not in the training set does the lemmatiser make some mistakes.

4. The Training and Test Data

Since we developed the SweSum text summarizer (Dalianis 2000, de Smedt et al. 2005) for ten languages, we have also obtained access to keyword dictionaries with full forms and their lemmas for Swedish (Stockholm Umeå Corpus 2006), Danish (STO 2006, Braasch & Olsen 2004), Norwegian (SCARRIE 2006) and Greek (NCSR Demokritos Part-of-speech dictionary 2006, Petasis 2003). Each keyword dictionary contains nouns, verbs and adjectives, but also other word classes are found. We trained the CST-lemmatiser on these key word dictionaries and we selected words from them to test both the stemmer and the lemmatiser.

Inspection of the word lists showed that the morphology in general was suffix based, but that there also were remarkable deviations in three of the four lists. The Norwegian word list has two variants (*bokmål* and *nynorsk*) of many words, for example *øsekar/ausekar* (bailer) or *fløtemysost/fløytemysost* (Norwegian easy spread brown cheese).

All words have only bokmål lemmas, however, causing many words to have lemmas that are rather dissimilar from the full form in other places than the suffix.

	Full forms	Lemmas
Swedish	477 920	40 194
Danish	412 534	55 181
Norwegian	430 305	63 117
Greek	564 699	52 158

Table 1: The keyword lists

To a lesser degree the same was true for the Danish wordlist, which has spelling variants like *raison/ræson* and *tsar/zar/czar*. Greek has many diacritics with positions that depend on the inflection. For software like the lemmatiser, accented characters are just as different from their unaccented counterpart as two different characters.

5. Training of the Lemmatiser

CST's lemmatiser is a tool that can be trained with different options to maximise its usefulness for different settings. Like the Euroling stemmer, the lemmatiser given any word as input, always returns an answer and this answer may be correct or wrong. But there is a third possibility: The lemmatiser may return two or more lemmas, each of which can be correct. In many settings it is undesirable to have ambiguous data and one would rather prefer to obtain just the most probable lemma and forget about the other lemmas. To mimic such a setting, we counted ambiguous results as "errors" for the purpose of this evaluation and therefore it is worthwhile knowing how to do the training to minimise the number of ambiguous lemmatisation rules.

Extended testing of the lemmatiser has shown that the number of ambiguous lemmatisation rules grows significantly as the training set grows. The growth *rate* even increases, because as words are added to the training set, each word has a probability to introduce an ambiguity with an already present word that is proportional to the steadily growing number of words already present. To a good approximation, the number of ambiguities grows quadratically with the number of training words. A quadratic growth of ambiguities necessarily outruns a linear growth of training data and so, unless one takes precaution not to include ambiguous words in the training data, one eventually will see lemmatisation results becoming less useful when one adds words to the training data.

Another adverse effect of big training data sets is a tendency to produce more curious results which are a consequence of the inclusion of curiously inflected words in the training data. One can easily avoid these unwanted (and often incorrect) results by instructing the lemmatiser to delete all lemmatisation rules that are based on fewer than, say, two examples from the training data. Not only does this remove many rules that tend to produce wrong results, it also removes most, if not all, ambiguous rules. On the downside, this not only reduces the number of incorrectly or ambiguously lemmatised "unknown" words, it also reduces the number of correctly lemmatised "known" words, "known" words being the words that the lemmatiser has been trained with. The latter effect can however easily be counteracted by running the lemmatiser with an extra option: the inclusion of the training data as a look-up dictionary. With this option set, the lemmatiser first tries to find a full form in the dictionary and uses the lemma of the dictionary if it is found there. If the full form is not present in the dictionary, the lemmatiser uses the lemmatisation rules to compute the full form of the lemma. In this way we get the best of both worlds: Optimal lemmatisation of known words by dictionary look-up and optimal lemmatisation of unknown words by applying the pruned rule set. As a bonus we obtain a much smaller set of rules, which is quicker to load and apply.

The CST lemmatiser produced from the data 90 000 rules for Danish, 50 000 rules for Swedish, 61 000 rules for Norwegian and 55 000 rules for Greek, i.e. the CST-lemmatiser generalises down to about 10 times less rules than the example data, while the 1 000 handmade stemming rules are 50 to 100 times less than the example data (keywords dictionaries).

The lemmatiser was trained with word lists that did not contain all word classes. Adding more word classes, for example by training with the STO-database for Danish that contains all word classes, allows us to safely lemmatise a text without first filtering unsupported word classes away. However, adding more word classes also adds more ambiguous rules. This effect can be counteracted by lemmatising POS-tagged input, but then the correctness of the result is also affected by errors the POS-tagger may introduce.

6. Evaluation

The evaluation of the Euroling stemmer and the CST lemmatiser was carried out in slightly different ways so as to acknowledge that they obtain their linguistic resources in different ways and that they have to perform slightly different tasks.

The CST lemmatiser is trained using a word list. Evaluation with words from the training set therefore gives far better results than evaluation with words that are not in the training set. Therefore we present the results for both types of evaluation words.

The Euroling stemmer is not automatically trained, but uses hand crafted rules. Therefore we present just one set of results for this tool. For the evaluation of the Euroling stemmer we had to manually check the results, which meant that we had to restrict the size of the test data. From the stemming results of each keyword dictionary we carefully selected 12 words in each language and evaluated these words: *Artikel, bil, blomma, bok, behandling, cykel, dans, Asien, mord, medel, medlem, projekt*, in English: *Article, car, flower, book, treatment, bicycle, dance, Asia, murder, means, member, project*. These words were selected because they are difficult to write stemming rules for in Swedish, Danish and Norwegian due to double consonant endings and lots of exceptions and "Umlaut". As we actually had run the stemmer for all words in the keyword dictionaries, we easily could enlarge the test by selecting more words. We did this by localising the 12 words and picking all the words that could be seen at the same time. This gave us another 40 words, that is in practice 50 words and their inflected endings for each language.

An important difference between the tools is the way they are expected to handle spelling variants. Whereas the Euroling stemmer does not attempt to bring spelling variants to the same stem, the lemmatiser does so if the training data does so. That means that the lemmatiser had to construct specialised rules to transform e.g. *ausekar* into *øsekar*, while the stemmer just left the word *ausekar* unchanged. Failure to change the spelling variant to the canonical form was counted as an error in the case of the lemmatiser, but not in the case of stemmer. Likewise, wrong diacritisation of Greek lemmas was counted as an error in the lemmatiser's case, but not in the case of the Greek stemmer, which did not handle diacritisation from the outset. This put a higher burden on the lemmatiser than on the stemmers when tested with unknown words.

For the evaluation we divided the results in two groups: words that were stemmed or lemmatised correctly and those that were not.

	Euroling stemmer			**CST's lemmatiser**					
	Test data: see text			**Test data = training data**			**Test data ≠ training data**		
	Words	**Errors**	**%**	**Words**	**Errors**	**%**	**Words**	**Errors**	**%**
Swedish	300	26	8.7	477 920	0	0	4 780	411	8.6
Danish	300	26	8.7	412 534	0	0	4 126	251	6.1
Norwegian	300	45	15.0	430 305	0	0	4 313	515	12.0
Greek*)	717	51	7.1	564 700	31 218	5.5	5 647	805	14.2

Table 2: Evaluation results. *) The Greek stemmer is developed separately from Euroling-SiteSeeker

Ambiguous results from the lemmatiser were counted as errors. The word lists used for this evaluation did not have ambiguous words, except for the Greek word list.

This explains why the lemmatiser seemingly performs much worse with Greek than with the other languages, especially with words from the training set, see Table 2

Tests of the lemmatiser with large Danish and English word lists that were not disambiguated gave results very similar to the results for Greek in Table 2 (Jongejan 2006).

The lemmatiser results in Table 2 are obtained without the option to include the training set as a look-up dictionary, in order to make the fairest comparison with the Euroling stemmer, which does not use a dictionary either. In a commercial setting, the rules might be obtained for a much lower price than the full dictionary, which is a much more valuable linguistic resource. As a consequence it cannot be taken for granted that the lemmatiser can be run with the dictionary in practice.

Using the optimum size of the rule set, the error percentage for unknown Greek words becomes one and a half percent points lower than those in Table 2, while the lemmatisation of known Greek words is not affected if the dictionary is included during lemmatisation to compensate for the left-out lemmatisation rules.

For Swedish, Danish and Norwegian the optimum set of lemmatisation rules is the full set.

7. Conclusions

The CST method works very well when one does not have knowledge of the language and needs to create a stemmer fast. To create a stemmer with manual rules from scratch can take from several days to several weeks, depending on how skilled the computational linguist is and of course whether she/he has knowledge of the language. However, the evaluation could be improved with a larger test set than the 12 difficult unknown words. The training sets has quite good coverage of the language with around 40 000 unique words per language

For languages with regular inflection in both ends of the word (or even in the middle), such as German and Dutch, CST's lemmatiser is not the right choice and one either needs a much more complex tool to generate rules or one has to make the rules by hand.

Also, in the absence of a full form dictionary one may be forced to make rules by hand, but it is also feasible to write a small full form dictionary to train the lemmatiser to start with and to add new words and retrain the lemmatiser as the need arises.

At the other end of the scale: Extended word lists to train the lemmatiser do not necessarily give better lemmatisation rules than word lists that only contain words that belong to the domains that the lemmatiser has to handle, because extended word lists tend to have words belonging to several different lemmas which in many cases do not all belong to the domains of interest.

Acknowledgements

We would like to thank Johan Carlberger at Euroling AB for letting us use the Euroling stemmer and also to Georgios Ntais for letting us use the Greek stemmer.

References

Braasch, A. and S. Olsen. (2004). STO: A Danish Lexicon Resource - Ready for Applications. In: *Fourth International Conference on Language Resources and Evaluation, Proceedings, Vol. IV.* Lisbon, pp. 1079-1082.

Carlberger, J. H. Dalianis, M. Hassel and O. Knutsson. (2001). Improving Precision in Information Retrieval for Swedish using Stemming. In the Proceedings of NoDaLiDa-01 - 13th Nordic Conference on Computational Linguistics, May 21-22, Uppsala, Sweden.

Dalianis, H. (2000). SweSum - A Text Summarizer for Swedish, Technical report TRITA-NA-P0015, IPLab-174, NADA, KTH, October 2000

Hassel, M. (2004). Evaluation of automatic text summarization – a practical implementation. Licentiate thesis, Stockholm, NADA-KTH.

Jongejan, B and Dorte Haltrup. (2005). The CST Lemmatiser. Center for Sprogteknologi, University of Copenhagen version 2.7 (August, 23 2005) http://cst.dk/online/lemmatiser/cstlemma.pdf

Jongejan, B. (2006). CST's lemmatiser for dansk. In *Sprogteknologi i dansk perspektiv*, Reitzel, København, pp 370-390.

NCSR Demokritos Part-of-speech dictionary (2006) http://iit.demokritos.gr/ skel/

Ntais, G. (2006). Development of a Stemmer for the Greek Language, Master Thesis, Department of Computer and Systems Sciences, KTH-Stockholm university.

Petasis, G, V. Karkaletsis, D. Farmakiotou, I. Androutsopoulos, and C.D. Spyropoulos. SA Greek Morphological Lexicon and its Exploitation by Natural Language Processing Applications. T. Lecture Notes on Computer Science (LNCS), vol. 2563, "Advances in Informatics - Post-proceedings of the 8th Panhellenic Conference in Informatics". Vol. editors: Yannis Manolopoulos, Skevos Evripidou, Antonis Kakas, pp. 401-419, 2003.

SCARRIE (2006) Scandinavian Proofreading Tools http://ling.uib.no/~desmedt/ scarrie/

de Smedt, K., A. Liseth, M. Hassel and H. Dalianis (2005). How short is good? An evaluation of automatic summarization. In Holmboe, H. (ed.) *Nordisk Sprogteknologi 2004. Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004*, Museum Tusculanums Forlag, pp. 267-287.

STO lexicon (2006), http://www.cst.dk/sto/uk/

Stockholm Umeå corpus (2006), http://www.ling.su.se/staff/sofia/suc/suc.html