Prefix Recognition Experiments*

Jaroslava Hlaváčová, Michal Hrušecký
hlavacova@ufal.mff.cuni.cz, michal@hrusecky.net

Charles University in Prague, ÚFAL MFF

Abstract. The paper deals with automatic methods for prefix extraction and their comparison. We present experiments with Czech and English and compare the results with regard to the size and type (wordforms vs. lemmas) of input data.

1 Introduction

Prefixation and/or suffixation is one of the major means of word-formation in many languages. Prefixes and suffixes serve also for word inflection in flective languages.

Sets of affixes are usually well known for a given language, together with their main functions or meanings. The same thing may be stated about prefixed and suffixed words — they are usually included in dictionaries and there is no need to invent them again and again.

However, as languages evolve, new affixes may appear to create new words. One of the main sources of the new affixes are foreign languages.

Let us take as an example the prefix *re* closely connected with repeating events. There are quite a lot of words with this prefix in Czech corpora. We present several examples from the corpus SYN2010 proving that the prefix *re* may be attached to nouns, adjectives and verbs having always the meaning of an action, event. The prefix *re* adds the meaning of repetition.

renormalizace, renormalizovatelnost, renormalizovat
redesign, redesignovat, redesignovaný
remodelace, remodelování, remodelovat

On the other hand, the language itself often serves as a repository of new affixes — they are usually transformed roots. Taken strictly, they are not prefixes in the very linguistic sense. Pure linguists would probably call them rather stems and the resulting words compounds. However, they behave like prefixes — they can be attached to many existing words, changing their meaning, very often in the same or similar way.

Typical examples are names of colours usually attached to adjectives and adverbs, not so often to nouns. The following examples were taken again from

* This paper is a result of the projects supported by the grants number P406/2010/0875 and P202/10/1333 of the Grant Agency of the Czech republic (GAČR), and the grant MSM 0021620838 of the Czech Ministry of Education.

the corpus SYN2010. We extracted some of those that were not present in the morphological dictionary:

žlutobéžová, žlutozelená, žlutorukých, žlutodřevu, žlutohněda, hnědoočky, hnědopurpurovými, hnědobéžovou, hnědočervena, hnědopyský

For an automatic language processing it is very convenient to have a list of all affixes for a given language, as it helps to recognize "new"words that were not included into dictionaries (due to various reasons). The recognition consists not only of guessing morphological properties of the "new"words, but usually also their meanings. Thus, it is possible to use such lists for synthesis too.

At the first glance one could expect that for a given language, there exists a complete list of all its affixes, or at least that such a list is easy to collect. It is not the case however.

There are several statistical methods how to extract affixes automatically from a large amount of words. Their overview can be found for instance in [1]. We implemented some of them into a complex tool Affisix [2] and used it for experiments with several languages. Some of them are described in [3], [4] and [5]. In this contribution, we present selected results of prefix extraction, processed on three big Czech corpora, namely SYN2000, SYN2005 and SYN2010, and British National Corpus, each having 100 million tokens. We concentrate especially at differences among the methods and among the properties of the intput data.

2 Methods

In this section, we briefly introduce methods we used for our recent experiments. The detailed description may be found in [1]. All the methods need as an input a long list of words or lemmas. Every word is divided into two or more strings — segments — and investigated, if the segmentation has certain properties or not. The properties are expressed numerically, so it is easy to compare different segmentations and select those that are the best (for instance the highest). Segments that pass certain threshold are marked and we call them prefix-candidates.

2.1 Naive method

This method is based on two simple assumptions, but it produces quite interesting results. The first assumption: prefixes can be attached to many words. The second assumption: if a string is a prefix, we can remove it and the rest is still a meaningful word. The second assumption is not true for many prefixes, but for searching the "new"prefixes in the language, it works very well. These prefixes are usually simply glued to the beginning of existing words.

For every initial segment we count number of words starting with that segment and the number of words this segment can be attached to as a prefix.

$$n_p = |\{x; x \in S \ \& \ \exists y; x = p :: y\}| + |\{x; x \in S \ \& \ p :: x \in S\}|$$

where S is a set of all meaningful words and :: denotes concatenation. The greater the number n_p, the greater chance that the segment p is a real prefix.

2.2 Squares

A square is a quadruple $< a, b, c, d >$ of strings such that any combination of the first two strings with the second two strings forms a valid word in the language. Any of the strings can be empty. In this method, we look at every initial segment and count the number of squares it is in. Bigger the number of squares the segment is in, more probable the segment is an affix. In contrast to the previous naive method, the Squares method recognizes even prefixes in words where prefix is obligatory (for instance in the *jednoruký*).

2.3 Entropy methods

This method is based on the observation that the entropy between morphemes is usually higher than elsewhere. After a prefix, the entropy increases because the prefix string may be followed by many other strings, which is not the case inside the prefix, nor inside a subsequent stem. Thus, we can check the entropies after initial strings of input words, sort them and take those with the highest values as good candidates for prefixes.

Entropy is in general computed using the following formula:

$$H(a) = -\sum_{s \in C} p(s|a) \log_2 p(s|a)$$

where C is a set of possible continuations of the beginning string a.

We modified this approach by taking a difference between entropies of two adjacent letters instead of the entropy itself. We call this modification the difference entropy.

The list of prefix candidates according to the difference entropy gives usually better results, which means that among first N prefix candidates with the highest values of difference entropy we can find more real prefixes than among the same amount of candidates extracted by the (pure) entropy method.

See the results in the section 4.

2.4 Economy method

For the description of this method, we cite from [1]: "If a word is divided in two segments, one of them occurring in many other words, while the other occurs in only a few others, and if the first one belongs to a small set of frequent segments, while the other to a potentially infinite set of low occurring segments, then a morphological cut can be proposed between these segments."

Thus, the set of possible prefixes is obtained as a list of segments from the beginning of the words that have more possible continuations than the rest of the word has possible beginnings.

For a segmentation of a word, the economy index is calculated as a ratio of sizes of two sets: size of subset of all possible continuations of the initial segment divided by a size of subset of possible beginnings of the ending segment. For a detailed description of the subsets and the method itself see [1].

3 Data

We were mainly interested in method results using different sets of data we performed experiments with several different corpora. We used SYN2000, SYN2005 and SYN2010 [6] corpora for the Czech language and BNC [7] for English. For all the corpora, we extracted lists of wordforms with the frequency more than 10 and 50, respectively. For Czech, we made similar lists for lemmas as well. These sets of data were selected in order to test how the amount of data would influence the results of individual methods and whether it is better to use wordforms or lemmas for the prefix recognition. For English we used only word forms and we were mainly interested whether results of methods comparison in Czech would reflect in English as well.

Table 1 shows number of tokens (words or lemmas) entering the experiments with different corpora. Lists of tokens that we used for our experiments are named after their properties visible from the table 1: for instance syn2005-word-10 is the name of the list of words from the corpus SYN2005 with the frequency more than 10.

corpus	**word-50**	**word-10**	**lemma-50**	**lemma-10**
syn2000	114 283	305 677	56 302	123 018
syn2005	118 838	319 156	56 639	122 831
syn2010	116 266	311 143	54 836	117 961
bnc	48 074			

Table 1. Corpora comparison

4 Experiments

For each method, we sorted the numerical characteristics of the prefix-candidates. Then, we manually checked the best 100 and selected those that act as real prefixes.

Each method was assigned the score acquired by subtracting number of bad results from 100 (number of all results). Naturally, the score is a function of number of prefix-candidates — see examples of graphs in the following subsections.

We also present a table 2 showing several prefix candidates for entropy methods. For other methods, we only briefly describe their results.

Naive method Though this method is very simple and its assumptions may not be always fulfilled, it gave quite good results. The results do not differ much for individual corpora. In other words, the score defined above decreases for all the corpora roughly equally.

Squares method Surprisingly, this method performs considerably worse than naive method. It is also the slowest one of all the tested methods. Again, there is no considerable difference among the corpora.

Economy principle This method does not perform very well, but there is a small difference in favor of lemmatized corpora as compared with their unlemmatized counterparts.

Entropy methods For this method, we present two graphs showing the difference among the input data.

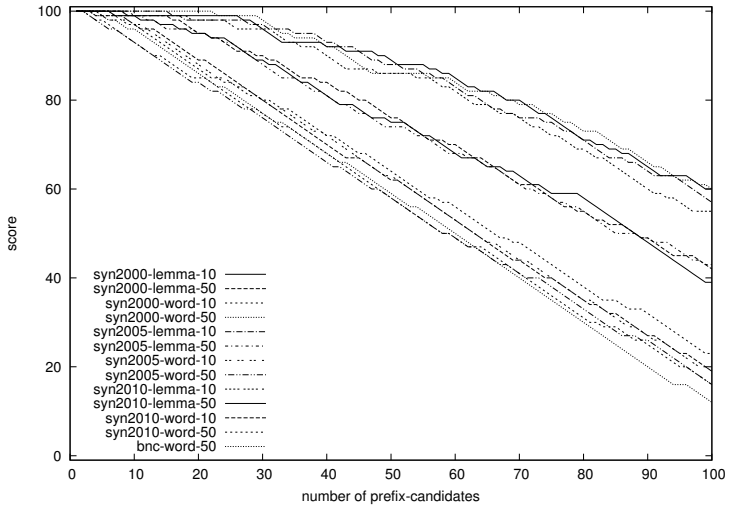

Fig. 1. Comparison of results of difference entropy method

The results of difference entropy method are presented in the graph 1. There are three main clusters of lines. The lowest one, showing the worst results, was obtained using Czech wordforms. Difference entropy tends to prefer longer prefixes. As cuts between a stem and a suffix are more obvious than between a prefix and a root, the difference entropy method got distracted many times by the suffixes.

On the contrary, English wordforms scored as good as the best Czech lemmas even though the English list is the smallest one. It can be explained by the not so rich English morphology — there are not many suffixes to distract the method.

The two upper clusters of the graph show, that amount of words matters. Both are results of lemmas, but the higher cluster belongs to the lists filtered with the frequency 10, while the middle with frequencies more than 50.

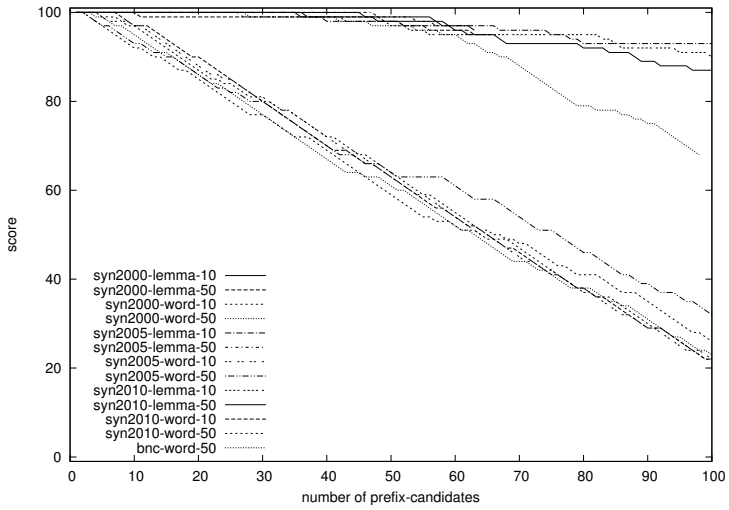

Fig. 2. Comparison of results of filtered difference entropy method

We tried additional filtering — using a constraint similar to the second assumption used for the naive method (see sec. 2.1). We demanded that among the words $x = p :: w$ with a prefix p, there must be at least 10 words, for which w is also a word. Unfortunately it turned out (see graph 2) that this constraint isn't limiting enough. Although it improved results, especially for smaller lists (these are now in the same cluster as their bigger counterparts), it still didn't improve the performance on wordforms. Neither did it improve results of BNC much (worst line from the upper cluster).

Filtered difference entropy			Entropy		
rank	score	prefix	rank	score	prefix
1.	2.5996971	**over**	1.	2.8503499	**non-**
2.	2.4349861	**micro**	2.	2.7454889	**inter**
3.	2.4228690	**water**	3.	2.6891198	*mar*
4.	2.4150519	**school**	4.	2.6787214	**back**
5.	2.3911490	**black**	5.	2.6780901	**pro**
6.	2.3825233	**super**	6.	2.6724777	**over**
7.	2.2052698	**stock**	7.	2.6367834	*car*
8.	2.0895109	**light**	8.	2.6299610	**under**
9.	2.0889339	**under**	9.	2.6107635	*cra*
10.	2.0280159	**self-**	10.	2.5970426	*man*

Table 2. Top ten prefix-candidates from entropy methods (prefixes are bold)

4.1 Comparison of all methods

The last experiment compares all the methods on the same list. Here we present only the graph for the results from the list syn2010-lemma-10 (Figure 3).

The experiments conducted on the other lists gave very similar results, the only visible difference being between lists of lemmas and wordforms. The former were always more successful, so it is better to use lemmatized data rather than wordforms. If a lemmatized corpus is not available, we recommend to use the naive approach to limit the search to prefixes only. The squares method and economy principle didn't perform well in our tests. On the other hand, entropy, and especially difference entropy performed well and were fast to compute. Surprisingly naive approach performed much better than more complicated methods.

All these facts can be derived from table 3 with the overall results from all the experiments.

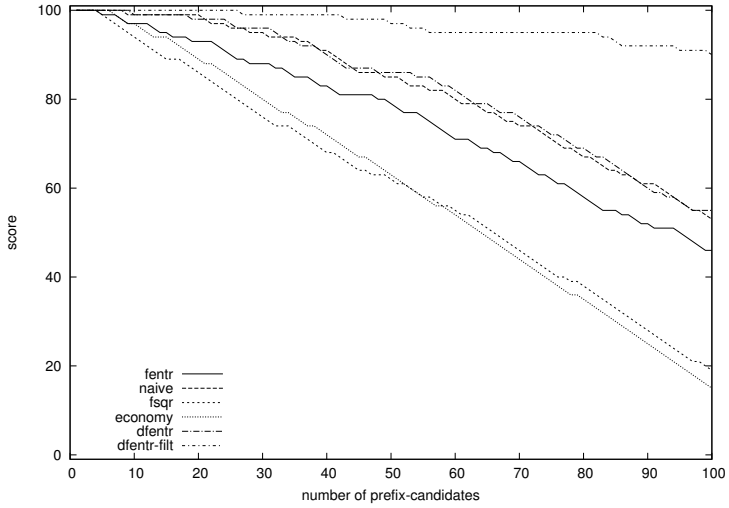

Fig. 3. Comparison of results on the list syn2010-lemma-10

5 Conclusions and Plans

Experiments conducted so far suggest that results of individual methods depend on the size of the input data (the corpus and its filtration) and language. For practical use (e.g. guessers of OOV[1] words or building morphematic databases) it would be important to select the appropriate method and the input corpus.

[1] out of vocabulary

	naive	squares	economy	dentr	dentr-filt
syn2000-lemma-10	66 %	20 %	26 %	78 %	98 %
syn2000-lemma-50	66 %	18 %	24 %	52 %	96 %
syn2000-word-10	82 %	26 %	38 %	28 %	28 %
syn2000-word-50	66 %	16 %	34 %	18 %	22 %
syn2005-lemma-10	74 %	22 %	28 %	76 %	96 %
syn2005-lemma-50	66 %	20 %	24 %	48 %	98 %
syn2005-word-10	80 %	28 %	38 %	26 %	26 %
syn2005-word-50	64 %	16 %	36 %	16 %	28 %
syn2010-lemma-10	70 %	24 %	26 %	72 %	94 %
syn2010-lemma-50	70 %	22 %	24 %	50 %	96 %
syn2010-word-10	78 %	22 %	40 %	24 %	26 %
syn2010-word-50	68 %	16 %	36 %	16 %	18 %
bnc-word-50	66 %	24 %	18 %	72 %	94 %

Table 3. Comparison of precision of all methods for top 50 prefix-candidates

Whenever it is possible, it is better to use lemmatized data rather than word-forms.

In the future we plan to continue with experiments and try to improve the results especially by some additional constraints or using combinations of the methods.

We also plan to try using these method for unsupervised stemming and compare the results against those of basic language-specific stemmers.

References

1. Urrea, A.M.: Automatic discovery of affixes by means of a corpus: A catalog of spanish affixes. Journal of Quantitative Linguistics **7** (2000) 97–114
2. Hrušecký, M.: Affisix. `http://affisix.sf.net`
3. Hrušecký, M., Hlaváčová, J.: Automatické rozpoznávání předpon a přípon s pomocí nástroje affisix. In Pardubská, D., ed.: Informačné technológie – Aplikácie a Teória, Zborník príspevkov prezentovaných na konferencii ITAT, Seňa, Slovakia, PONT s. r. o. (2010) 63–67
4. Bojar, O., Straňák, P., Zeman, D., Jain, G., Hrušecký, M., Richter, M., Hajič, J.: English-hindi translation – obtaining mediocre results with bad data and fancy models. In Sharma, D., Varma, V., Sangal, R., eds.: Proceedings of ICON 2009: 7th International Conference on Natural Language Processing, Hyderabad, India, NLP Association of India, Macmillan Publishers, India (2009) 316–321
5. Hlaváčová, J., Hrušecký, M.: "affisix" tool for prefix recognition. In Sojka, P., Horák, A., Kopeček, I., Pala, K., eds.: Lecture Notes in Artificial Intelligence, Proceedings of the 11th International Conference, TSD 2008. Volume 5246 of Lecture Notes in Computer Science., Berlin / Heidelberg, Springer (2008) 85–92
6. Ústav Českého národního korpusu FF UK: Český národní korpus - syn2000, syn2005, syn2010. `http://ucnk.ff.cuni.cz` (2000)
7. Oxford University Computing Services on behalf of the BNC Consortium: The british national corpus. `http://www.natcorp.ox.ac.uk` (2007)