Unsupervised Morphological Analysis of Central European Languages for Part-of-Speech Tagging

Daniel Hládek*, Ján Staš* , Jozef Juhár*

*Dept. of Electronics and Multimedia Communications, FEI,
Technical University of Košice, Park Komenského 13, 040 01 Košice, Slovakia
{daniel.hladek, jan.stas, jozef.juhar}@tuke.sk

Abstract

The ability to identify a stem and suffix of a word in an unsupervised way without additional knowledge about the target language is an important part of the language-independent systems for natural language processing. This paper proposes an unsupervised morphological analysis method for part-of-speech tagging. Unsupervised word segmentation can provide sufficiently precise results for any language just by inspecting its vocabulary. Experimental section will show effect of the unsupervised suffix identification as a feature for classification of unknown words for several Central European languages.

1. Introduction

Identification of semantics of a word or its grammatical function is a crucial part of many natural language processing systems for information retrieval, automatic translation, semantic parsing or part-of-speech tagging. In order to extract useful features, potentially helpful for uncovering meaning or structure of the sentence, it is necessary to utilize knowledge about morphology of the target language.

Morphology of a language in a natural language processing system can be expressed in two basic ways:

- **Supervised morphological analysis**: a set of exact rules for identification of morphology of a word, designed by an expert.
- **Unsupervised morphological analysis**: a vocabulary of the language is searched for repeating patterns and rules are identified by a machine learning algorithm.

The most precise approach for word morphology analysis is to use expert knowledge explicitly expressed as a rules in a specialized system. Classical example is Porter stemmer (Porter, 2006) which describes special formal grammar language to define suffix stripping rules. Advantage of this rule-based approach is that a result for covered cases will be very precise. On the other hand, design of such database is a very difficult and expensive process. It is unarguable that using the hand-crafted analysis of the language is the best option, but this option is often available just for languages with sufficient language resources, such as English, German or French.

Systems that should be able to handle a large number of languages (including under-resourced Central European languages) have to rely on an unsupervised analysis. The other possible approach is to examine lexicon and look for repeating patterns, analyze it statistically and extract rules that might be relevant. This approach is more general and is able to describe any given lexicon in a way independent on the lexicon contents or language rules. Some papers talking about this approach are (Paik and Parui, 2011; Saharia et al., 2013) that propose statistical stem identification and (Šnajder et al., 2008) present method for statistical suffix identification.

This paper focuses on the unsupervised morphological analysis and shows that a simple suffix identification can significantly improve precision of part-of-speech tagging if compared to the case when no additional morphological information is used. The testing of part-of-speech tagging that uses information from the morphological analysis is presented and evaluated in a set of experiments on multiple types of morphologically annotated corpora.

2. The Morphological Analysis Algorithm

Inflectional languages are characterized by a large number of possible word forms. Each language has a set of possible morphemes - set of sub-word units and many complex rules how to put morphemes together to create words according to the context and expressed information. A single concept can be written in many ways, depending on its grammatical function.

Usually, meaning of the word is encoded in the beginning part of a word and grammatical function of a word is expressed by the ending part of the word. For the purpose of this paper and for some simplification, the beginning part of a word is called a *stem* and the ending part, carrying information about grammatical function of a word is called a *suffix*. It is also assumed that each word can be divided into stem and suffix, even if the suffix had zero length. We do not take existence of prefixes into the account and take them as part of the stem.

Most of the UMA algorithms, like (Kirschenbaum, 2013) or (Creutz and Lagus, 2007) split words into several parts. This would be possible after some modifications are done, but in order to utilize results in part-of-speech tagging the focus is given on suffix identification and a word is split just in two parts as it is in (Paik and Parui, 2011; Saharia et al., 2013).

2.1. Signature List Construction

The first step of the algorithm is to find all potential stems and their signatures called *signature list*. Paper (Goldsmith, 2001) defines a signature as a ”statement of a morphological pattern” which can be seen as list of

suffixes commonly appearing with one or more stems. In this paper signature is a stem with a set of possible suffixes.

The process of signature list construction is displayed in details in Figure 1 as Python code, In this step, each word in the dictionary is analyzed for presence of potential stems and suffixes. Each possible stem and suffix of a word is identified, inserted to a dictionary and counted. The signature list is updated and suffix is assigned to its corresponding stem. Result of this step is a list of signatures for each stem found in the corpus and counts for each stem and suffix.

The next step of the algorithm is to evaluate items in the signature list and remove those signatures that are unfeasible according to a metric, calculated from information gathered in the previous step - counts and lengths of stems and corresponding suffixes.

It is possible to evaluate a signature according to this information in many ways (e.g. using entropy or minimum description length principle), but for the purpose of suffix identification it is sufficient to choose a simple metric based on a threshold.

To distinguish identified signatures, the following heuristics is used: *A stem is considered to be certain if all its suffixes are considered certain.* If there is an uncertain suffix, the whole signature is discarded.

This heuristics has one threshold parameter - *suffix threshold* that is a criterion of telling if the potential suffix occurred sufficient number of times in the training corpus.

If there is a suffix whose count is equal to the stem count, it means that this suffix was found only together with this stem. In this case there is no evidence that this suffix is certain, because it is not used with any other word. This fact has to be calculated in the heuristics and stem count have to be subtracted from the suffix count when comparing with the suffix threshold.

The whole process is depicted in Figure 2. Result is a list of possible suffixes and a signature list that can be used for morphological analysis of an unknown word in the analyzed language. This approach favors longer suffixes and shorter stems which is an advantage for pat-of-speech tagging task. In order to use this approach for stem identification for information retrieval a different metric for signature evaluation should be used.

```
# Iterate over each word
# in the vocabulary
for token,count in words.items():
    # Iterate over each
    # possible word division
    for i in range(2,len(token) -1 ):
        # Remember possible
        # stem and suffix
        stem = token[0:i]
        # Suffix is from i-th
        # character to the end
        suff = token[i:]
        stems[stem] += count
        suffs[suff] += count
        # Add suffix to stem signature
        signatures[stem].add(suff)
```

Figure 1: Python code to construct a signature list (a dictionary of sets)

```
# Iterate over each signature
for stem, sufset in signatures.items():
    isgood  = True
    # Check each suffix in signature
    for suf in sufset:
        # If there is a bad suffix,
        # signature is bad
        if thr >= suffs[suf] - stems[stem]:
            isgood = False
            break
     # Remember good signature
     # for further use
     if isgood:
        final_signatures[stem] = sufset
        for s in sufset:
            final_suffixes.add(s)
```

Figure 2: Python code to filter a signature list

The experiments will focus on the utilization of the suffix identification. Evaluation of the stem identification is left for the future research.

Word suffix can tell much about grammatical function of a word. Accuracy of the classification of contexts depends on handling of unknown words and correct identification of the suffix of an unknown word requires deep knowledge about target language morphology which is not always available. Correctly identified suffix of an unknown word can tell much about the function of the word. The resulting suffix list is used as helper for unknown word classification.

3. Morphological analysis using HMM model and suffix identification

HMM classification has been many times proven useful for POS tagging (Hajič et al., 2007; Halácsy et al., 2007). The experiments are conducted using our HMM based classifier Dagger (Hládek et al., 2012), previously used for annotation of a training corpus for language model for speech recognition (Rusko et al., 2014) and several other corpora (Ondáš et al., 2014; Hládek et al., 2014).

The classifier consists of these components:

- **Transition Model**: The algorithm takes two previous states (tags) into the account (second-order HMM).
- **Observation Model**: Gives probability of an observation (word) according to a given state (a part-of-speech tag).
- **Morphological Lexicon**: If a morphological lexicon is used, just valid word-tag combinations have to be searched. This feature increases both classification

Corpus Code	Sentences	Tokens	Vocabulary size	Annotation
CAK (Hladká et al., 2008)	31 707	652 131	79 105	manual
NKJP (Przepiórkowski et al., 2010)	85 663	1 216 695	143 867	manual
HUNWEB (Halácsy et al., 2004)	1 000 000	16 854 195	874 700	automatic
SKWEB (Majlis and Zabokrtskỳ, 2012)	1 000 000	18 101 436	753 498	automatic

Table 1: Evaluation corpora characteristics

speed and accuracy. The lexicon gives list of possible tags for each word seen in the training corpus.

- **Morphological Analyzer**: If an unknown word is present, the morphological analyzer must be used. The morphological analyzer is trained on the training corpus and tries to extract an useful feature from an unknown word - in this case it is a suffix. Suffix is identified by searching the set of possible suffixes for the longest matching suffix. Usual way is to take all suffixes occurring in the training corpus. The experiments below will compare this method to the method of suffix identification described above.

- **Additional Observation Model**: The additional observation model is similar to the basic observation model. It takes information from the morphological analyzer and estimates probability of state (tag) according to the identified suffix.

The model is calculated from the training corpus. The corpus is analyzed and counts of significant events are calculated. These counts are converted to probabilities using some modification of Maximum Likelihood method. After the model and its parameters are estimated, Viterbi algorithm is used to find the best matching sequence of tags to the presented sequence of words.

If a number of word forms of a language is low and so is number of morphological tags, the baseline HMM model can be fairly effective. On the other hand, if a training corpus is small and a number of possible word forms and morphological classes is high, state-transition model and state-observation model becomes sparse and as a consequence the Viterbi search assigns zero value to a perfectly possible state-observations. It is necessary to find a way to estimate probabilities of events that were not present during training.

In the case of the baseline POS tagging one observation corresponds to one word. It is easy to find all possible states for a word seen in the training set, but for unseen words all possible states must be examined. If there is no additional heuristics, it is hard to make classification for words unseen in the training phase. This is a big issue in the case of highly inflectional language, where one basic form of a word can have many inflections. Several additional techniques have been used to improve tagging accuracy, as it is presented in other papers:

1. Transition model smoothing using Knesser-Ney method, usual for trigram language models.

2. Observation model smoothing by Laplace Method.

3. First-capital word back-off and number string replacement as it is proposed in (Halácsy et al., 2007)

4. Beginning and end of sentence tokens, as it is in (Brants, 2000)

4. Experiments

In all Central European languages morphological form of a word strongly depends on a suffix of a word. It is assumed that in all these languages (Czech, Slovak, Hungarian and Polish) the identified suffix has a strong effect on the part-of-speech tagging precision.

The main purpose of the experiments is to prove usefulness of the UMA method of unknown word suffix identification for the task of part-of-speech tagging. Several corpus languages, annotation types and corpus sizes are used for evaluation. The proposed approach is compared to one of the common tagging systems and more approaches to suffix identification are used.

4.1. Evaluation Data

All used evaluation corpora are summarized in Table 1.

The first two evaluation corpora are manually annotated and rather small. Effect of smoothing techniques on the observation model and transition model should be more visible when compared to larger databases. There is also a higher probability of unknown word occurrence in the testing.

Czech language is represented by the Czech Academic Corpus (Hladká et al., 2008) (CAK). The other manually annotated corpus is the one million word sub-corpus of National Corpus of Polish (Przepiórkowski et al., 2010) (NKJP), available from site[1].

On the other hand, to mitigate effect of HMM state and observation probability model smoothing, we have chosen much larger web-based corpora of Hungarian and Slovak. These corpora are automatically annotated, because manual annotation of such corpus would be impractical or impossible. It is possible that these text are not totally clean and contain some words from other languages, the suffix identification algorithm should be able to deal with this kind of words. As the vocabulary extracted from a larger corpus is also larger, the suffix identification should be also more precise.

Hungarian data were obtained from morphologically analyzed part of Hungarian Web Corpus (Halácsy et al., 2004), downloaded from ftp[2] (HUNWEB). The Slovak

[1] http://clip.ipipan.waw.pl/LRT?action=AttachFile&do=get&target=NKJP-PodkorpusMilionowy-1.0.tgz

[2] ftp://ftp.mokk.bme.hu/Language/Hungarian/Corp/Webcorp/ana/xaa.tagged.gz

	# OOV	tokens	% OOV	vocab
CAK	4 648	65 455	7,10 %	19 113
NKJP	8 652	120 764	7,16 %	32 666
HUNWEB	52 043	1 684 857	3,08 %	215 144
SKWEB	41 056	1 810 355	2,26 %	212 547

Table 2: Testing set characteritics

	TreeTagger	No back-off	All	UMA
CAK	13,10	14,94	12,65	9,46
NKJP	15,63	16,55	13,94	11,83
HUNWEB	2,55	4,22	2.94	1,97
SKWEB	10,30	4,78	4,63	4,47

Table 3: Classification error rates (in %)

data (SKWEB) were taken from web2corpus project (Majlis and Zabokrtskỳ, 2012), Slovak part of the Aranea effort (Benko, 2014) and uses tag set from the Slovak National Corpus (Garabík and Šimková, 2012).

4.2. Evaluation Methodology

In the first step, the corpus is divided into a training and a testing part, where each tenth sentence goes to the testing set and the rest goes to training. The resulting testing sets are analyzed for vocabulary and compared to the training sets to count out-of-vocabulary words (OOV). Short summary of the testing sets is in the Table 2.

The training part is used to automatically create a set of rules for suffix identification as it was described above and for estimation of HMM model.

In order to make these results comparable, a one of the most commonly used taggers was selected - HMM model utilizing ID3 regression tree TreeTagger (Schmid, 1995) [3] and trained on the same training data.

TreeTagger training is run with the following command (as it is recommended by the authors):

```
train-tree-tagger -cl 2 -dtg 0.50
  -sw 1.00 -ecw 0.15 -atg 1.20
```

Numbers in the training corpus of the TreeTagger were replaced by a replacement string, in a similar way than in our approach so all numbers are seen as the same observation.

The trained model is evaluated on the testing set by calculating classification error rate (fraction of the bad classification results over all results when compared to the reference tagging).

In order to evaluate effect of suffix identification on the accuracy of the part-of-speech tagging, the proposed system was run in three configurations.

In the first configuration the back-off scheme was turned off so the HMM classifier was unable to deal with unknown words. Each OOV word was tagged with "unknown" class.

The second configuration used approach that is common in other part-of-speech tagging systems - the algorithm counted all suffixes occurring in the training in a similar way as it expressed in Python code in Figure 1. When compared to the presented algorithm, the signature list and signature list criterion was not taken into the account.

The third configuration was the presented algorithm with UMA-based suffix identification, as it was presented above. The training set was used for UMA analysis and all suffixes found with good stems were used. The longest matching suffix was used as a feature for classification of unknown tokens.

[3] http://www.cis.uni-muenchen.de/šchmid/tools/TreeTagger/

4.3. Experimental Results

Results of all experiments are summarized in Table 3. It can be seen that the proposed suffix identification for unknown words significantly reduced classification error rate when compared to other testing runs with our and reference classification systems.

In all cases, the presented classifier reached lower error rates than the reference TreeTagger. Authors are aware that it is possible to reach better precision with some other part-of-speech tagging systems. But these experiments focus only on evaluation of unknown words classification. Better precision of the presented systems could be reached by taking larger context into the account or by different smoothing methods in a similar way than in (Hajič et al., 2007; Halácsy et al., 2007).

5. Conclusion

The paper presented a method for unknown word classification based on a suffix identification. A method of unsupervised morphological analysis was proposed as a promising way of improvement of part-of-speech tagging precission, especially for unknown words.

After some further modifications are designed and evaluated, it would be possible to utilize the proposed method for more difficult task than suffix identification. It would be interesting to try more realistic methods for signature list evaluation, taking counts and lengths of identified items into the account. The proposed approach can be also used for more detailed morphological segmentation by repeated application on resulting stems in signature list to form a tree of possible word splits.

The paper (Hammarström and Borin, 2011) says: *"Most ULM (unsupervised learning of morphlology) approaches reported in the literature are small proof-of-concept experiments..."* and *"It can be seen that ULM systems are mature enough to enhance IR (information retrieval), but so far, ULM systems are not close to full accuracy on the gold standard.."*. In general, we can agree with these statements. On the other hand, in recent years there has been a great rise in multi-lingual natural language processing systems and demmand to process languages with insufficient infrastructure. This is the area where unsupervised methods can find their use.

Authors hope that these results will have application in other natural language processing tasks, such as information retrieval or language modelling which will be shown in the future research.

Acknowledgement

The research presented in this paper was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the project VEGA 1/0075/15 (50%) and the Research & Development Operational Programme funded by the ERDF under the project implementation University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, ITMS: 26220220182 (50%).

6. References

Benko, Vladimír, 2014. Aranea: Yet another family of (comparable) web corpora. In *Text, Speech and Dialogue*. Springer.

Brants, Thorsten, 2000. TnT: a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics.

Creutz, Mathias and Krista Lagus, 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1).

Garabík, Radovan and Mária Šimková, 2012. Slovak morphosyntactic tagset. *Journal of Language Modelling*, 1:41–63.

Goldsmith, John, 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Hajič, Jan, Jan Votrubec, Pavel Krbec, Pavel Květoň, et al., 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*. Association for Computational Linguistics.

Halácsy, Péter, András Kornai, Nemeth Laszlo, Rung Andras, István Szakadát, and Tron Viktor, 2004. Creating open language resources for Hungarian. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004)*.

Halácsy, Péter, András Kornai, and Csaba Oravecz, 2007. HunPos: an open source trigram tagger. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics.

Hammarström, Harald and Lars Borin, 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Hládek, Daniel, Ján Staš, and Jozef Juhár, 2012. Dagger: The Slovak morphological classifier. In *ELMAR, 2012 Proceedings*. IEEE.

Hládek, Daniel, Ján Staš, and Jozef Juhár, 2014. Slovak web discussion corpus. In *Advances in Natural Language Processing*. Springer International Publishing, pages 463–469.

Hladká, Barbora, Jan Hajič, Jirka Hana, Jaroslava Hlaváčová, Jiří Mírovskỳ, and Jan Raab, 2008. The Czech academic corpus 2.0 guide. *The Prague Bulletin of Mathematical Linguistics*, 89:41–96.

Kirschenbaum, Amit, 2013. Unsupervised segmentation for different types of morphological processes using multiple sequence alignment. In Adrian-Horia Dediu, Carlos Martn-Vide, Ruslan Mitkov, and Bianca Truthe (eds.), *Statistical Language and Speech Processing*, volume 7978 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pages 152–163.

Majlis, Martin and Zdenek Zabokrtskỳ, 2012. Language richness of the web. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC2012)*.

Ondáš, Stanislav, Daniel Hládek, and Jozef Juhár, 2014. Semantic roles labeling system for slovak sentences. In *Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on*. IEEE.

Paik, Jiaul H. and Swapan K. Parui, 2011. A fast corpus-based stemmer. *ACM Transactions on Asian Language Information Processing*, 10(2).

Porter, Martin F, 2006. An algorithm for suffix stripping. *Program*, 40(3):211–218.

Przepiórkowski, Adam, Rafal L Górski, Marek Lazinski, and Piotr Pezik, 2010. Recent developments in the national corpus of Polish. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010)*.

Rusko, Milan, Jozef Juhár, Marián Trnka, Ján Staš, Sakhia Darjaa, Daniel Hládek, Róbert Sabo, Matúš Pleva, Marián Ritomskỳ, and Martin Lojka, 2014. Slovak automatic dictation system for judicial domain. In *Human Language Technology Challenges for Computer Science and Linguistics*. Springer International Publishing, pages 16–27.

Saharia, Navanath, Kishori M. Konwar, Utpal Sharma, and Jugal K. Kalita, 2013. An improved stemming approach using HMM for a highly inflectional language. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'13. Berlin, Heidelberg: Springer-Verlag.

Schmid, Helmut, 1995. Treetagger: a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43:28.

Šnajder, J., B. Dalbelo Bašić, and M. Tadić, 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Inf. Process. Manage.*, 44(5):1720–1731.