DMoG: A Data-Based Morphological Guesser

Vojtěch Kovář[1,2], Pavel Rychlý[1,2]

[1] Natural Language Processing Centre
Faculty of Informatics, Masaryk University, Brno, Czechia
[2] Lexical Computing
Brno, Czechia

{xkovar3,pary}@fi.muni.cz
name.surname@sketchengine.eu

Abstract. We present a novel corpus-based approach to lemmatization of unknown words. The tool learns affix patterns from annotated data, and based on these patterns, it predicts other word forms that should be present in the corpus. A lemma candidate then comes from the pattern whose predictions are really found in the corpus.
We present a prototype implementation and an initial evaluation on Czech, which shows promising results.

Keywords: Lemmatization · Morphological guesser · Morphological analysis · Morphological guessing

1 Introduction

Lemmatization of natural languages is the process of assigning a lemma (base form) to each word in the input text. Typically, it is solved by a look-up in a large database of all possible word-lemma or word-tag-lemma combinations.

However, there are always words missing in the database, so-called out-of-vocabulary (OOV) words: rare words, neologisms etc. In other cases, namely in low-resourced languages, there is no large word-lemma database available. In these cases, a morphological guesser is needed which suggests lemmas and/or parts-of-speech for OOV words.

In this paper we present a novel approach to the problem of morphological guessing based on checking guesser's predictions against corpus data. We also present a prototype implementation which is so far only limited to guessing lemmas (not tags) based on suffixes – on the other hand, the tool is extremely simple (less than 120 lines of Python code) and extensions are straightforward. Also, for some languages (including Czech, our testing language), this may already be useful and sufficient.

2 Related work and its drawbacks

Existing solutions which include [1] or [2] rely on longest affix matching between a particular OOV word and patterns learned from an available database.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 135–138, 2021.

In certain contexts, this leads to wrong lemma candidates that sound funny to native speakers, such as the following output for a few Czech OOV words from [1]:

```
buřtguláš     buřtgulat     k5eAaImIp2nS,k5eAaPmIp2nS
knedlo        knednout      k5eAaPmAgNnS
flash         flasha        k1gFnPc2
groupe        groupat       k5eAaPmIp3nS
nVidia        nVidium       k1gNnSc2
komorbiditou  komorbidity   k2eAgFnSc7d1
```

In all cases except the last one, the lemma should be the same as the word form and the lemma proposed by the tool does not exist in Czech at all. The last case is a noun in instrumental (*comorbidity*) and its lemma should be *komorbidita*.

3 Corpus-based approach

In this paper we present a different approach. Our tool learns morphological patterns from available data as well, but the patterns represent declination schemata as a whole; and instead of matching an isolated OOV word and searching for longest affix match, it generates word forms that the particular pattern predicts (including the candidate lemma) and checks how many of them occur in the corpus.

For example, if *buřtguláš* has a lemma *buřtgulat* then it corresponds to a pattern which also predicts existence of the following word forms:

```
buřtgulat  buřtgulal
buřtgulám  buřtguláme
buřtguláš  buřtguláte
buřtgulá   buřtgulají
...
```

If we check this list against the corpus, we find out that the only existing word form is *buřtguláš* – so this is not a really good candidate, although the suffix indicates it might be a verb.

On the other hand, if it is a noun with lemma *buřtguláš*, then it corresponds to another pattern which predicts the existence of the following forms:

```
buřtguláš
buřtguláše
buřtguláši
buřtgulášem
...
```

Let's say 3 of these forms really occurred in the corpus (or corpus word list, respectively). Then we say this pattern is more suitable for this OOV word than the verb pattern, even if the common suffix is short or non-existent.

3.1 Patterns

A *pattern* in our understanding is a set of suffix pairs (*s1, s2*) where *s1* needs to be stripped from a word form and then *s2* needs to be added, to get a lemma. For example, the pattern for the verb schema mentioned above would contain

```
(-ám, -at)
(-áš, -at)
(-á, -at)
(-ál, -at)
(-ám, -at)
(-l, -t)
(-áme, -at)
(-áte, -at)
(-jí, -t)
...
```

This would be learned from many Czech verbs like *dělat, hledat* etc.

4 Implementation

Our prototype implementation consists of two Python scripts, `train.py` and `guess.py`. The first one reads a list of correct word-lemma pairs (obtained from manual annotation, morphological database, or a high-quality corpus) and saves the learned patterns into a so-called *model* (which is, however, just a set of patterns like the one above).

The `guess.py` script reads the model, together with an input word list generated from a corpus (i.e. not just isolated OOV words, but the complete corpus word list). For each of the words in the list, it tries to match the word suffixes, for each pattern from the model. If there is a suffix match, the tool generates all the potential word forms predicted by the pattern, and checks how many of them are there in the word list. The pattern who predicts the most existing lemmas wins the game, and its predicted lemma is returned as the result for the particular word.

5 Evaluation

As a preliminary evaluation, we trained the model on the word-lemma list of the manually disambiguated DESAM corpus [3], including only word-lemma pairs with frequency at least 5.

As testing data, we took the 40 most frequent OOV words from the csTenTen17 web corpus [4]. The results of our tool were as follows:

- correct lemmas: 36
- incorrect lemmas: 4
- accuracy: 90%

We have compared this result with the tool introduced in [1], on the same 40 words. Its results were as follows:

- correct lemmas: 26
- incorrect lemmas: 14
- accuracy: 65%

Although we admit that the testing set is very small and that it contained some noise (like a few frequent English terms used within Czech texts), the difference seems to be quite significant. Based on this result, we believe our DMoG prototype is worth further development, as well as a deeper research of the method itself.

6 Conclusions

We have introduced a new method for guessing lemmas for out-of-vocabulary words. We have explained the method and presented a prototype implementation, the DMoG tool. Although the current implementation only deals with lemmas and suffixes (and not prefixes, infixes and tags), it can be extended in a straightforward way, which is also the main goal of the future work.

Although the work itself, as well as the evaluation, are so far only preliminary, the tool shows promising results.

Acknowledgements. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

1. Šmerk, P.: Towards Czech morphological guesser. In Petr Sojka, A.H., ed.: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, Brno, Masarykova univerzita (2008) 1–4
2. Jongejan, B., Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. (2009) 145–153
3. Pala, K., Rychlý, P., Smrž, P.: DESAM — annotated corpus for Czech. In: Proceedings of SOFSEM'97, Berlin, Springer (1997) 523–530
4. Suchomel, V.: csTenTen17, a recent Czech web corpus. In Aleš Horák, P.R., Rambousek, A., eds.: Proceedings of the Twelfth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2018, Brno, Tribun EU (2018) 111–123