Named Entity Recognition for Highly Inflectional Languages: Effects of Various Lemmatization and Stemming Approaches

Michal Konkol and Miloslav Konopík

Department of Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia
Univerzitní 8, 306 14 Plzeň, Czech Republic
konkol, konopik@kiv.zcu.cz

Abstract. In this paper, we study the effects of various lemmatization and stemming approaches on the named entity recognition (NER) task for Czech, a highly inflectional language. Lemmatizers are seen as a necessary component for Czech NER systems and they were used in all published papers about Czech NER so far. Thus, it has an utmost importance to explore their benefits, limits and differences between simple and complex methods. Our experiments are evaluated on the standard Czech Named Entity Corpus 1.1 as well as the newly created 2.0 version.

Keywords: Named Entity Recognition, Lemmatization, Stemming

1 Introduction

Named entity recognition (NER) is a standard natural language processing (NLP) task. A NER system detects phrases with interesting meaning and classifies them into predefined groups – typically persons, organizations, locations, etc. NER is used as a preprocessing for many other NLP tasks, including question answering [1], machine translation [2], or summarization [3]. Together with named entity disambiguation, it can be used to link entities from text to a knowledge base.

NER is mostly studied on English. In this paper, we study the NER task on Czech – a highly inflectional Slavic language. All the published results on Czech NER use lemmatization to reduce the overwhelming amount of different word forms.

In this paper, we explore various lemmatization and stemming approaches. These approaches range from very simple methods to very complex hybrid (combining rule-based and machine learning techniques) systems. We also experiment with a morphological analyzer, which does not disambiguate the lemmas as full lemmatizer.

Our main goal is to measure the difference between the simplest and the most complex systems, i.e. to answer the question: Is it worth using a complicated, computationally expensive lemmatization (resp. stemming) system?

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 267–274, 2014.
This is preprint prepared by Proceedings editor for Springer International Publishing Switzerland.

2 Related Work

The first NER system for Czech was published together with the CNEC 1.0 corpus [4]. It was based on Decision Trees and achieved 68% F-measure. This system was outperformed by SVM-based NER system [5] with 71% F-measure. The following two systems were based on Maximum Entropy [6] (72.94% F-measure) and Conditional Random Fields [7] (58% F-measure). Both systems were evaluated by different methods and were not directly comparable to the previous systems and to each other. Another system based on Conditional Random Fields was published in [8] together with comparison of all previous systems. It achieved 74.08% F-measure on the CoNLL version of the CNEC 1.0 corpus and outperformed all of the previous systems. A system based on Maximum Entropy Markov Models was published at the same time [9]. It was evaluated on the original version of CNEC 1.0 corpus and achieved 82.82% F-measure. The last two systems form the current state-of-the-art for Czech NER. The former for the CoNLL versions of the CNEC corpora and the latter for the original (hierarchical) versions of the CNEC corpora.

All the published systems seem to use the PDT 2.0 corpus [10] for training of their lemmatizers. The first two systems [4,5] use directly the annotations that can be found in the CNEC corpus. In [6,8,7], they use a standard HMM tagger trained on PDT 2.0. The last system [9] uses a lemmatizer based on average perceptron sequence labeling [11].

3 Lemmatizers and Stemmers

This section briefly describes the lemmatization and stemming approaches we use in this paper. It should provide a basic idea about the quality and complexity of each method.

3.1 OpenOffice based lemmatizer

This lemmatizer (proposed by authors of this paper) is inspired by the approach from [12]. It uses the dictionaries and rules created for error correction in the OpenOffice. These resources are meant to be used in a generative process – the words forms are created from the dictionary using the rules.

In our approach, we try to do an inverse operation. This operation is ambiguous and in many cases there are more possible lemmas. We simply choose the first one. The necessary OpenOffice resources are freely available for many languages, thus this approach can be used for many languages.

3.2 HPS Stemmer

The High Precision Stemmer[1] (HPS) is an unsupervised stemmer. It works in two phases. In the first phase, lexically similar words are clustered using MMI clustering [13]. The word similarity is based on the longest common prefix. The output of this phase are clusters which share a common prefix and have the minimal MMI loss. The method assumes that the common prefix is a stem and the rest is a suffix.

[1] http://liks.fav.zcu.cz/HPS/

The second phase consists of training of a Maximum Entropy classifier. The clusters created in the first phase are used as the training data for the classifier. The classifier uses general features of the word to decide where to split the word into a stem and a suffix.

3.3 HMM tagger

The HMM (Hidden Markov Model) tagger represents a standard (pure) statistical approach to lemmatization [14]. Transition probabilities in HMM are estimated using 3-gram Kneser-Ney smoothing [15]. This approach can be easily reproduced using common machine learning libraries. It is trained on the PDT 2.0 data [16].

3.4 PDT 2.0 Lemmatizer

The PDT 2.0 lemmatizer [16] uses the most complex approach. The system as a whole is a hybrid system[2]. It is based on two main components – a morphological analyser and a tagger. The morphological analyser is rule-based. It is based upon a dictionary with 350,000 entries and derivation rules. The tagger is statistical (feature-based). The system also contains a statistical guesser for out-of-dictionary words.

3.5 Majka

Majka [17] is rule-based morphological analyser. It provides all possible word forms for a given word. It is not a tagger as it does not disambiguate the proposed lemmas and tags. The authors of Majka are currently working on the disambiguation tool, but it has not been released as a usable library yet.

For our use, we always select the most frequent lemma-tag pair. This is definitely not an optimal solution, but it will be interesting to compare it to the other lemmatization and stemming approaches. Keep in mind, that in this way, we do not use the full potential of Majka and the results would be probably better using some state-of-the-art tagging approach.

3.6 MorphoDiTa

MorphoDiTa[3] [18] is a state-of-the-art tool for morphological analysis, which is based on the averaged perceptron algorithm [11]. The algorithm is derived from standard HMMs, but the transition and output scores are given by a large set of binary features and their weights.

[2] According to `http://ufal.mff.cuni.cz/pdt2.0/browse/doc/tools/machine-annotation/`

[3] `https://ufal.mff.cuni.cz/morphodita`

4 NER System

Our NER system is based on Conditional Random Fields (CRF) [19], which are considered as the best method for NER by many authors. We use Brainy [20] implementation of CRF.

All features are used in a window $-2, \ldots, +2$. We use the following feature set for our experiments:

Word – Each word that appears at least twice is used as a feature.

Lemma – The lemmatization approaches are described in section 3. A lemma has to appear at least twice to be used as a feature.

Affixes – We use both prefixes and suffixes of the actual word. Their length ranges from 2 to 4. The affixes are based on lemmas and have to appear at least 5 times.

Bag of lemmas – Identical to bag of words, but uses lemmas instead of words. The lemma has to appear at least twice to be used as a feature.

Bi-grams – Bi-grams of lemmas have to appear at least twice to be used. Higher level n-grams did not improve the results, probably due to the size of the corpora.

Orthographic features – Standard orthographic features. Including *firstLetterUpper; allUpper; mixedCaps; contains ., ', –, &; upperWithDot; various number formats; acronym*

Orthographic patterns – Orthographic pattern [21] rewrites the word to a different representation, where every lower case letter is rewritten to a, upper case letter to A, number to 1 and symbol to -.

Orthographic word pattern – A compressed orthographic pattern is created for each word in the window. The combination of these patterns forms the orthographic word patter feature. Each combination has to appear at least five times.

Gazetteers – We use multiple gazetteers. They are acquired from publicly available sources such as list of cities from the Czech Ministry of Regional Devel- opment

5 Corpora

In this paper we use the Czech Named Entity Corpus (CNEC) versions 1.1 and 2.0 [4]. We use the older 1.1 version for smooth transition to the 2.0 version, i.e. we can directly compare the results on these corpora and use only the 2.0 version in the future work. We use the CoNLL versions [8] of both corpora.

The 1.1 version contains 5,868 sentences (149,538 tokens), the 2.0 version 8,993 sentences (199,216 tokens). The CNEC 1.1 has a higher density of named entities than common texts. This problem was addressed by adding 3,000 sentences with only a few entities in the new version. The CNEC 2.0 was also extended by some sentences that contains addresses and emails. Both CoNLL versions use seven types of named entities – time (T), geography (G), person (P), address (A), media (M), institution (I) and other (O).

6 Experiments

Our experiments are relatively straightforward. We train a NER model on the training data using each lemmatization (or stemming) approach. Then, we evaluate these models on the validation and test data. As we do not use the validation data for choosing any parameters of the system, they have the same information value as the test data. For all experiments, we use the feature set described in section 4. It consists of frequently used features with default parameters and should work very well in all our experiments.

We use two separate metrics – the strict CoNLL evaluation and the lenient evaluation used in GATE[4]. Both are based on the precision, recall and F-measure. The strict metric considers the marked entity as correct only if it agrees with gold data in both span and type. The lenient metric is a supplement to the strict metric and covers the cases, where the system guesses the correct type, but the span is partially wrong (e.g. two words of three are marked).

The results of our experiments are shown in Tables 1-2. An important finding is that even the simplest methods improve the results, even though the word-based baseline is much stronger on the CNEC 2.0.

The methods based on the standard tagging approaches significantly outperform the methods based on approximative techniques (OO lemmatizer, HPS). This also holds for our approach of using Majka, which in fact, do not use disambiguation but only a morphological analysis. The HMM tagger, MorphoDiTa and PDT 2.0 lemmatizer outperforms our Majka-based approach, but we believe that some combination of our HMM approach and Majka would perform better than both individual methods.

The best results were achieved using the PDT 2.0 lemmatizer with a slight edge over MorphoDiTa. The difference is probably caused by the OOV guesser as entities are more often OOV words than common words. Both significantly outperformed our basic HMM approach.

Generally, the more complex the method is, the better the result is achieved in our tests. This trend is much more obvious than we expected at the beginning.

7 Conclusion and Future Work

We have tested six different lemmatization and stemming approaches in the NER task. They range from very simple to complex state-of-the-art systems. We use a standard setting for Czech NER – the CNEC corpus and CoNLL evaluation.

The results show that supervised lemmatization approaches significantly outperform both the simple rule-based system and the unsupervised stemmer. All lemmatization and stemming approaches were better than the word-based baseline. The best results were achieved using the PDT 2.0 lemmatizer – 74.23% F-measure on the CNEC 1.1 corpus and 74.37% F-measure on the CNEC 2.0 (both in CoNLL format). These results outperform the current state-of-the-art on the CNEC 1.1 corpus and define it for the newly created CNEC 2.0 (as there are no previous results on this corpus).

[4] `http://gate.ac.uk/sale/tao/splitch10.html#x14-26900010.2`

Table 1. Results for the CNEC 1.1.

		Strict			Lenient		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Validation set	Baseline	69.67	66.18	67.87	76.65	72.41	74.47
	OO lemmatizer	76.16	72.74	74.41	82.93	78.79	80.81
	HPS stemmer	76.02	72.53	74.23	82.91	78.56	80.68
	HMM tagger	77.57	74.67	76.09	83.90	80.29	82.05
	PDT 2.0 lemmatizer	78.20	75.52	76.84	84.62	81.29	82.92
	Majka	77.03	73.65	75.30	83.51	79.43	81.42
	MorphoDiTa	78.57	75.63	**77.07**	84.58	80.99	82.79
Test set	Baseline	69.69	66.47	68.05	76.68	72.79	74.69
	OO lemmatizer	72.87	68.36	70.55	80.03	74.80	77.33
	HPS stemmer	73.13	69.10	71.05	80.62	75.70	78.09
	HMM tagger	75.40	71.09	73.18	81.75	76.79	79.19
	PDT 2.0 lemmatizer	76.16	72.40	**74.23**	82.06	77.73	79.84
	Majka	74.58	70.19	72.32	81.55	76.40	78.89
	MorphoDiTa	75.76	71.67	73.66	82.36	77.51	79.86

Table 2. Results for the CNEC 2.0.

		Strict			Lenient		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Validation set	Baseline	74.70	70.47	72.52	81.27	76.26	78.69
	OO lemmatizer	76.91	72.26	74.51	83.36	77.89	80.53
	HPS stemmer	75.66	72.06	73.82	82.33	77.83	80.02
	HMM tagger	77.42	74.34	75.85	83.93	80.12	81.98
	PDT 2.0 lemmatizer	78.06	75.24	**76.62**	84.91	81.39	83.11
	Majka	77.56	73.40	75.42	84.27	79.28	81.70
	MorphoDiTa	78.24	74.99	76.58	84.41	80.37	82.34
Test set	Baseline	73.40	69.14	71.20	80.39	75.35	77.79
	OO lemmatizer	73.99	69.34	71.59	81.33	75.88	78.51
	HPS stemmer	73.87	69.58	71.66	81.38	76.25	78.73
	HMM tagger	75.27	70.91	73.03	82.63	77.42	79.94
	PDT 2.0 lemmatizer	76.41	72.43	**74.37**	82.58	78.01	80.23
	Majka	74.99	70.86	72.87	82.27	77.36	79.74
	MorphoDiTa	76.39	72.33	74.31	82.87	78.17	80.45

Acknowledgements

This work was supported by grant no. SGS-2013-029 Advanced computing and information systems, by the European Regional Development Fund (ERDF). Access to the MetaCentrum computing facilities provided under the program “Projects of Large Infrastructure for Research, Development, and Innovations” LM2010005, funded by the Ministry of Education, Youth, and Sports of the Czech Republic, is highly appreciated.

References

1. Mollá, D., Van Zaanen, M., Smith, D.: Named entity recognition for question answering. In: In Lawrence Cavedon and Ingrid Zukerman, editors, Proceedings of the 2006 Australasian Language Technology Workshop. (2006) 51–58
2. Babych, B., Hartley, A.: Improving machine translation quality with automatic named entity recognition. In: Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT. EAMT ’03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 1–8
3. Kabadjov, M., Steinberger, J., Steinberger, R.: Multilingual statistical news summarization. In Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R., eds.: Multilingual Information Extraction and Summarization. Volume 2013 of Theory and Applications of Natural Language Processing. Springer Berlin Heidelberg (2013) 229–252
4. Ševčíková, M., Žabokrtsky, Z., Krůza, O.: Named entities in Czech: annotating data and developing NE tagger. In: Proceedings of the 10th international conference on Text, speech and dialogue. TSD’07, Berlin, Heidelberg, Springer-Verlag (2007) 188–195
5. Kravalová, J., Žabokrtský, Z.: Czech named entity corpus and SVM-based recognizer. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration. NEWS ’09, Stroudsburg, PA, USA, Association for Computational Linguistics (2009) 194–201
6. Konkol, M., Konopík, M.: Maximum entropy named entity recognition for Czech language. In: Proceedings of the 14th international conference on Text, speech and dialogue. TSD’11, Berlin, Heidelberg, Springer-Verlag (2011) 203–210
7. Král, P.: Features for Named Entity Recognition in Czech Language. In: KEOD. (2011) 437–441
8. Konkol, M., Konopík, M.: Crf-based czech named entity recognizer and consolidation of czech ner research. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue. Volume 8082 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2013) 153–160
9. Straková, J., Straka, M., Hajič, J.: A new state-of-the-art czech named entity recognizer. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue: 16th International Conference, TSD 2013. Proceedings. Volume 8082 of Lecture Notes in Computer Science., Berlin / Heidelberg, Západočeská univerzita v Plzni, Springer Verlag (2013) 68–75
10. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Karolinum, Charles University Press, Prague, Czech Republic (2004)
11. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. EMNLP ’02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 1–8
12. Kanis, J., Skorkovská, L.: Comparison of different lemmatization approaches through the means of information retrieval performance. Lecture Notes in Artificial Intelligence **2010** (2010) 93–100

13. Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Comput. Linguist. **18** (1992) 467–479
14. Kupiec, J.: Robust part-of-speech tagging using a hidden markov model. Computer Speech & Language **6** (1992) 225 – 242
15. Chen, S.F., Goodman, J.T.: An empirical study of smoothing techniques for language modeling. Technical report, Computer Science Group, Harvard University (1998)
16. Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., Ševčíková Razímová, M.: Prague dependency treebank 2.0 (PDT 2.0) (2006)
17. Šmerk, P.: Fast morphological analysis of czech. In: Proceedings of the Raslan Workshop 2009, Brno, Masarykova univerzita (2009)
18. Spoustová, D.j., Hajič, J., Raab, J., Spousta, M.: Semi-Supervised Training for the Averaged Perceptron POS Tagger. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece, Association for Computational Linguistics (2009) 763–771
19. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 282–289
20. Konkol, M.: Brainy: A machine learning library. In Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., eds.: Artificial Intelligence and Soft Computing. Volume 8468 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014)
21. Ciaramita, M., Altun, Y.: Named-Entity Recognition in Novel Domains with External Lexical Knowledge. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. (2005)