Czech Named Entity Corpus and SVM-based Recognizer

Jana Kravalová
Charles University in Prague
Institute of Formal and Applied Linguistics
kravalova@ufal.mff.cuni.cz

Zdeněk Žabokrtský
Charles University in Prague
Institute of Formal and Applied Linguistics
zabokrtsky@ufal.mff.cuni.cz

Abstract

This paper deals with recognition of named entities in Czech texts. We present a recently released corpus of Czech sentences with manually annotated named entities, in which a rich two-level classification scheme was used. There are around 6000 sentences in the corpus with roughly 33000 marked named entity instances. We use the data for training and evaluating a named entity recognizer based on Support Vector Machine classification technique. The presented recognizer outperforms the results previously reported for NE recognition in Czech.

1 Introduction

After the series of Message Understanding Conferences (MUC; (Grishman and Sundheim, 1996)), processing of named entities (NEs) became a well established discipline within the NLP domain, usually motivated by the needs of Information Extraction, Question Answering, or Machine Translation. For English, one can find literature about attempts at rule-based solutions for the NE task as well as machine-learning approaches, be they dependent on the existence of labeled data (such as CoNLL-2003 shared task data), unsupervised (using redundancy in NE expressions and their contexts, see e.g. (Collins and Singer, 1999)) or a combination of both (such as (Talukdar et al., 2006), in which labeled data are used as a source of seed for an unsupervised procedure exploiting huge unlabeled data). A survey of research on named entity recognition is available in (Ekbal and Bandyopadhyay, 2008).

There has been considerably less research done in the NE field in Czech, as discussed in (Ševčíková et al., 2007b). Therefore we focus on it in this paper, which is structured as follows. In Section 2 we present a recently released corpus of Czech sentences with manually annotated instances of named entities, in which a rich classification scheme is used. In Section 3 we describe a new NE recognizer developed for Czech, based on the Support Vector Machine (SVM) classification technique. Evaluation of such approach is presented in Section 4. The summary is given in Section 5.

2 Manually Annotated Corpus

2.1 Data Selection

We have randomly selected 6000 sentences from the Czech National Corpus[1] from the result of the query `([word=".*[a-z0-9]"] [word="[A-Z].*"])`. This query makes the relative frequency of NEs in the selection higher than the corpus average, which makes the subsequent manual annotation much more effective, even if it may slightly bias the distribution of NE types and their observed density.[2]

2.2 Annotation NE Instances with Two-level NE Classification

There is no generally accepted typology of Named Entities. One can see two trends: from the viewpoint of unsupervised learning, it is advantageous to have just a few coarse-grained categories (cf. the NE classification developed for MUC conferences or the classification proposed in (Collins and Singer, 1999), where only persons, locations, and organizations were distinguished), whereas those interested in semantically oriented applications prefer more informative (finer-grained) categories (e.g. (Fleischman and Hovy, 2002) with

[1] http://ucnk.ff.cuni.cz

[2] The query is trivially motivated by the fact that NEs in Czech (as well as in many other languages) are often marked by capitalization of the first letter. Annotation of NEs in a corpus without such selection would lower the bias, but would be more expensive due to the lower density of NE instances in the annotated material.

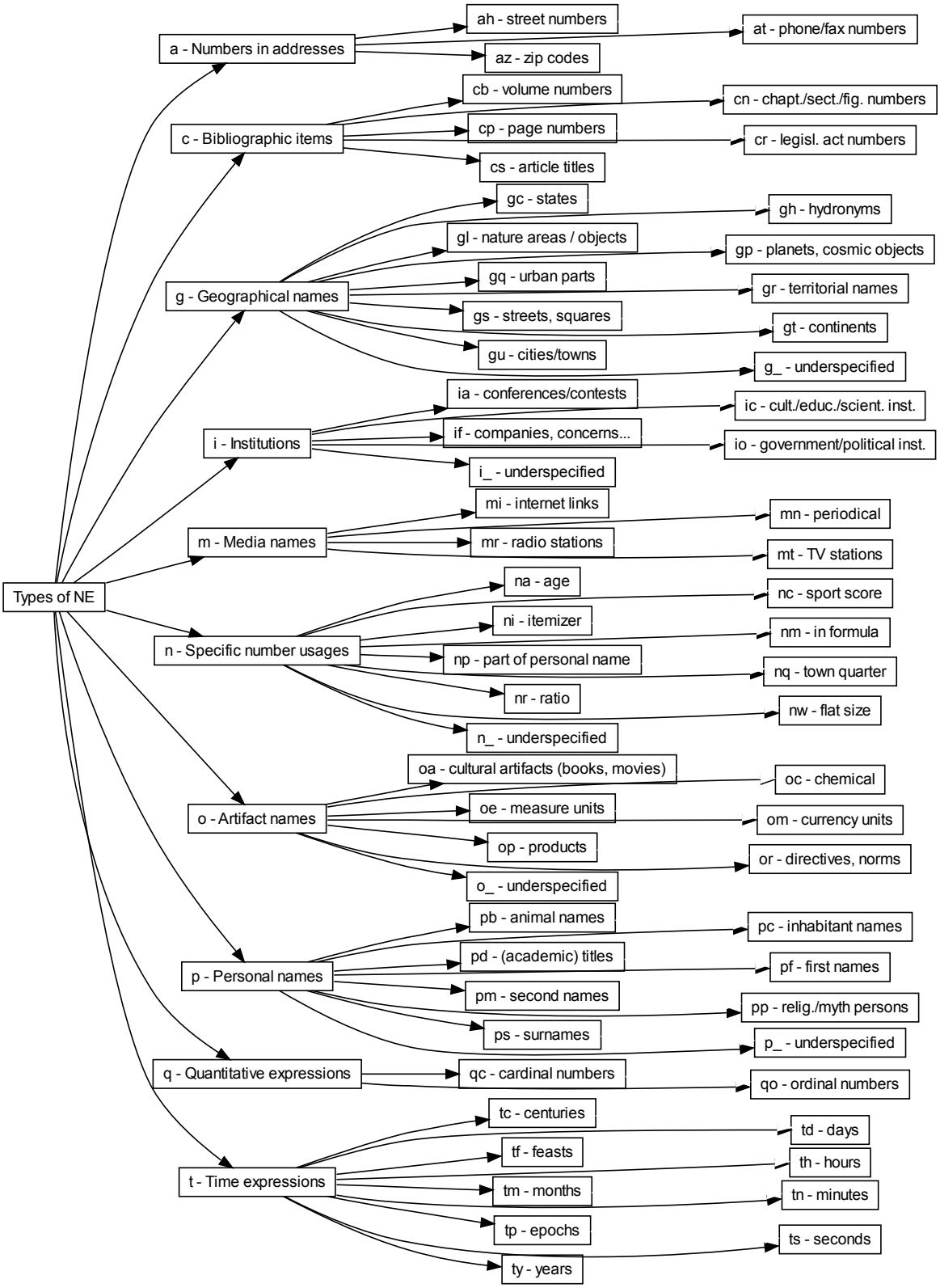

Figure 1: Two-level hierarchical classification of NEs used in the corpus. Note that the (detailed) NE types are divided into two columns just because of the space reasons here.

eight types of person labels, or Sekine's Extended NE Hierarchy, cf. (Sekine, 2003)).

In our corpus, we use a two-level NE classification depicted in Figure 1. The first level corresponds to rough categories (called *NE supertypes*) such as person names, geographical names etc. The second level provides a more detailed classification: e.g. within the supertype of geographical names, the *NE types* of names of cities/towns, names of states, names of rivers/seas/lakes etc. are distinguished.[3] If more robust processing is necessary, only the first level (NE supertypes) can be used, while the second level (NE types) comes into play when more subtle information is needed. Each NE type is encoded by a unique two-character tag (e.g., `gu` for names of cities/towns, `gc` for names of states; a special tag, such as `g_`, makes it possible to leave the NE type underspecified).

Besides the terms of NE type and supertype, we use also the term *NE instance*, which stands for a continuous subsequence of tokens expressing the entity in a given text. In the simple plain-text format, which we use for manual annotations, the NE instances are marked as follows: the word or the span of words belonging to the NE is delimited by symbols < and >, with the former one immediately followed by the NE type tag (e.g. <pf *John*> *loves* <pf *Mary*>).

The annotation scheme allows for the embedding of NE instances. There are two types of embedding. In the first case, the NE of a certain type can be embedded in another NE (e.g., the river name can be part of a name of a city as in <gu *Ústí nad* <gh *Labem*>>). In the second case, two or more NEs are parts of a (so-called) *container NE* (e.g., two NEs, a first name and a surname, form together a person name container NE such as in <P<pf *Paul*> <ps *Newman*>>). The container NEs are marked with a capital one-letter tag: `P` for (complex) person names, `T` for temporal expressions, `A` for addresses, and `C` for bibliographic items. A more detailed description of the NE classification can be found in (Ševčíková et al., 2007b).

[3]Given the size of the annotated data, further subdivision into even finer classes (such as persons divided into categories such as lawyer, politician, scientist used in (Fleischman and Hovy, 2002)) would result in too sparse annotations.

2.3 Annotated Data Cleaning

After collecting all the sentences annotated by the annotators, it was necessary to clean the data in order to improve the data quality. For this purpose, a set of tests was implemented. The tests revealed wrong or "suspicious" spots in the data (based e.g. on the assumption that the same lemma should manifest an entity of the same type in most its occurrences), which were manually checked and corrected if necessary. Some noisy sentences caused e.g. by wrong sentence segmentation in the original resource were deleted; the final size of the corpus is 5870 sentences.

2.4 Morphological Analysis of Annotated Data

The sentences have been enriched with morphological tags and lemmas using Jan Hajič's tagger shipped with Prague Dependency Treebank 2.0 (Hajič et al., 2006) integrated into the TectoMT environment (Žabokrtský et al., 2008). Motivation for this step was twofold

- Czech is a morphologically rich language, and named entities might be subject to paradigms with rich inflection too. For example, male first name *Tomáš* (Thomas) migh appear also in one of the following forms: *Tomáše, Tomášovi, Tomáši, Tomášem, Tomášové, Tomášům* ... (according to grammatical case and number), which would make the training data without lemmatization much sparser.
- Additional features (useful for SVM as well as for any other Machine Learning approach) can be mined from the lemma and tag sequences, as shown in Section 3.2.

2.5 Public Data Release

Manually annotated and cleaned 6000 sentences with roughly 33000 named entities were released as Czech Named Entity Corpus 1.0. The corpus consists of manually annotated sentences and morphological analysis in several formats: a simple plain text format, a simple xml format, a more complex xml format based on the Prague Markup Language (Pajas and Štěpánek, 2006) and containing also the above mentioned morphological analysis, and the html format with visually highlighted NE instances.

For the purposes of supervised machine learning, division of data into training, development

and evaluation subset is provided in the corpus. The division into training, development and evaluation subsets was made by random division of sentences into three sets, in proportion 80% (training), 10% (development) and 10% (evaluation), see Table 1. Other basic quantitative properties are summarized in Table 2 and Table 3.

The resulting data collection, called Czech Named Entity Corpus 1.0, is now publicly available on the Internet at *http://ufal.mff.cuni.cz/tectomt*.

Set	#Sentences	#Words	#NE instances
train	4696	119921	26491
dtest	587	14982	3476
etest	587	15119	3615
total	5870	150022	33582

Table 1: Division of the annotated corpus into training, development test, and evaluation test sets.

Lenght	#Occurrences	Proportion
one-word	23057	68.66%
two-word	6885	20.50%
three-word	1961	5.84%
longer	1679	5.00%
total	33582	100.00%

Table 2: Occurrences of NE instances of different length in the annotated corpus.

NE type	#Occurrences	Proportion
ps	4040	12.03%
pf	3072	9.15%
P	2722	8.11%
gu	2685	8.00%
qc	2040	6.07%
oa	1695	5.05%
ic	1410	4.20%
ty	1325	3.95%
th	1325	3.95%
s	1285	3.83%
gc	1107	3.30%
if	834	2.48%
io	830	2.47%
tm	559	1.66%
n_	512	1.52%
f	506	1.51%

Table 3: Distribution of several most frequent NE types in the annotated corpus.

3 SVM-based Recognizer

3.1 NER as a classification task

In this section, we formulate named entity recognition as a classification problem. The task of named entity recognition as a whole includes several problems to be solved:

- detecting "basic" one-word, two-word and multiword named entities,
- detecting complex entities containing other entities (e.g. an institution name containing a personal name).

Furthermore, one can have different requirements on what a correctly recognized named entity is (and train a separate recognizer for each case):

- an entity whose span and type are correctly recognized,
- an entity whose span and supertype are correctly recognized,
- an entity whose span is correctly recognized (without regard to its type).

Therefore, we subdivide the classification problem into a few subproblems. Firstly, we independently evaluate the recognition system for one-word named entities, for two-word named entities and for multiword named entities. For each of these three problems, we define three tasks, ordered from the easiest to the most difficult:

- Named entity *span recognition* – all words of named entity must be found but the type is not relevant. For one-word entities, this reduces to 0/1 classification problem, that is, each word is either marked as named entity (1) or as regular word (0). For two-word entities, this 0/1 decision is made for each couple of subsequent words (bigram) in the sentence.
- Named entity *supertype recognition* – all words of named entity must be found and the supertype must be correct. This is a multiclass classification problem, where classes are named entity classes of the first level in hierarchy (`p, g, i, ...`) plus one class for regular words.

- Named entity *type recognition* – all words of named entity must be found and the type must be correct.

In our solution, a separate SVM classifier is built for one-word named entities, two-word named entities and three-word named entities. Then, as we proceed through the text, we apply the classifier on each "window" or "n-gram" of words – one-word, two-word and three-word, classifying the n-gram with the corresponding SVM classifier. We deliberately omit named entities containing four and more words, as they represent only a small portion of the instances (5%).

3.2 Features

Classification features which were used by the SVM classifier(s), are as follows:

- *morphological features* – part of speech, gender, case and number,
- *orthographic features* – boolean features such as capital letter at the beginning of the word or regular expression for time and year ,
- *lists* of known named entities – boolean features describing whether the word is listed in lists of Czech most used names and surnames, Czech cities, countries or famous institutions,
- *lemma* – some lemmas contain shortcuts describing the property of lemma, for example "Prahou" (Prague, 7th case) would lemmatize to "Praha_;G" with mark "_;G" hinting that "Praha" is a geographical name,
- *context features* – similar features for preceding and following words, that is, part of speech, gender, case and number for the preceding and following word, orthographic features, membership in a list of known entities and lemma hints for the preceding and following word.

All classification features were transformed into binary (boolean) features, resulting in roughly 200-dimensional binary feature space.

3.3 Classifier implementation

For the classification task, we decided to use Support Vector Machine classification method. First, this solution has been repeatedly shown to give better scores in NE recognition in comparison to other Machine Learning methods, see e.g. (Isozaki and Kazawa, 2002) and (Ekbal and Bandyopadhyay, 2008). Second, in our preliminary experiments on our data it outperformed all other solutions too (based on naive Bayes, k nearest neighbors, and decision trees).

As an SVM classifier, we used its CPAN Perl implementation `Algorithm-SVM`.[4]

Technically, the NE recognizer is implemented as a Perl module included into TectoMT, which is a modular open source software framework for implementing NLP applications, (Žabokrtský et al., 2008).[5]

4 Evaluation

4.1 Evaluation metrics

We use the following standard quantities for evaluating performance of the presented classifier:

- precision – the number of correctly predicted NEs divided by the number of all predicted NEs,
- recall – the number of correctly predicted NEs divided by the number of all NEs in the data,
- f-score – harmonic mean of precision and recall.

In our opinion, simpler quantities such as accuracy (the percentage of correctly marked words) are not suitable for this task, since the number of NE instances to be found is not known in advance.[6]

4.2 Results

The results for SVM classifier when applied on the evaluation test set of the corpus are summarized in Table 4. The table evaluates all subtasks as defined in Section 3.1, that is, for combination

[4] *http://www.cpan.org/authors/id/L/LA/LAIRDM/*

[5] One of the reasons for integrating the classifier into TectoMT is the fact that it requires the input texts to be sentence-segmented, tokenized, tagged and lemmatized; all the necessary tools for such preprocessing are already available in TectoMT.

[6] Counting also all non-NE words predicted as non-entities as a success would lead to very high accuracy value without much information content (obviously most words are not NE instances).

	All NEs			One-word NEs			Two-word NEs		
	P	R	F	P	R	F	P	R	F
span+type	0.75	0.62	0.68	0.80	0.71	0.75	0.68	0.62	0.65
span+supertype	0.75	0.67	0.71	0.87	0.78	0.82	0.71	0.64	0.67
span	0.84	0.70	0.76	0.89	0.80	0.84	0.76	0.69	0.72

Table 4: Summary of the SVM classifier performance (P=precision, R=recall, F=f-measure). Recognition of NEs of different length is evaluated separately. The other dimension corresponds to the gradually released correctness requirements.

true type	predicted type	true type description	predicted type description	errors
oa	x	cultural artifacts (books, movies)	no entity	184
ic	x	cult./educ./scient. inst.	no entity	74
x	gu	no entity	cities/towns	71
x	P	no entity	personal name container	66
if	x	companies, concerns ...	no entity	60
x	ic	no entity	cult./educ./scient. inst.	59
io	x	government/political inst.	no entity	57
x	ps	no entity	surnames	47
P	x	personal name container	no entity	43
ps	x	surnames	no entity	41
gu	x	cities/towns	no entity	37
x	td	no entity	days	35
op	x	products	no entity	33
x	pf	no entity	first names	31
T	x	time container	no entity	30

Table 5: The most frequent types of errors in NE recognition made by the SVM classifier.

of subtask defined for all entities, one-word entities and two-word entities and with gradually released requirements for correctness: correct span and correct (detailed) type, correct span and correct supertype, correct span only.

The most common SVM classification errors are shown in Table 5.

4.3 Discussion

As we can see in Table 4, the classifier recognizes span and type of all named entities in text with f-measure $= 0.68$. This improves the results reported on this data in (Ševčíková et al., 2007a), which was 0.62. For one-word named entities, the improvement is also noticeable, from 0.70 to 0.75.

In our opinion, the improvement is caused by better feature selection on one hand. We do not use as many classification features as the authors of (Ševčíková et al., 2007a), instead we made a preliminary manual selection of features we considered to be helpful. For example, we do not use the whole variety of 15 Czech morphological categories for every word in context, but we use only part of speech, gender, case and number. Also, we avoided using features based on storing words which occurred in training data, such as boolean feature, which is true for words, which appeared in training data as named entity. We tried employing such features, but in our opinion, they result in sparsity in space searched by SVM.

It would be highly difficult to correctly compare the achieved results with results reported on other languages (such as f-score 88.76% achieved for English in (Zhang and Johnson, 2003)), especially because of different task granularity (and obviously highly different baselines). Furthermore, in Czech the task is more complicated due to inflection: many named entities can appear in several many different forms. For example, the Czech capital city "Praha" appeared in these forms in training data: *Praha, Prahy, Prahou, Prahu*.

Table 5 describes the most common errors made by classifier. Clearly, the most problematic classes are objects (`oa`) and institutions (`ic, if, io`),

which mostly remain unrecognized. The problem is that, cultural artifacts like books or movies, or institutions, tend to have quite new and unusual names, as opposed to personal names, for which fairly limited amount of choice exists, and cities, which do not change and can be listed easily.

Institutions also tend to have long and complicated names, for which it is especially difficult to find the ending frontier. We believe that dependency syntax analysis (such as dependency trees resulting from the maximum spanning tree parser, (McDonald et al., 2005)) might provide some clues here. By determining the head of the clause, e.g. *theatre, university, gallery* and it's dependants, we might get some hints about which words are part of the name and which are not.

Yet another improvement in overall performance could be achieved by incorporating hypernym discovery (making use e.g. of Wikipedia) as proposed in (Kliegr et al., 2008).

5 Conclusions

We have presented a new recently published corpus of Czech sentences with manually annotated named entities with fine-grained two-level annotation. We used the data for training and evaluating a named entity recognizer based on Support Vector Machines classification technique. Our classifier reached f-measure 0.68 in recognizing and classifying Czech named entities into 62 categories and thus outperformed the results previously reported for NE recognition in Czech in (Ševčíková et al., 2007a).

We intend to further improve our classifier, especially recognition of institution and object names, by employing dependency syntax features. Another improvement is hoped to be achieved using WWW-based ontologies.

Acknowledgments

This research was supported by MSM 0021620838, GAAV ČR 1ET101120503, and MŠMT ČR LC536.

References

Michael Collins and Yoram Singer. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, pages 189–196.

Asif Ekbal and Sivaji Bandyopadhyay. 2008. Named Entity Recognition using Support Vector Machine: A Language Independent Approach . *International Journal of Computer Systems Science and Engineering*, 4(2):155–170.

Michael Fleischman and Eduard Hovy. 2002. Fine Grained Classification of Named Entities . In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, volume I, pages 267–273.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference - 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, volume I, pages 466–471.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková. 2006. Prague Dependency Treebank 2.0.

Hideki Isozaki and Hideto Kazawa. 2002. Efficient Support Vector Classifiers For Named Entity Recognition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*.

Tomas Kliegr, Krishna Chandramouli, Jan Nemrava, Vojtech Svatek, and Ebroul Izquierdo. 2008. Wikipedia as the premiere source for targeted hypernym discovery. *WBBT ECML08*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP)*, pages 523–530, Vancouver, BC, Canada.

Petr Pajas and Jan Štěpánek. 2006. XML-based representation of multi-layered annotation in the PDT 2.0. In Richard Erhard Hinrichs, Nancy Ide, Martha Palmer, and James Pustejovsky, editors, *Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006)*, pages 40–47, Paris, France.

Satoshi Sekine. 2003. Sekine's Extended Named Entity Hierarchy. http://nlp.cs.nyu.edu/ene/.

Magda Ševčíková, Zdeněk Žabokrtský, and Oldřich Krůza. 2007. Named Entities in Czech: Annotating Data and Developing NE Tagger. In Václav Matoušek and Pavel Mautner, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*, volume 4629 of *Lecture Notes in Computer Science*, pages 188–195, Pilsen, Czech Republic. Springer Science+Business Media Deutschland GmbH.

Partha Pratim Talukdar, Thorsten Brants, Mark Liberman, and Fernando Pereira. 2006. A Context Pattern Induction Method for Named Entity Extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, pages 141–148.

Magda Ševčíková, Zdeněk Žabokrtský, and Oldřich Krůza. 2007. Zpracování pojmenovaných entit v českých textech. Technical report, ÚFAL MFF UK, Praha.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL.*

Tong Zhang and David Johnson. 2003. A robust risk minimization based named entity recognition system. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 204–207. Edmonton, Canada.