University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

Named Entity Recognition

Ing. Michal Konkol

Doctoral Thesis
Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Computer Science and Engineering

Supervisor: Ing. Roman Mouček, Ph.D.
Consulting Specialist: Ing. Miloslav Konopík, Ph.D.

Pilsen, 2015

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Rozpoznávání pojmenovaných entit

Ing. Michal Konkol

Disertační práce
k získání akademického titulu doktor
v oboru Informatika a výpočetní technika

Školitel: Ing. Roman Mouček, Ph.D.
Konzultant-specialista: Ing. Miloslav Konopík, Ph.D.

Plzeň, 2015

Declaration of Authenticity

I hereby declare that this doctoral thesis is my own original and sole work. Only the sources listed in the bibliography were used.

In Pilsen on October 9, 2015

Prohlašení o původnosti

Prohlašuji tímto, že tato disertační práce je původní a vypracoval jsem ji samostatně. Použil jsem jen citované zdroje uvedené v přehledu literatury.

V Plzni dne 9. října 2015

Ing. Michal Konkol

Acknowledgement

First of all, I would like to thank Dr. Miloslav Konopík and Dr. Ivan Habernal for steering me in the right direction in my beginnings.

I would like to thank my colleagues and dear friends, especially Dr. Tomáš Brychcín for dying so easily in computer games, Dr. Miloslav Konopík for our "wood vs. stone" discussions, Dr. Pavel Král for comming to lunch so early, Tomáš Hercig for all his sarcasm, Lukáš Svoboda for his travel stories, Dr. Ivan Habernal for all the vegetables he ate instead of me, Dr. Josef Steinberger for leaving at least one beer for me, and last but not least to Dr. Kamil Ekštein for his BMW statistics.

Many thanks goes to my parents Petr and Eva, who supported me through my studies and life in general.

Finally, I would like to thank my wife Helena for her love, tolerance, ability to survive with me, and motivation to finish this thesis.

Abstract

The idea of automatic extraction of important information from text documents comes from the time of first steps in the natural language processing. Its importance rapidly grows with the rise of the digital news, social media, blogging, etc. The amount of information is overwhelming and information extraction can help to manage it.

Named entity recognition is a critical subtask of information extraction. It tries to recognize and classify multiword expressions with special meaning, e.g. persons, organizations, locations, dates, etc. In many cases, these expressions hold the key information of the document. This information has many uses. It can be used for better organization of documents, filtering of important documents, or simply as an input for other natural language processing tasks such as machine translation, question answering, or summarization.

We believe that the there are two main problems of the current named entity recognition systems. The first problem is the necessity to fine-tune the system for every new domain or language. There is a big drop in the quality of the output, when a system designed for one domain is used for another one. The transition from one language to another is even more problematic. The second problem is the lack of semantic and external knowledge, which is crucial for people to recognize names in texts, especially in informal texts such as internet forum posts.

In this thesis, we address these problems by exploiting machine learning, semantic features, and by focusing on multilinguality. We show that this combination provides very good results and improves the adaptability and performance of the system.

Abstrakt

Automatická extrakce důležitých informací z textových dokumentů má kořeny už v počátcích oboru zpracování textu v přirozeném jazyce. Její důležitost rychle roste s rozvojem webu, novin v elektronické podobě, sociálních médií, blogování apod. Množství dostupných informací je obrovské a jejich automatické zpracování začíná být velmi důležité.

Rozpoznávání pojmenovaných entit je základní podúlohou extrakce informací. Jejím cílem je rozpoznání a třídění slovních spojení se speciálním významem, např. jména osob, organizací a míst, datumů atd. V mnoha případech tato slovní spojení skrývají klíčové informace celého dokumentu. Získané informace je možné využít mnoha způsoby. Můžeme je použít k lepší organizaci dokumentů, k filtrování dokumentů nebo jednoduše jako obohacení vstupu jiných úloh zpracování přirozeného jazyka, např. strojového překladu, zodpovídání otázek nebo sumarizace.

Podle našeho názoru trpí současné systémy pro rozpoznávání pojmenovaných entit dvěma hlavními problémy. Prvním problémem je nutnost systém opakovaně ladit pro každou novou doménu nebo jazyk. Pokud použijeme systém vytvořený pro jednu doménu na jiné doméně, dochází k výraznému zhoršení kvality výstupu. Přechod od jednoho jazyka k jinému je většinou ještě problematičtější. Druhým problémem je nepochopení významu textu a nedostatek externích znalostí, které jsou pro lidi při rozpoznávání jmen v textech velmi důležité a to především v neformálních textech jako jsou příspěvky na sociálních mediích.

V této práci se snažíme oba problémy řešit pomocí strojového učení, sémantických příznaků a zaměřením se na vícejazyčnost. Naše experimenty ukazují, že tato kombinace dosahuje velmi dobrých výsledků a zlepšuje adaptabilitu i kvalitu výstupu systému.

Contents

1 Introduction

"The lurking suspicion that something could be simplified is the world's richest source of rewarding challenges."
Edsger Dijkstra

Named entity recognition originates from information extraction (IE). The task of IE is to transform unstructured data into structured information. In our case, the unstructured data are texts written in natural language and we want to extract the important information into a well-defined format, e.g. relational database. For example we want to monitor news for mentions of terrorist attacks and for each attack we need the name of the responsible terrorist organization, location, date, and casualties. Generally, we are interested in events and for each event we want to answer questions "Who? When? Where? What? Whom?". It turned out that answers to these questions can be easily classified into classes based on their semantics (i.e. persons, organizations, locations, dates, times, etc.) and these classes are important independently of the monitored events. A logical step was to create a subtask called named entity recognition (NER) which tries to find and classify expressions belonging to these classes (named entities, NEs).

Since its introduction, NER proved to be a very useful preprocessing step for many natural language processing tasks, e.g. in question answering the answer is very often a NE (Habernal and Konopík, 2013); in machine translation the NEs are very often translated differently than other words (Nikoulina et al., 2012); in summarization segments containing NEs may be more important (Kabadjov et al., 2013); in document clustering the documents containing the same NEs are likely to be about the same subject (Steinberger et al., 2013); in information retrieval we want to handle queries about NEs differently (Artiles et al., 2010); etc. It is also used as a self-standing component, e.g. for interlinking a news article with a knowledge base.

Even though the NER task may seem easy at the first glance, it has still not been fully solved (in 2015) after 20 years of research. On one hand, the NER component is nowadays good enough to be part of many commercial systems. On the other hand, the best NER systems do not achieve the results manually done by humans. The quality of the output is still significantly dependent on the type, domain or language of processed texts. Also a significant manual work is still needed to create (or train) a new NER system for a new domain or language.

The NER task can be extended by the named entity disambiguation (NED) task. The NED task is responsible for grouping mentions of named entities referring to the same named entity or linking the mentions to a knowledge base, e.g. distinguish different persons (Louis, Neil, or Lance Armstrong) referred to by the same expression (Armstrong).

In this thesis, we experiment with various aspects of NER in order to improve the performance (in the meaning of quality) and/or the adaptability. In our research, we work with multiple methods, approaches for preprocessing of the texts, features describing the text and representations of the task. We focus especially on the use of semantic information in the NER task, which we believe is the way to a big improvement in the future. Our work in NED are the first steps in this direction for Czech language.

1.1 Thesis goals

The following goals were set for this thesis in author's Ph.D. thesis exposé (Konkol, 2012). The goals are sorted by importance with the first one being the primary goal. The last goal is a memento, that the research should be applied.

- Develop new recognition methods and features to improve performance for Czech and other languages.
- Propose semi-supervised approaches to improve the adaptability of NER.
- Experiment with disambiguation on small subset of selected named entities.

- Create quality and reusable NER system, which will provide standard interfaces.

1.2 Outline

The thesis is divided into two parts. In the first part, we summarize the theoretical background and previous work related to NER. The first part consists of 4 chapters.

Chapter 2 defines the named entity recognition task and the evaluation metrics used for this task.

Chapter 3 is an attempt to summarize all the work done in the last 20 years of research in named entity recognition as well as the current state of the art.

Chapter 4 introduces the supervised machine learning approach to named entity recognition, which is currently the state of the art. We briefly introduce algorithms, features and segment representations.

Chapter 5 describes models for distributional semantics, which are used for our experiments.

In the second part, we describe our experiments and results. It represents our contribution to the tasks of named entity recognition and disambiguation.

Chapter 6 is devoted to our experiments with semantic features. We propose our semantic features and their combination. This semantic features are used in a multilingual system with a very good performance.

Chapter 7 is focused on segment representations. We evaluate and compare various types of segment representations and show, that choosing the best segment representation is not straightforward.

Chapter 8 covers our experiments with various methods for stemming and lemmatization as preprocessing for NER. Stemming and lemmatization is (currently) considered as a must for highly inflectional languages.

Chapter 9 is a self-standing part of this thesis, which introduces named entity disambiguation and our first steps in this task.

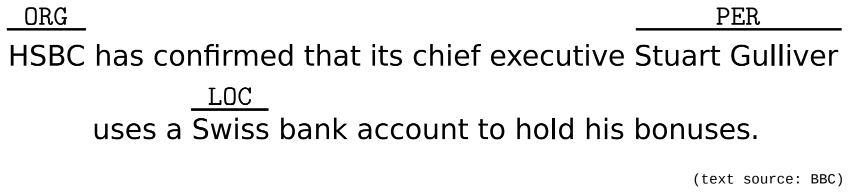
Chapter 10 briefly introduces the design of our NER system.

Chapter 11 summarizes our work, reveals our future plans and discusses the fulfillment of our goals.

2 Task definition

"Measure what is measurable, and
make measurable what is not so."
Galileo Galilei

The NER task tries to correctly detect and classify textual expressions into a set of predefined classes. The classes may vary, but very often classes like persons (PER), organizations (ORG) or locations (LOC) are used. An example NER output is shown on Figure 2.1.


ORG
PER
HSBC has confirmed that its chief executive Stuart Gulliver
LOC
uses a Swiss bank account to hold his bonuses.

(text source: BBC)

Figure 2.1: Example named entity recognition output.

Each NE has two primary properties: span and type. Suppose that our text contains "Bank of England". The type of the entity is organization and it spans three words. Marking only "England" with type organization is surely incorrect as well as marking "Bank of England" as country. Marking "England" as a country is questionable (in this case) and correctness of this output depends on our needs. As you can see, there are various possible definitions of the correct answer. In the following section, we describe evaluation metrics used for NER together with multiple definitions of the desired output.

2.1 Evaluation metrics

In any area of research it is important to evaluate and compare results of new methods. There is thus a need to use some objective measure (or measures), which would well cover the purpose of the research.

Unlike some other NLP tasks (e.g. Machine Translation) NER uses a standard set of metrics (even though the use may vary), which is generally accepted. This set includes three metrics to describe the performance of NER system, each for different aspect of the task. These metrics are called precision, recall and F-measure (also F-score or F_1 score).

We will define these measures on a general classification of objects into two classes: positive and negative. There exist four following classes of classification results.

- Positive (P) - positive object marked as positive.
- Negative (N) - negative object marked as negative.
- False positive (FP)- negative object marked as positive.
- False negative (FN) - positive object marked as negative.

This is shown well on Figure 2.2, where the curves show the distribution of positive and negative objects, the dotted line shows the decision threshold of the classifier. In the areas denoted as FN and FP are some objects marked incorrectly.

Now we can easily define precision, recall and F-measure as follows.

$$\text{Precission} = \frac{P}{P + FP} \tag{2.1}$$

$$\text{Recall} = \frac{P}{P + FN} \tag{2.2}$$

$$\text{F-measure} = \frac{2P}{2P + FP + FN} \tag{2.3}$$

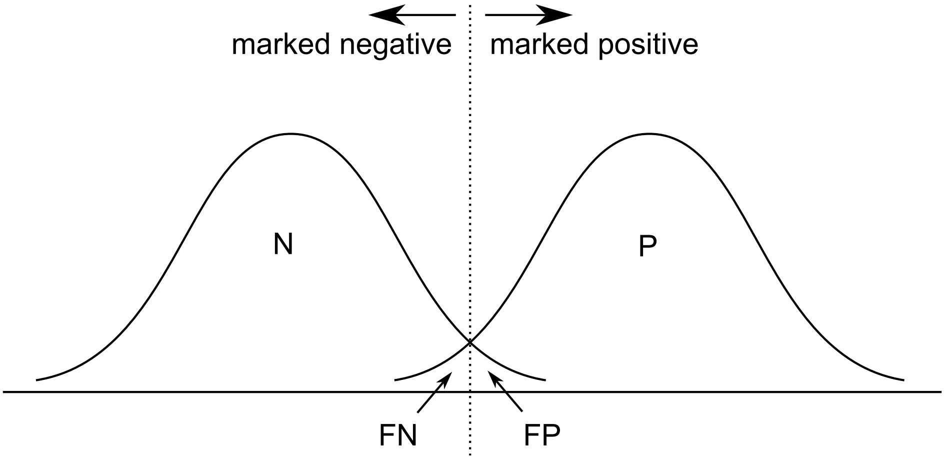


Figure 2.2: Precision and recall

Precision is a measure of trust, that the objects marked as positive are really positive. Recall is a measure of trust, that all the positive objects are marked. It is obvious that precision and recall describe different aspects of results. Moreover, these measures are competing. As shown on Figure 2.3, if the decision threshold moves to the left, there will be fewer FN objects and more FP objects, resulting in higher recall and lower precision. This is important in evaluation of a classifier, because high recall (resp. precision) classifier can be better for various tasks. F-measure is a harmonic mean between precision and recall and represents the overall perspective.

Now it is possible to define, what is counted as P, N, FP and FN. Multiple definitions were proposed. The following sections cover the most common of them.

2.2 MUC-6 evaluation

The NER task was introduced at MUC-6 (Grishman and Sundheim, 1996). So the initial evaluation technique was defined at this conference. The choice of evaluation metrics was based on other information extraction tasks. Since that time precision, recall and F-measure are used as a standard in NER.

At MUC-6 span (or text) and type of the entity was handled separately.

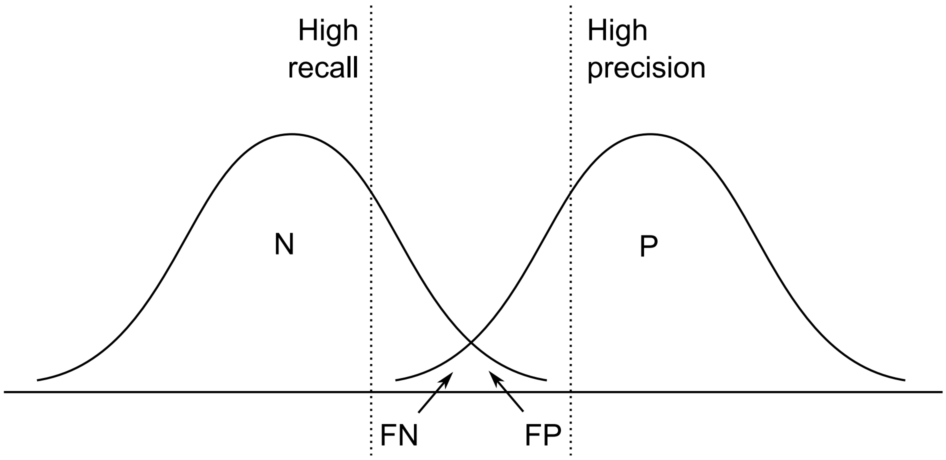


Figure 2.3: Precision and recall change

The type is counted as correct, if it is the same as the original type and the span overlaps the entity. The text is considered correct, if it matches the original text of the entity. The text comparison involves operations like trimming or removing unimportant parts (ltd., the, etc.). Each entity can have an alternative text, which is also considered as correct.

Three numbers were counted for both type and span: COR (correct answers), POS (number of original entities) and ACT (number of guesses). The overall results of the system were acquired by adding these number for type and span together. I.e. correctly marked entity adds 2 to COR. The precision, recall and F-measure were then computed in a standard way using these numbers.

The evaluation techniques for all MUC tasks are covered in The Message Understanding Conference Scoring Software User's Manual[1].

2.3 CoNLL evaluation

The CoNLL 2002 (Tjong Kim Sang, 2002) and 2003 (Tjong Kim Sang and De Meulder, 2003) have used an exact match evaluation. The entity is considered correct only if it has exactly the same span and type.

[1] http://www-nlpir.nist.gov/related_projects/muc/muc_sw/muc_sw_manual.html

The advantage of this method is, that it is clear, simple and gives a lower estimate of the evaluated system. The disadvantage is, that in some cases it is too strict. If the original entity is “The United States” and “United States” is marked by the system, then “The United States” is considered as FN and “United States” as FP. The result is, that the system is penalized in two ways for almost correct answer.

2.4 ACE evaluation

The Automatic Content Extraction program consists of various NLP tasks. There are two tasks directly focused on NER, entity detection and tracking (EDT)(Doddington et al., 2004) and time expression recognition and normalisation (TERN) (Ferro et al., 2005). Both tasks extends the standard definition of NER tasks with deeper level of detail.

The evaluation of EDT task did not used standard metrics. The evaluation is based on a special scoring system, where each type of error and also each type of entity has different weight. The scoring system is very complex. On one hand, it can be adjusted and used to properly evaluate systems regarding various needs. On the other hand, the weights must be the same to compare two systems and it is hard to get direct feedback.

2.5 Lenient evaluation

The GATE framework, a widely NLP toolset, has multiple options of evaluation. While evaluating using F-measure, they offer two values: strict and lenient evaluation. The strict evaluation is equivalent to the CoNLL evaluation. The lenient evaluation is not commonly used in NER, but we found it very helpful.

The problem of the CoNLL evaluation is, that in many cases the system outputs almost correct results, but they are regarded as two mistakes. The lenient metric tries to loosen this conditions. If the result has correct type and the span overlaps with entity of the same type, the lenient metric considers it as correct.

We need to redefine the classes to rewrite the general formulas. We distinguish the following classes when comparing the truth (human annotated data) with the system output.

- Correct c, if the entity is marked on correct span with correct type.
- Partially correct pc, if the entity is marked with correct type, but the span is not exact. For each entity outputted by system.
- Partially marked pm, if the entity is marked with correct type, but the span is not exact. For each entity in the data.
- Not correct nc, if something is marked, but it is not an entity.
- Not marked nm, if entity is not marked.

An example of these classes is shown on Figure 2.4. The entities on top are the truth and the ones on the bottom are the system output. The entity "United States of America" is not classified correctly. It is *partially marked*, because there is at least one entity with overlap that has the correct type. The entities "United States" and "America" outputted by the system are *partially correct.* The found entity "republic" is *not correct.* The entity "Bank of England" is *correct.* The entity "United Kingdom" is *not marked.* To summarize the example, we have $c = 1$, $pc = 2$, $nc = 1$, $pm = 1$ and $nm = 1$.

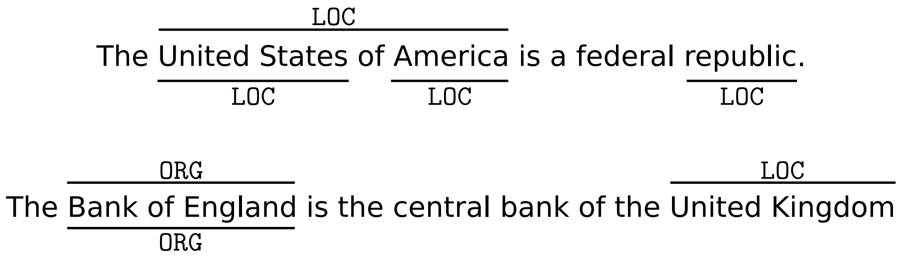

LOC
The United States of America is a federal republic.
LOC
LOC
LOC
ORG
LOC
The Bank of England is the central bank of the United Kingdom
ORG

Figure 2.4: Example of the classes used to compute strict and lenient metrics.

Following our definitions, the equations for precision (2.1), recall (2.2) and F-measure (2.3) for strict (CoNLL) and lenient metric can be rewritten as follows.

Strict

$$\text{Precission} = \frac{c}{c + nc + pc} \tag{2.4}$$

$$\text{Recall} = \frac{c}{c + nm + pm} \tag{2.5}$$

$$\text{F-measure} = \frac{2 \cdot c}{2 \cdot c + nc + nm + pc + pm} \tag{2.6}$$

Lenient

$$\text{Precission} = \frac{c + pc}{c + nc + pc} \tag{2.7}$$

$$\text{Recall} = \frac{c + pm}{c + pm + nm} \tag{2.8}$$

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.9}$$

The main advantage of using both these metrics is, that the strict metric defines the lower bound on the useful outputs of the system, because some partially correct results are still very useful, e.g. “United States” instead of “United States of America”. The lenient metric defines the upper bound, because it tries to cover all possibly correct results, but in some cases treats errors as partially correct, e.g. “Bank” instead of “Bank of England”. We can say that the effective performance is somewhere between the lower and upper bound.

We should note, that our equations are not the same that are implemented in GATE, but only inspired by GATE. These equations may be slightly different from the one used in GATE, because GATE does not use the *partially marked* class. The partially marked class is important when the system output contains two (partially correct) entities related to one entity in the data.

3 Related work

"I find my greatest pleasure, and
so my reward, in the work that precedes
what the world calls success."
Thomas A. Edison

The named entity recognition (NER) was defined as a subtask of information extraction in 1995. Since that time, the NER systems went through many changes. In this chapter, we will make a survey of these changes. There are many different views on NER systems, therefore we will cover each view in a separate subsection. Of course, these views are interconnected with each other and very important papers are covered in several of them.

3.1 NER applications

Most often NER is used as a preprocessing tool for another natural language processing tasks. In the *machine translation* task, there are multiple reasons why it is necessary to handle NEs with special care. First, NEs are very sparsely represented in texts. A name can appear only once even in a big corpus and the context based patterns (or features) used in state-of-the-art machine translation systems are unable to solve this. NEs of the same type appear in similar contexts, which can significantly reduce the sparsity problem. Second, NEs are ambiguous with "normal" words (e.g. Bush), but they are translated differently. Person names are usually not translated, but can be transliterated. Dates are translated in specific formats. Units can be changed from imperial system to metric system (e.g. 3.2 feet to 0.97 meters). Some location names are translated while others are not. Multiple approaches for integration of NER were proposed (Nikoulina et al., 2012; Pal et al., 2010; Hermjakob et al., 2008; Huang, 2006).

NER is a core component in *question answering* systems (Habernal and Konopík, 2013; Mollá et al., 2007; Zaanen and Mollá, 2007; Molla et al., 2006; Toral et al., 2005; Narayanan and Harabagiu, 2004). NEs are used both in parsing of the query and in searching for the answer. The answer is very often a NE as a natural answer to questions "Who? When? Where?". The semantic information in a question is also very often connected to NEs, e.g. the important information in the question "When were Isaac Newton born?" is the NE and its property birth.

There is a rise of NER in *information retrieval*. Modern systems such as Apache Lucene allow us to extend the query with custom properties. As NEs are very important in many systems, it is crucial to allow the user to use them. This also applies for the search engines such as Google or Yahoo, which try to handle the query containing or asking for NEs differently, e.g. they show a box with basic information about the NE with a link to a knowledge-base. Also the results for NEs are ranked in a different way (Artiles et al., 2010).

In *summarization* we can use the information about NEs to choose better segments of text (Kabadjov et al., 2013; Nobata et al., 2003, 2002). As we have already written, NEs often hold the key information in the document. Thus in summarization, we can expect that sentences containing NEs are important.

The role of NER in *sentiment analysis* is to find targets of the sentiment (Souza and Vieira, 2013; Kumar and Sebastian, 2012). A good example is mining of a forum about cell phones. Sentiment analysis can tell us which post is positive or negative, but the information is much more interesting if we know which post is about Nokia and which is about Sony. If we know the target (NE) in advance, we can also create a prior distribution of sentiment about this entity (Brychcín and Habernal, 2013).

NER was also successfully used in (multilingual) *document clustering* (Steinberger et al., 2013; Montalvo et al., 2006a,b). The motivation is, that documents (e.g. news articles) mentioning the same entities are probably similar. If the documents are from different languages, it is necessary to have NER system for each language. The recognized NEs need to be linked across documents, because they can be written differently, transliterated, etc.

The results of NER features in *document classification* are inconsistent. The results of Král (2014) and Moschitti and Basili (2004) show that the performance is not significantly different if we use NEs. Liu and Li (2009) reports a significant improvement. Gui et al. (2012) states that NEs help only in well-represented classes and they are not useful in rare classes.

NER can also be used as a *self-standing* component in common user applications, where we need to mark NEs or connect them to a knowledge base. For example online publishing news papers try to interlink their articles with Wikipedia through NEs.

3.2 Methods

There are various aspects which can be used to divide or describe the methods. The divisions presented in the following paragraphs are compilation of notations used in NER or generally NLP field. They are not rigid, but they should give some idea about the methods.

Most often, the systems are divided into two groups – *rule-based* and *machine learning.* The rule-based systems work on the basis of rules created by an domain expert. The term is vague in some cases, because the rules can be learned from data by some machine learning algorithm. In fact, all the machine learning approaches create a (huge and very complex) set of rules. Sometimes, the term *hand-crafted* is used to explicitly admit the expert work.

The machine learning systems use some data to learn regularities or patterns, which can be exploited to find entities. They can be further divided based on the type of data they use. If the system needs corpus with already labeled entities, then the system uses *supervised learning.* The system uses *unsupervised learning*, if it does not use any examples of desired output, e.g. no input of type "Prague → city". *Semi-supervised learning* is a special class of supervised learning, where the system uses labeled data, but it can also exploit unlabeled data. In the case of NER, authors often use the term unsupervised for semi-supervised systems.

At the MUC-6 in 1995 (Sundheim, 1995), the majority of the presented systems was rule-based, e.g. (Iwanska et al., 1995; E. Appelt et al., 1995), but there were few exceptions that were based on or incorporated machine learning techniques, e.g. (Cowie, 1995; Fisher et al., 1995). A higher percentage of machine learning systems appeared at the MUC-7 (Lin, 1998; Borthwick et al., 1998), but there were still majority of rule-based systems (Fukumoto et al., 1998; Black et al., 1998). At the CoNLL NER shared tasks (2002 and 2003), there were no systems based purely on rules. This trend continued and currently, most of the NER systems presented by researchers are based on machine learning techniques. Single purpose commercial systems are still often based on the rule-based approach. Steinberger et al. (2013) present a rule-based multilingual NER system used in European Media Monitor[1] and show some advantages of this approach.

The main advantage of machine learning systems is their adaptability. While an expert is needed to create rules for new domain or language in the rule-based approach, machine learning approaches rely on data annotations, that can be done by (almost) anyone. The main advantage of rule-based systems is their deterministic execution, that is easily understandable by human. It is easier to analyze rules that were used for a sentence than combinations of hundred thousand features in the machine learning approach. Nevertheless, it may be hard to add rules to fix errors in a complex system, i.e. system with a lot of rules (Miller et al., 1998). Another advantage of rule based systems is their ability to handle entities with complex structure (e.g. addresses).

Most of the systems use the supervised learning paradigm. Many machine learning algorithms have been used for NER. The list includes hidden Markov models (Bikel et al., 1999; Zhou and Su, 2002), decision trees (Cowie, 1995; Paliouras et al., 2000; Ševčíková et al., 2007), support vector machines (Takeuchi and Collier, 2002; Ekbal and Bandyopadhyay, 2008; Kravalová and Žabokrtský, 2009), maximum entropy classifiers (Chieu and Ng, 2003; Curran and Clark, 2003; Bender et al., 2003; Konkol and Konopík, 2011), maximum entropy Markov models (McCallum et al., 2000; Straková et al., 2013) and conditional random fields (McCallum and Li, 2003; Lin and Wu, 2009; Konkol and Konopík, 2013). Currently, conditional random fields are regarded as the best self-standing method. We describe the supervised machine learning approach to NER in Section 4, where we also briefly introduce some of the mentioned methods.

To our best knowledge, there is no fully unsupervised NER system, i.e. NER system that does not need any examples of entities "Prague → city" or rules "city of X → X = city". Some systems are called unsupervised, because they use only a few examples (Etzioni et al., 2005; Collins and Singer, 1999). These systems are (clearly) semi-supervised, if we hold strictly to the definitions. They usually use *bootstrapping*, a two-phase iterative technique. In

[1] http://emm.newsbrief.eu/overview.html

the first phase, we use the set of examples to find context patterns typical for NEs. In the second phase, we apply these patterns to extend the set of examples. The patterns need to have high precision to achieve good performance (Etzioni et al., 2005). Such systems were introduced by Etzioni et al. (2005); Nadeau et al. (2006); Collins and Singer (1999). This approach is closely related to automatic gazetteer creation (Kazama and Torisawa, 2008; Pasca et al., 2006). Gazetteers can also created using rule-based systems (Rau, 1991).

It has been shown, that combination of methods can outperform all combined (single) methods (Florian et al., 2003). There are multiple ways how to combine methods. One of the basic approaches is *voting* (van Halteren et al., 2001; Kozareva et al., 2007). In voting, each of N NER systems selects one of the Y classes, we denote this vote v_i for the i-th system. We then count the votes for each class y_j (for the j-th class) and combine them using weights w_i to get the final class y according to (3.1), where $\mathbf{1}_{v_i=y_j}$ is a binary function equal to one, if the condition holds. If the NER system models a distribution $p_i(y|x)$ of classes y based on context x, then $v_i = \arg\max_y p_i(y|x)$. In the simplest case, all the weights are equal and we talk about *majority voting.* In the other cases, we use *weighted voting.*

$$y = \arg\max_{y_j} \sum_{i}^{N} w_i \mathbf{1}_{v_i=y_j} \tag{3.1}$$

Sometimes, the term voting is used interchangeably with *linear interpolation* (Florian et al., 2003). This can be misleading, because linear interpolation combines the probability distributions $p_i(y|x)$ into a single distribution $p(y|x)$ according to (3.2) and not only the most probable classes. It is clear, that both approaches lead to different results.

$$p(y|x) = \sum_{i}^{N} w_i p_i(y|x) \tag{3.2}$$

For both voting and linear interpolation, the weights may depend on multiple factors, e.g. class (classifier is only good for one class), context (classifier works well only in specific contexts). Multiple approaches were introduced to find optimal weights, including genetic algorithms (Desmet and Hoste, 2010; Ekbal and Saha, 2010) or multi-objective optimization (Ekbal and Saha).

Another approach for method combination is *stacking* (Florian, 2002). This approach uses output of one NER system as a feature for second one. We can further extend this basic pattern and create model of multiple layers, where outputs of classifiers in the lower layer are features for classifiers in the upper layer. In a special case called *repeated classification*, we use the same model over and over (Straková et al., 2013).

3.3 Languages

Considering the language, we can define two views on the NER systems. First, we look at each language separately. Second, we study the transition from monolingual to multilingual systems.

3.3.1 Individual languages

In this section, we provide examples of systems for individual languages together with their results. The results show, that there are big differences between languages, that are probably caused by their properties – word order, morphology, rules for writing proper names, etc. We are not trying to cover all languages and all systems, but rather give an idea about the performance across languages.

The research was mainly conducted on *English*. We do not try to cover all work done on English, but only some systems that currently represent the state of the art. The best system at CoNLL-2003 was created by Florian et al. (2003). It achieved 88.76% F-measure. Even though many systems have currently better results, they are often compared to this system. The state of the art results were achieved by Lin and Wu (2009) (90.90%), Ratinov and Roth (2009) (90.57%), Turian et al. (2010) (90.36%), and Tkachenko and Simanovsky (2012) (91.02%).

Our native language is *Czech*, so it is natural that we focus on it. The first system for Czech was introduced by Ševčíková et al. (2007). They proposed two baselines. First baseline was based only on the capitalization of words and ended up with 16% F-measure. The second baseline created a dictionary of entities found in the training data and search them in test data and had 43% F-measure. Their system based on decision trees improved the results to 68%. It proves that trivial approaches are not sufficient to solve the NER task. They also created the Czech Named Entity Corpus (Section 3.6) that is currently a standard for evaluation of Czech NER. This system was followed by Kravalová and Žabokrtský (2009), who used support vector machines and further improved the results to 71%. Two other systems were introduced by Konkol and Konopík (2011) (72.94%) and Král (2011) (58%). Konkol and Konopík (2013) shows that the previous papers used different evaluation metrics. They created a system based on conditional random fields and evaluated it using all the previously used metrics. They achieved 74.08% F-measure using the standard CoNLL evaluation and outperformed the previous systems. At the same time, Straková et al. (2013) presented a system based on maximum entropy Markov models. They achieved 82.82% using the same evaluation as the first two systems. Konkol and Konopík (2014) studied various approaches of stemming and lemmatization used for Czech NER. They achieved slightly better (74.23%) results than in (Konkol and Konopík, 2013). Konkol et al. (2015) implemented a language-independent system and evaluated it also on Czech. With 74.08% F-measure the results are on the same level as the state-of-the-art language-dependent systems. Demir and Ozgur (2014) introduced a system exploiting a neural-network based word embeddings and achieved 75.61%.

Spanish was one of the languages, that have been covered at the CoNLL-2002 conference. The best (out of 12) system for Spanish at this conference was created by Carreras et al. (2002). It was based on AdaBoost and achieved 81.39%. Ferrández et al. (2006) introduced system based on a combination of a machine learning and a rule based approach. The machine learning system was used as an input for the rule based part, which made the final decisions. To our best knowledge, they currently hold the best result for Spanish with 83.37%. Another system was presented by Kozareva et al. (2007), who combined multiple machine learning methods using weighted voting. Their system avoids the use of morphological and syntactical features, because these features lower the adaptability of the system. They final score on the CoNLL-2002 data was 78.59%, which would be the third place on the original conference. Konkol et al. (2015) introduced a language-independent system, which achieved 83.08% on Spanish, i.e. worse only by 0.29% than the best, language-dependent system.

The *German* NER started with the CoNLL-2003. At CoNLL-2003 the best system for German (Florian et al., 2003) had 72.41%. Faruqui and Padó

(2010) presented a system which outperformed the previous systems with 78.2%. Their success was based on the use of semantic and morphological similarity of words. Recently, a GermEval NER shared task (Benikova et al., 2014) presented a series of systems. The workshop organizers announced three best systems (Benikova et al., 2014).

Dutch is the last language selected for the CoNLL conferences. The best system at CoNLL-2002 had 77.05%. Curran and Clark (2003) at CoNLL-2003 evaluated their system also on Dutch and ended up with 79.63%. Konkol et al. (2015) presented a multilingual system using latent semantics, which achieved 83.01%. Desmet and Hoste (2010) tried to use weighted combination of various methods, where the weights were assigned based on genetic algorithms. The evaluation was done on the SoNaR corpus and achieved 84.44%. The results are not comparable with the other systems, because of the different corpora.

The rest of the languages are not used in experiments in this thesis and we cover them only informatively. There are systems for *Chinese* (Fu and Luke, 2005), *Japanese* (Sasano and Kurohashi, 2008), *Estonian* (Tkachenko et al., 2013), *Hungarian* (Varga and Simon, 2007), *Turkish* (Demir and Ozgur, 2014), *Indian languages* (Ekbal and Saha, 2011), *Bulgarian* (Georgiev et al., 2009), *Arabic* (Shaalan, 2014).

It is evident that the language has a major impact on the NER system performance. The majority of the research is done on English, which seems to be one of the easiest languages. All the languages have some special properties, that play a crucial role in performance. These properties include the level inflection and word-order freedom (Konkol and Konopík, 2014), capitalization (Faruqui and Padó, 2010), tokenization (different tokenization for entities in Chinese) (Gao et al., 2005), agglutination (Shaalan, 2014).

3.3.2 Multilinguality

The NER task was defined at MUC-6 (Grishman and Sundheim, 1996). This conference was focused purely on English. The following conferences gradually attached more importance to processing multiple languages. At MUC-7/MET-2, the presented NER systems processed English, Japanese and Chinese, but it was not mandatory to evaluate the system on all these languages. In fact, the majority of the systems were evaluated on only one of these languages (pro, 1998).

For both CoNLL-2002 (Tjong Kim Sang, 2002) and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), all systems had to be evaluated on a pair of languages (Dutch and Spanish, English and German). Although the systems presented at these conferences are generally considered multilingual, they had different levels of language independence. Arguably, the systems were able to adapt to a new language only to a limited extent without some expert work (e.g., part-of-speech, gazetteers were required).

Currently, there are multiple state-of-the-art systems, which are (or should be) able to process a wide range of languages (Lin and Wu, 2009; Konkol et al., 2015; Steinberger et al., 2013). These systems avoid purposely language dependent tools like lemmatizers, POS taggers, or chunk taggers for two reasons. First, they are not available for many languages. Second, these tools are usually not multilingual or support only a limited set of languages. It is then hard to integrate and manage them in a single highly multilingual system (Steinberger et al., 2013).

The key to the highly multilingual systems is their adaptability. A highly multilingual system need to be able to adapt to a new language easily without much effort.

3.4 Domains

Another aspect of NER is the domain of the data. NER was applied on a wide variety of domains including news articles, biomedical applications, business, social media, parliament speeches, Wikipedia, etc. Some domains seem to be easier than other domains, e.g. news papers and social media. It has been shown, that system trained on one domain performs significantly worse on the other domains. Ciaramita and Altun (2005a) has shown a gap greater than 26%, when they trained a system on the CoNLL corpus and used it on texts from Wall Street Journal. Similar performance degradation was reported in (Poibeau and Kosseim, 2001) for NER trained on MUC-6 corpus and used for more informal texts like emails. The domain also evolves in time. The language and style of the texts change. Entities appear (new company, new president, etc.) and disappear (company bought by other company, retirement, etc.).

The general goal is to create a domain-independent system, i.e. system that works for all domains similarly to human. Currently, there are two approaches. In the first approach, we create a domain-dependent system and an adaptation layer, which is responsible for altering the system for other domains (Guo et al., 2009). In the second approach, we directly try to create domain-independent system. The first step in this direction is to replace the domain-dependent features (e.g. gazetteers, rules, etc.) with more general features (Faruqui and Padó, 2010), i.e. to limit the work necessary to train the system for other domain.

3.5 Annotation schemes

An important perspective is based on the annotation scheme used for NER data.

It is obvious, that recognition of some classes is much harder than other classes, e.g. if we choose to recognize given names, it is easier than locations, because the variability of given names is usually lower than variability of location names. The number of classes is also a factor. Generally, classification is harder if the number of classes is high, because the classes would probably be close together (i.e. harder to separate) and more parameters or rules would be needed. A certain amount of data is needed to train each class. With high number of classes, it is necessary to annotate more data (in the machine learning approach) or write more rules (in the rule-based approach).

Another view is the structure of entities. The entity classes can be simply on the same level or they can be organized hierarchically, e.g. person is a superclass of actor. Sekine et al. (2002) defined and gradually extended a complex hierarchy, which contains about 150 classes. The hierarchy can be defined in various ways. Sekine et al. (2002) uses semantic hierarchy (person is a superclass of actor), but there are also examples of functional hierarchy (person name is superclass for surname and forename) (Ševčíková et al., 2007).

The entities can also be nested (Finkel and Manning, 2009), e.g. an organization name (e.g. Bank of England) contains a location (e.g. England). Even though the problem of nested entities looks harder, Konkol and Konopík (2013) shows that it can actually be easier in some cases.

3.6 Corpora

For the evaluation of the NER task, it is necessary to create a corpus with marked entities. If we use a supervised machine learning approach, we need the corpus also for estimation of optimal parameters. Most often, the corpus is created by humans and we call it the *gold data*. Sometimes the corpus can be created automatically from already existing resources (Hahm et al., 2014; Nothman et al., 2008). This approach usually allows us to create much bigger corpus, but the corpus contains errors. We use the term *silver data* for such corpus.

The corpus is usually divided into parts with different purpose. The *training* part of the corpus is used for training of parameters of the system. The *test* part is used only for the final evaluation of the system. Evaluating the system on the test set during its development is a bad practice, because the final results are probably better than they would be on unseen data. The *heldout* or *validation* part is used to verify the system design during development and also to find the optimal hyperparameters of the model (e.g. size of windows for features, parameters for combination of models).

In the machine learning approach the size of the data has a major impact on the results (An et al., 2003). A complex system trained on too few data will not be able to generalize and the results on unseen data can be catastrophic. This problem is called *overfitting*. A less complex system needs less data, but it may be unable to model the problem properly. For this reasons, *cross-validation* is often used. In this technique, the corpora (or the training and validation part) is split into $N > 2$ parts. In the i-th iteration ($i = 1, \ldots, N$), the system is trained on all parts except the i-th part and the i-th part is used for evaluation. In this manner all the data are used for both training and evaluation.

The most commonly used corpora have been prepared for the CoNLL conferences. Corpora for four languages were created: Spanish, Dutch, English and German. These corpora use four entity classes – PER (person names), ORG (organizations), LOC (locations) and MISC (miscellaneous). The sizes of these corpora are around 250,000 tokens.

The first corpus was created for the MUC-6 conference, where NER was defined. It contains 130 (100 training, 30 test) articles from the Wall Street Journal (Sundheim, 1995).

For Czech, we have two corpora. The first one, the Czech Named Entity Corpus was created by Ševčíková et al. (2007). The first version consisted of 5868 sentences. The second version was extended with 125 sentences containing email addresses and 3000 sentences with only a few entities to model the real distribution of entities in texts[2]. The entities can be nested. Originally, the entities were organized into a two-level hierarchy with 10 classes and 64 subclasses, but some of the classes were not marked throughout the whole corpus. That is the reason why only a subset of classes is used for the evaluation. The corpus was later transformed into the CoNLL-like format (together with some minor changes) by Konkol and Konopík (2013). All versions of the corpus are publicly available for non-commercial purposes.

We have created the second corpus for the Czech News Agency (CTK). It uses 14 classes specifically designed for the purposes of CTK. The corpus contains 333,754 tokens. Unfortunately, it is not publicly available.

More corpora are available for different languages and domains: SoNaR corpus (Desmet and Hoste, 2010), Hungarian (Varga and Simon, 2007), Bulgarian (Georgiev et al., 2009), Indian (Ekbal and Saha, 2011), Turkish (Demir and Ozgur, 2014), Estonian (Tkachenko et al., 2013), etc.

[2] http://ufal.mff.cuni.cz/cnec/cnec2.0

4 Machine learning approach to NER

"An algorithm must be seen to be believed."
Donald Knuth

In this section, we will describe the supervised machine learning approach to NER, which is today the most commonly used approach among researchers and is also used in our experiments later in this thesis.

The schema of this approach is depicted on Figure 4.1. There are two phases: training and test. The input of the training phase is simply the text and labels, where labels are triples (start index, end index, type). This input can be preprocessed in multiple ways, commonly with tokenization, part-of-speech tagging, stemming, or lemmatization.

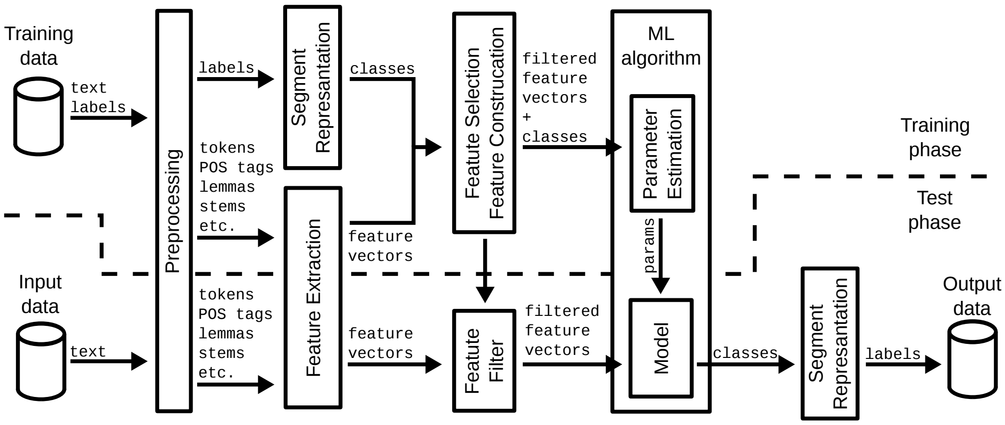


Figure 4.1: General schema of supervised machine learning approach to NER.

The next step is the transformation of the input to a machine understandable format. The token (≈ word) is taken as a basic unit for the classification.

Thus for each word, we create a vector representation, where each important property (of the word and its context) is represented by a number. This process is called *feature extraction* in this thesis (the terminology is ambiguous). A class is assigned to each feature vector based on the labels based on the chosen *segment representation*.

The *feature selection* and/or *construction* step is optional. The purpose of this step is to reduce the feature vector dimension without reducing the information (feature selection) or to create new features by their combination (feature construction).

The last step is the *parameter estimation* of the selected machine learning algorithm. The training phase ends with creation of a *model*, which is used in the test phase.

In the test phase we follow the same steps, but without labels in the input data. The same features have to be used. The *feature filter* remembers and applies the decisions about features made by feature selection in the training phase. The *model* is responsible for assigning classes to tokens based on its experience acquired on the training data. The experience is expressed by parameters of the model. The classes assigned by the model are transformed to labels using the chosen segment representation.

We describe some of the common machine learning algorithms in Section 4.1. Features used in NER are covered in Section 4.2. Segment representations are introduced in Section 4.3.

4.1 Algorithms

4.1.1 Hidden Markov models

Markov models are modelling a Markov process and are based on a state graph. Markov process is a stochastic process for which the state transmission probability distribution depends only on the present state. hidden Markov models are modelling a process, where the states are not directly observable. Hidden Markov model is fully described by the following properties.

- $X = \{x_1, \ldots, x_n\}$ – Set of observations.

- $Y = \{y_0, y_1, \ldots, y_m\}$ – Set of states.
- y_0 – Initial state.
- $p(y_k|y_{k-1})$ – The state transition probability distribution, where k is the position in the sequence of states from X.
- $p(x|y_k, y_{k-1})$ – The observation emission probability distribution.

For NER the states are the NE classes and observations are words. The Viterbi algorithm is then used to find the sequence of states (NE classes) with highest probability. The Baum-Welch algorithm can be used to improve the parameters of HMM using unmarked texts. A typical example of a HMM system is (Zhou and Su, 2002). HMM were only used in combination with other classifier at CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003). Recent systems often prefer conditional random fields, which have similar ability to handle sequences.

4.1.2 Support vector machines

The support vector machines will be described in the simplest possible way following the original description (Cortes and Vapnik, 1995). We will assume only binary classifier for classes $y = -1, 1$ and linearly separable training set $\{(\mathbf{x}_i, y_i)\}$, i.e. the training examples can be separated by a line defined by vector $\mathbf{w}$ and transition b. It means that the conditions (4.1) are met.

$$\begin{array}{rclcrcr} \mathbf{w} \cdot \mathbf{x}_i + b & \leq & -1 & \quad \text{if} & y_i = & -1 \\ \mathbf{w} \cdot \mathbf{x}_i + b & \geq & 1 & \quad \text{if} & y_i = & 1 \end{array} \tag{4.1}$$

Thanks to the choice of y labels, we can rewrite the conditions (4.1) in one equation (4.2) that covers all objects in the training set.

$$y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{4.2}$$

SVM are based on the search of the optimal hyperplane (4.3) that separates both classes with the maximal margin. We need to measure the distance between the classes in the direction given by $\mathbf{w}$. The formula for this is (4.4).

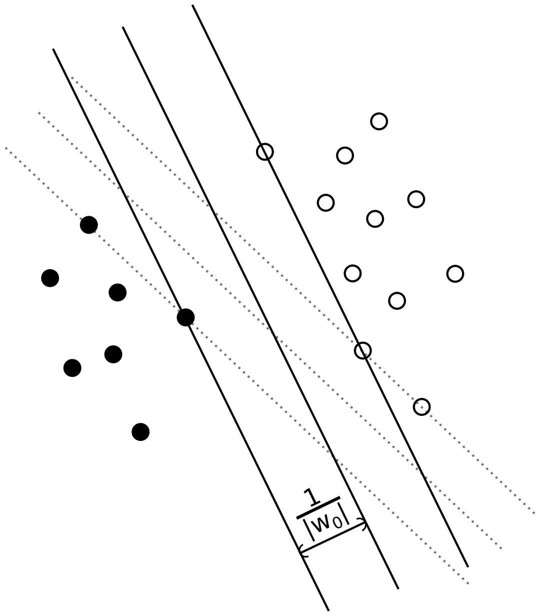


Figure 4.2: Optimal (and suboptimal) hyperplane.

$$\mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0 \tag{4.3}$$

$$d(\mathbf{w}, b) = \min_{x;y=1} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} - \max_{x;y=-1} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} \tag{4.4}$$

The optimal hyperplane maximizes the distance $d(\mathbf{w}, b)$ and can be expressed as (4.5). Therefore the parameters $\mathbf{w}_0$ and b_0 can be found by maximizing $|\mathbf{w}_0|$. For better understanding the optimal hyperplane (and also one suboptimal) is shown on Figure 4.2.

$$d(\mathbf{w}_0, b_0) = \frac{2}{|\mathbf{w}_0|} \tag{4.5}$$

The classification is then done by looking on which side of the hyperplane the object is. Mathematically written as (4.6).

$$l(\mathbf{x}) = \mathrm{sign}(\mathbf{w}_0 \cdot \mathbf{x} + b_0) \tag{4.6}$$

The best presented result for Czech NER was achieved with SVM (Kravalová and Žabokrtský, 2009). SVM are also often used in systems which combine multiple classifiers (Ekbal and Saha), because they are able to generalize very well and the principle is quite different from other methods.

4.1.3 Maximum entropy classifier

The most uncertain probability distribution is the uniform one, because then everything has the same probability. If some constraints are added to the model, the model has to be modified to satisfy these constraints, but there is infinite number of probability distributions satisfying them. The principle of maximum entropy (Guiasu and Shenitzer, 1985) says that the best distribution is the most uncertain one subject to the constraints. A constraint is given in the following form.

$$E_{\tilde{p}}(f_i) = E_p(f_i) \tag{4.7}$$

Where $E_{\tilde{p}}(f_i)$ is the expected value of feature f_i observed from data and $E_p(f_i)$ is the expected value of maximum entropy model. The features are in the following form.

$$f(x,y) = \begin{cases} 1 & \text{if } y \text{ is PERSON and } x \text{ starts with capital letter} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Where parameter y is a class of a NE and x is the classified object, in our case word (or lemma). It is not necessary to have only binary features, but all feature values have to be positive.

For named entity recognition we want to find conditional probability distribution $p(y|x)$, where y is class of word and x are words used for classification. Following the principle of maximum entropy we want $p(y|x)$ to have maximum entropy H of all possible distributions.

$$\arg\max_{p(y|x)} H(p(y|x)) = -\sum_{x \in \Omega} p(x) \sum_{y \in \Psi} p(y|x) \log p(y|x)$$

Because $H(p(y|x))$ is a concave function, there is only one maximum. It can be shown by Lagrange method (Berger et al., 1996) that the best probability distribution has the following parametric form.

$$p(y|x) = \frac{1}{Z(x)} \exp \sum_{i=1}^{n} \lambda_j f_j(x, y) \tag{4.9}$$

$$Z(x) = \sum_{y} \exp \sum_{i} \lambda_i f_i(x, y)$$

$Z(x)$ is only a normalizing factor which ensures that $p(y|x)$ is a probability distribution. $\Lambda = \{\lambda_0, \ldots, \lambda_n\}$ are parameters and have to be set properly to gain maximum entropy. The parameters are found using generalized iterative scaling (Darroch and Ratcliff, 1972), improved iterative scaling (Berger et al., 1996), limited memory BFGS (Liu and Nocedal, 1989; Nocedal, 1980) or other minimization method (Malouf, 2002).

ME was one of the most popular and successful methods. On CoNLL 2003 five of 16 systems used ME (Tjong Kim Sang and De Meulder, 2003). A typical pure ME classifier is presented in (Chieu and Ng, 2003).

4.1.4 Maximum entropy Markov models

A maximum entropy Markov model (McCallum et al., 2000) is a combination of maximum entropy and Markov models. The motivation is to get the best of both methods. The HMM's ability to find sequences and ME's ability to use a lot of diverse features. In other words, we want to model probability distribution $p(y|x, y')$ using the maximum entropy principle. This can be achieved by splitting the problem into probability distributions $p_{y'}(x|y)$, creating one ME classifier (4.10) for each previous class.

$$p_{y'}(x|y) = \frac{1}{Z_{y'}(x)} \exp \sum_{i=1}^{n} \lambda_j f_j(x, y) \tag{4.10}$$

The transformation of dependencies can be seen on Figure 4.3. The black points are observations and the white ones are states or labels. The HMM uses a generative approach and models two distributions for state transition and observation emission. The ME uses a discriminative approach, but cannot exploit the Viterbi or Baum-Welch algorithm. The MEMM is a discriminative method that can use altered Viterbi and Baum-Welch method.

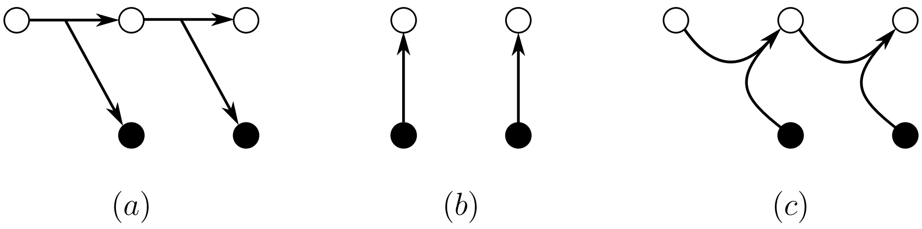


Figure 4.3: Dependency graphs for (a) HMM, (b) ME nad (c) MEMM.

So far, MEMM seems to have only advantages. There is also one very important disadvantage that is data sparseness. While in ME data are used to train one classifier, in MEMM the data has to be split and used for $|y|$ classifiers, where $|y|$ is number of labels.

4.1.5 Conditional random fields

Conditional Random Fields (CRF) were introduced in (Lafferty et al., 2001b). The idea of CRF is strongly based on ME. The difference is that ME classifies one instance after another while CRF classify the whole sequence at once. Mathematically written, ME estimates $p(y_i|x_i)$ for $i = 1, \ldots, n$ and CRF estimate $p(\mathbf{y}|\mathbf{x})$ using (4.11) where $\mathbf{y}$ and $\mathbf{x}$ are n-dimensional vectors. The probability $p(\mathbf{y}|\mathbf{x})$ can be computed using matrices and a variant of forward-backward algorithm.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\,(\sum_j \sum_i \lambda_i f_i(y_j, y_{j-1}, \mathbf{x}, j)) \tag{4.11}$$

The features are extended and can use the previous state in contrast to ME. Two types of features are used, state s and transition t. The state features can be considered as a subset of transition features, where the previous state is not used. We can define general features f as union of transition and state features (4.12).

$$\{f(y_{j-1}, y_j, \mathbf{x}, j)\} = \{s(y_j, \mathbf{x}, j)\} \cup \{t(y_{j-1}, y_j, \mathbf{x}, j)\} \tag{4.12}$$

Following the dependency graphs from Figure 4.3 we can see the dependency graph for CRF on Figure 4.4. The parameters of this model are found using similar methods like for ME, e.g. L-BFGS.

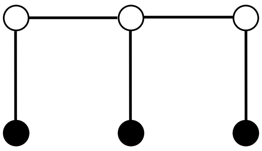


Figure 4.4: Dependency graph for CRF.

Initial tests on the NER task was done in (McCallum and Li, 2003). Since their introduction, many systems used them with very good results (Konkol and Konopík, 2013; Georgiev et al., 2009; Benajiba et al., 2008). CRF are considered to be the most successful method for NER.

4.2 Features

If a machine learning approach to NER is used, the features are something like the senses for human. That is why choosing the right feature set has the highest importance. From the beginning of the NER task various features have been used. We follow with an introduction of terminology used for features.

From the machine learning point of view, the features can be divided into binary, categorical, ordinal, real-valued, etc. Some algorithms may have restrictions for features, e.g. only binary features can be used.

The features can be divided by the context they use. Commonly, two categories are used, *local* and *global*. Local features use only a small neighborhood of the currently processed word (e.g. two preceding and two succeeding words) often called *window*, while global features use the whole document, sentence, or corpus.

The terms *character-level* (Collier and Takeuchi, 2004) and *word-level* features (Nadeau and Sekine, 2007) are used in an ambiguous way. In some papers both are used for features, that are based on characters of a single word (e.g. affixes). In this thesis, we use the term character-level features for this purpose and the term word-level features for features based on individual words. With this terminology choice, the character n-grams are character-level features and word n-grams are word-level features, which is logical in our opinion.

The term *external* features is used to indicate a feature which uses an external system or information source (e.g. Wikipedia). There is a special subclass of external features called *dictionary* or *list-lookup* features. This group is often handled separately, because it plays (or played) an important role in NER.

Another important property of the feature is a language dependency. The feature is *language independent*, if it can be used for another language without any changes. Language independent NER systems obviously need to use only language independent features.

4.2.1 Character-level features

Orthographic features

Orthographic features are based on the appearance of the word, e.g. the first letter is a capital letter, all letters are capital or the words consists of digits. These features are used very often (Tjong Kim Sang and De Meulder, 2003), because they need only the word, are language independent and still very effective for many languages.

Orthographic patterns

Orthographic patterns or word shapes (Collins, 2002b; Ciaramita and Altun, 2005a) are features based on rewriting symbols of the word based on their type, i.e. upper case letters are rewritten to ‘A’, lower case to ‘a’, etc. Sometimes a compressed version is used, where multiple characters of the same type (‘aaaa’) are rewritten to a shortened form (‘a*’ or ‘aa’), which has the meaning of multiple characters of this type.

Affixes

Another language independent feature are affixes (or more precisely fixed width beginnings and endings of a word). Some types of NEs often share the same word ending or prefix. Only a small portion of affixes are meaningful and thus some kind of threshold or feature selection is needed to choose a reasonable number of affixes.

Character n-grams

Character n-grams are generalization of affixes. A bag of n-grams paradigm can be used (discarding their position) or the n-gram can be used together with its position.

4.2.2 Word-level features

Word

A word itself can be used as a feature. In many cases all letters are converted to upper or lower case to capture a word at the start of a sentence as the same feature as in the middle (Kozareva et al., 2007).

Stems and lemmas

Stemming is a task which is trying to find the normalized form of each word, usually by removing semantically unnecessary ending characters. A stem can be used directly as a feature similarly to a simple word feature. Stemming can also improve the performance of other features, e.g. gazetteers.

Lemmatizetion is a task similar to stemming but the output is a lemma instead of a stem. Lemma is basic word form of a word often used as a dictionary entry.

The importance of stemming and lemmatization is influenced by the language. For highly inflectional languages like Czech, stemming or lemmatization is almost a must because it is necessary to reduce the high number of different word forms (Konkol and Konopík, 2014).

Word n-grams

Word n-grams (n-tuples of consecutive words) are used to capture the context of the current word more precisely than single words.

Part of speech and morphology

Morphological tags are a useful feature. Generally, NEs are most often nouns, adjectives and numbers. Other types like prepositions appear less frequently and some like verbs are rare. In inflective languages, morphological tags also give us a possibility to detect consecutive words in the same case which can improve the NE detection.

Patterns

Various types of patterns have been used in NER. The rule-based systems are based on patterns, but machine learning methods can also exploit patterns as features. Patterns can be extracted automatically (Saha et al., 2008; Talukdar et al., 2006). The use of more complex hand-made rules in machine learning systems tends to have a negative impact on the adaptability.

An interesting approach was presented in (Carreras et al., 2003b). One of six categories is assigned for each word. Each category is represented by one character. The pattern is then a string of the category characters.

4.2.3 Global features

Previous appearance

Some authors use a previous appearance of an NE in the document. If NE is already marked in the text, new appearance of the same NE will be probably NE with the same class.

Sometimes this is not used as a feature but as a postprocessing step. After the classification step, all the found NEs are reviewed and the NE classes can be changed if they appeared in other class with higher probability.

Finkel et al. (2005) introduce a new graphical model based on CRFs, which automatically increases the probability that the same words share the same class.

Meta information

For some documents meta information is available. Good example of documents that can include meta information are news articles or emails. When dealing with news articles (or emails) it can be useful to use category of that article, its title (or subject) etc. However, usage of meta information has negative impact on adaptability, because the meta information will not be available or will be different for other domain or data source.

4.2.4 List-lookup features

Gazetteers

Gazetteers are lists of NEs. Systems with gazetteers are obviously loosing the possibility of direct use for another languages or even domains (Mikheev et al., 1999). There were attempts to mitigate this problem by unsupervised or semi-supervised gazetteer creation.

Trigger words

Trigger words are words which are not NEs, but are often in the neighbourhood of NEs. For example 'president' can be a trigger word for a person. A list of trigger words for each entity type is used as a feature. These lists can be created automatically from the training corpus.

4.2.5 External features

Wikipedia

Wikipedia is a rich source on information, it is thus natural that some authors try to exploit it in NER. In (Kazama and Torisawa, 2007; Richman et al., 2008) the authors use the categories of some word sequences as a feature for NER. The results are obviously dependent on the language of Wikipedia, because English has many times more articles than other languages.

Wikipedia have been used in another ways. A corpus for NER was automatically created from Wikipedia (Nothman et al., 2012). Also it can be exploited in automatic gazetteer creation (Toral and Munoz, 2006).

Semantic features

Semantic features are based on semantic similarity of words. The basic idea of these features is simple. If we know that a word "president" is usually followed by a person name, then the word "king" which is semantically similar to "president" should be also followed by a person name. Of course, this idea can be extended in many ways.

Semantic features (and the methods for semantic similarity) are currently a hot topic. Ratinov and Roth (2009) used clusters created using the Brown algorithm (Brown et al., 1992). Turian et al. (2010) tested three different methods: Brown clustering (Brown et al., 1992), C&W embeddings (Collobert and Weston, 2008), and HLBL embeddings (Mnih and Hinton, 2007). Tkachenko and Simanovsky (2012) tested Brown clusters, Clark clusters, and LDA clusters. Brown clusters were used by Straková et al. (2013).

Some of the methods for distributional semantics (important in this thesis) are described in Chapter 5.

4.3 Segment representations

Many entities consist of multiple words (e.g. *Golan Heights*). If we use (standard) machine learning approach to NER, it is necessary to assign exactly one class to each token (word) in the corpus. The simplest way is to have one class for each type of named entity (and one extra type for normal words). This solution has a major limitation – it is not possible to correctly encode subsequent entities of the same type, e.g *"... the Golan Heights Israel captured from ..."* from CoNLL-2003 dataset where *Golan Heights* and *Israel* are both the *location* type. The result would look like this *"... word Location Location Location word word ..."* and it is impossible to decide, where the first entity ends and the second starts. Another motivation for more complex segment representations is that they can increase recognition performance. For example, the recognition rules may differ for the first word and subsequent words of an entity. A segment representation that distinguishes the beginning of an entity then may help with the recognition. The idea can be further extended by more complex segment representations.

There are multiple models for representing multi-word named entities (or more generally multi-word expressions). All the models (except the simplest one) use more than one class for each type of named entity, e.g B-PERSON, I-PERSON for PERSON named entity. To our best knowledge, the most complex model uses 4 classes for each entity (plus one for not-an-entity class). As already shown in the example, the classes are usually distinguished by a single letter prefix. The prefixes have a meaning of relative position in the named entity. The following list summarizes commonly used prefixes.

B (Beginning) Represents the first word of the entity.

I (Inside) Represents a part of the entity, which is not represented by other prefix.

L (Last, sometimes also **E**nd) Represents last word of the entity.

O (Outside or other) Represents word that is not a part of the entity.

U (Unit, sometimes also **W**ord or **S**ingle token) Represents a single word entities.

As we have said earlier, these models have two major purposes. The first one is to distinguish two subsequent entities. The model is able to do that, if it uses at least the **O**utside, **I**nside and one of the **B**egin and **E**nd classes. The second one, is to improve performance. Each class represents a different set of statistics that can be used in the decision process. The intuition tells us, that the statistics accumulated over the corpus may be different for the first word of the entity (**B**), the inside word (**I**) and the other cases. For example the first word of the entity has much higher probability of having the first letter uppercase in Czech. The following models are used in NER.

IO model is the name we use for the simplest representation, even though this model has no well-known or widely accepted name. Each entity is represented only by one class, which obviously does not need any prefix. This model is unable to decode subsequent entities of the same type, but it is not as important as it may seem at first sight, because subsequent entities of the same type are rare.

BIO model (or IOB) representation decodes each entity with two classes. There are two versions of the representation. The BIO-2 uses the **B**egin class for each first word of an entity. The BIO-1 uses the **B**egin class for the first word, only if it follows entity of the same type. In other words, the BIO-1 uses the **B**egin class only if it has to distinguish subsequent entities.

BIEO model (BIOE, OBIE) representation uses both **B**egin and **E**nd classes.

BILOU model (C+O) representation is the most complex model used in NER. It adds the **U**nit class for single word entities.

The simplest segment representation (IO) was used by some of the first ML systems, e.g. (Bikel et al., 1997; Collins and Singer, 1999; Béchet et al., 2000).

The CoNLL-2002 and CoNLL-2003 shared tasks used the BIO representation for annotations in their corpora (BIO-1 in 2002, BIO-2 in 2003) and many authors have adopted this model in their NER systems. The BIO model is the most commonly used model since these conferences.

The BIEO model was used in few papers (Cucerzan and Yarowsky, 2002; Mao et al., 2007; Sun et al., 2010), but it is very rare compared to the BIO model.

Some of the recent papers (Liu et al., 2011; Ratinov and Roth, 2009; Straková et al., 2013) adopted the BILOU representation probably based on the comparison in (Ratinov and Roth, 2009), where the authors provide a comparison of the BIO and BILOU representations on English CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and MUC-7 corpora using CRFs.

5 Latent semantics

"Do you wish me a good morning,
or mean that it is a good morning
whether I want it or not;
or that you feel good this morning;
or that it is a morning to be good on?"
J.R.R. Tolkien, The Hobbit

We use various methods for modelling latent semantics to improve the quality of our NER system. The basic idea behind these methods is based on distributional hypothesis (Harris, 1954; Firth, 1957) that claims *"a word is characterized by the company it keeps"*. In other words, the meaning of a word can be guessed from contexts in which it often appears. This hypothesis is supported in (Rubenstein and Goodenough, 1965; Charles, 2000), where authors carry out empirical tests on humans.

The computational models (that exploit this hypothesis) try to model the contexts of words based on statistics. The output of these methods are usually high-dimensional vectors each representing the meaning for one word or context. The words represented as vectors form a vector space model. Thanks to the vector representation we can easily compare word meanings using similarities or distances of their vectors.

The methods can be roughly divided based on the context they use into *context-word* and *context-region* methods (Riordan and Jones, 2011; McNamara, 2011). In this paper, we use slightly different notation for the same division – *local context* and *global context*. A good overview of semantic models can be found in (Turney and Pantel, 2010; Riordan and Jones, 2011; McNamara, 2011).

The *local context methods* use only a limited context around the word to infer its vector. This limited context is usually referred to as a context window and contains only a few (e.g. four) words before and after the processed word. We use the following methods for modelling the local context – HAL (Sect. 5.2), COALS (Sect. 5.3), RI (Sect. 5.4), BEAGLE (Sect. 5.5) and P&P (Sect. 5.6). These methods belong to a large group of algorithms known as *semantic spaces*. Later in this paper, we use the term semantic spaces as a reference to these models.

The global context methods use a much wider context, usually the whole section or document. The most prominent global context methods are LSA (Latent Semantic Analysis) (Deerwester et al., 1990), PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999) and LDA (Latent Dirichlet Allocation) (Sect. 5.1). In this paper, we use only LDA, which belongs to the current state-of-the-art models for global semantics.

The local and global context methods usually discover different kinds of relations between words. For the local context approaches, the most similar words to word *hockey* can be *tennis*, *football*, or *baseball*. For the global context approaches, these can be *puck*, *player*, or *stadium*.

In the following subsections we introduce models we use.

5.1 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is a topic model. It is a generative graphical model which represents the document as a mixture of abstract topics where each topic is a mixture of words.

Plate notation of LDA is shown in Figure 5.1. The nodes in this figure represent random variables. A random variable $\theta_{D_i} \sim Dirichlet(\alpha)$ represent probabilities of topics for document D_i. The variable $\phi_k \sim Dirichlet(\beta)$ represents probabilities of words in topic k. The nodes α and β are parameters of the Dirichlet distributions. The variable $z_i \sim Multinomial(\theta_{D_i})$ represents an abstract topic for position i in the document D_i. The variable $w_i \sim Multinomial(\phi_{z_i})$ is a word for position i in the document. For the inference of the model we use Gibbs Sampling as described in (Griffiths and Steyvers, 2004).

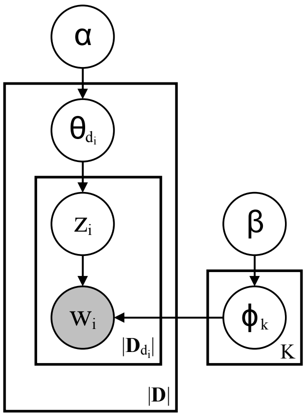


Figure 5.1: Graphical model representation of LDA.

5.2 HAL

Hyperspace analogue to language (HAL) (Lund and Burgess, 1996; Burgess and Lund, 1997) models the similarities between words by collecting statistics about word co-occurrences. The HAL model uses two important assumptions. The first assumption is that the left context and the right context of a word contains different information and that it is important to keep their statistics separate. The second assumption is that the distance between words (in a sentence) is important and more distant words are less informative.

These assumptions are used in a creation of a co-occurrence matrix M. The size of the matrix is $|W| \times |W|$, where $|W|$ is the number of unique words in the corpus. The cell $m_{i,j}$ contains the level of co-occurrence for words w_i and w_j, more precisely for word w_j being in left context of w_i and w_i being in right context of w_j. The value $m_{i,j}$ is incremented all the times word w_j appears in the left context of w_i and the increment is weighted by the distance. If the distance between words exceeds some threshold then the word is not counted as co-occurring any more. More details about creation of the matrix can be found in (Lund and Burgess, 1996). Even though there is not a full information about word ordering, the model still exploits this information partially by incorporating distance weighting and side dependency of context. It is obvious that many words do not occur together so the matrix is very sparse.

The dimensionality of the matrix can be reduced using entropy. The words which are the most uniformly distributed over all other words (have the highest entropy) can be removed.

5.3 COALS

Correlated occurrence analogue to lexical semantic (or COALS) (Rohde et al., 2004) is based on the combination of ideas from HAL and LSA.

The first phase of the model training is the creation of the co-occurrence matrix similarly to HAL. The difference to HAL is that it does not distinguish between left and right contexts. The co-occurrence is counted on both sides of the word and the matrix becomes symmetric. After gathering all statistics the matrix is normalized by correlation. Subsequently, all negative values are replaced by zeros and square-roots of positive values are used.

The second phase is based on LSA. Singular value decomposition is used on the matrix. This has two desired effects. The dimensionality can be rapidly reduced. The assumption is that the reduction should combine similar words together and reveal latent semantic, i.e. transitive relations between words. The second phase can be skipped for some uses.

5.4 Random indexing

Random indexing (RI) (Sahlgren, 2005) is a version of HAL, where the dimension is reduced from the beginning using the theory of random projections.

In HAL, every time a word appears in the context we add a vector with a single non-zero element (which represents the word) to the co-occurrence matrix. The length of the vector corresponds to the number of unique words. In RI, this vector is initialized at the beginning using random projections. The length of the vectors is chosen at the beginning and the elements are generated randomly – most elements are zero and a few elements are non-zero (e.g. -1 and 1). This random initialization causes, that most of the word vectors are still orthogonal, but the dimension is much lower.

An extension of the random indexing method is introduced in (Sahlgren et al., 2008). It allows RI to take word order information into account. It is inspired by the BEAGLE method (subsection 5.5), but it uses permutation of vector coordinates instead of convolution, because of the lower computational cost.

5.5 BEAGLE

Bound encoding of the aggregate language environment (BEAGLE) (Jones and Mewhort, 2007) is a model similar to random indexing (subsection 5.4).

The first phase also generates an index vector with high dimension for each word. The main difference is that the values are taken from Gaussian distribution. The mean value of the Gaussian distribution is set to 0 and the variance to $1/D$, where D is the dimension (usually $D = 1024$).

The final context vector is given by a combination of co-occurrence information and word order information. The co-occurrence information is gathered in a similar way to random indexing, i.e. summing index vectors of co-occurring words. The word order information is a vector given by a convolution of index vectors for all n-grams containing the currently processed word. The context vector is then given as a combination of both information sources.

5.6 Purandare and Pedersen

The Purandare and Pedersen (P&P) (Purandare and Pedersen, 2004) is another type of model. The model works in two phases.

The first phase is a feature selection. The model uses two types of features – words and bi-grams. In the first phase, the training data are used to select only words and bi-grams which are statistically significant. Only a small context (e.g. five words) is used for this purpose. The statistically insignificant features (e.g. “the”) are ignored.

The second phase creates context vectors. It goes through the data again, but now uses a longer context (e.g. 20 words). Only the previously selected words are used. There is an assumption that these contexts represent different meanings of the word. Contexts are then clustered and each cluster should represent one meaning. The final context vector is given by a combination of these clustered vectors.

6 Latent semantics in NER

"Man - a being in search of meaning."
Plato

In this chapter, we describe two of our attempts to incorporate semantic information into our NER system.

6.1 Word similarity

This section is based on our first paper (Konkol and Konopík, 2011). There were two main goals of this paper. First, we wanted to create a good NER system comparable to the Czech state of the art, that can be used for future experiments. Second, we proposed new features which exploited semantic information.

6.1.1 Proposed features

The proposed features are based on the assumption, that words more similar to known NEs have higher probability of being NEs. The similarity is in this experiment modelled using the COALS (Section 5.3) semantic model and cosine similarity.

There are multiple ways how to implement features given our assumption. We have experimented with the following features. All of them use only the most frequent NEs, because of problems with sparsity. In the following paragraph, we denote $f(x)$ the feature for word (or context) x in the data, G the list of entities, and w the word on this list.

- *Bigger groups* – we have manually created lists of about 20 most frequent words for each NE category y. The feature is then the average similarity of the classified word to the words on the list.

$$f(x) = \frac{\sum_{w \in G_y} similarity(x, w)}{|G_y|} \qquad \forall y \tag{6.1}$$

- *Smaller groups* – we have manually created about 20 smaller groups G_i with 2-4 words. One group was for example Prague, Pilsen and Brno, which are largest Czech cities. The feature is again average similarity to these words.

$$f(x) = \frac{\sum_{w \in G_i} similarity(x, w)}{|G_i|} \qquad \forall i \tag{6.2}$$

- *Single words* – the last feature uses similarity to individual words. Each feature is a similarity to a particular word. This feature is very computationally demanding.

$$f(x) = similarity(x, w) \qquad \forall w \in V \tag{6.3}$$

6.1.2 Experimental setup

We use Maximum Entropy classifier in this experiment. The baseline feature set consists of commonly used NER features, namely lemmas, morphological features, word shape features, gazetteers and lists of words related to NEs learned during training. The classifier implementation is provided by the Brainy library (Konkol, 2014).

The COALS semantic model used a dimension reduced to 1000. We use the S-Space library implementation of COALS (Jurgens and Stevens, 2010).

The experiments are evaluated on the Czech Named Entity Corpus 1.0 (Section 3.6). We use 10-fold cross-validation. The precision, recall and F-measure is used, but in a word-by-word version (Konkol and Konopík, 2013).

6.1.3 Discussion

The results are shown in Table 6.1. We have outperformed the previously published Czech NER systems as shown by Konkol and Konopík (2013), who

	precision	recall	F-measure
Baseline	76.78	69.58	72.94
Big groups	76.89	69.71	73.08
Small groups	76.84	69.18	72.76
Single words	76.61	69.30	72.72

Table 6.1: A comparison of previous results with our experiments.

compared the systems with different evaluation measures. The first goal was successfully accomplished.

Unfortunately, the proposed features did not improve the results or the improvement was insignificant, which means that the second goal was not fulfilled. Later, a bug was found in the COALS library, which could negatively influence the results. We plan to revisit this approach together with more alternative approaches of incorporating semantic information to NER.

6.2 Word clusters

Our second attempt to incorporate semantic features to NER was published in (Konkol et al., 2015). In this paper, we experiment with multiple distributional semantics methods (Section 5). The paper has multiple goals. First, design a language independent, highly adaptable system. Second, compare different semantic models. Third, experiment with combinations of different models. Four, explore the effects on unsupervised stemming method in the context of semantic features.

The experiments were performed on four languages (English, Spanish, Dutch, and Czech) in order to prove, that the system is highly adaptable.

6.2.1 Method

In this paper, we use the CRF (Section 4.1.5). The implementation is provided by the Brainy library (Konkol, 2014). The baseline feature set consists of *word*, *bag of words*, *n-grams*, *orthographic features*, *orthographic patterns*, and *affixes*. We intentionally exclude features like gazetteers and PoS tags, which limit the system adaptability.

6.2.2 Proposed features

Stemming features

We use the High Precision Stemmer[1] (HPS) (Brychcín and Konopík, 2015) for our experiments. The HPS is an unsupervised stemmer. The main idea is that the same stems should share the same semantic information.

The HPS works in two steps. In the first step, lexically similar words are clustered using maximal mutual information clustering (Brown et al., 1992). The word similarity is based on the longest common prefix. The output of this phase are clusters which share a common prefix and have a high mutual information. The method assumes, that the common prefix is stem and the rest is a suffix.

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

The HPS is freely available. The trained models for all tested (and many more) languages are provided with the stemmer.

Stemming features are identical to the word features, but use the stem instead of using directly the word found in the text. We use stems for the following features: *stem*, *bag of stems*, *stem n-grams*.

Latent Dirichlet allocation

We incorporate LDA (see section 5.1) in a special way, where we use the probability of a topic z_i directly as a feature for the classifier. There are two options: smoothed and unsmoothed version. The unsmoothed version assigns a probability to a topic based on the histogram of topics sampled for the document, i.e. if some topic was not sampled for the document, then its probability is 0. The smoothed version changes the probability distribution in

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

a way, that all topics have a small probability. Our preliminary experiments showed that both versions have almost the same results, but smoothed version slows the classifier training significantly, because the feature vector is not then sparse. Thus the unsmoothed version is used in our experiments.

We also experiment with LDA preprocessed by stemming. In this case we simply use stems instead of words as an input of training. We denote this version as S-LDA. The motivation of the S-LDA model is that the topic of the document is mostly influenced by the semantic information of a word, and we assume that this semantic information should be the same for words with the same stem. We also assume that the use of stems instead of words reduces the data sparsity problem and leads to a better trained model.

Semantic spaces features

The semantic spaces are incorporated as clusters, i.e. we use the high-dimensional vector representation of words provided by semantic spaces as an input for a clustering algorithm. The clustering is very computationally intensive thus the choice of a good algorithm has major importance. A top down method is used, i.e. starting with one cluster and dividing it, because the desired number of clusters is relatively small compared to the number of words. The number of division operations of top down (partitioning) methods is much smaller then the number of joining operations for bottom up (hierarchical) methods. The partitioning method itself is still not enough to solve our problem. An approximative clustering method have to be used. We use an implementation of repeated bisection algorithm (Zhao and Karypis, 2002) from the CLUTO library (Karypis, 2003), that has been already used in language modelling (Brychcín and Konopík, 2014).

The clusters allow us to represent each word in a $[-2, 2]$ window by a vector $\mathbf{v}$ with dimension C, where C is the number of clusters. For each word we find the corresponding cluster i and set the value v_i to 1. All the other values remains 0.

6.2.3 Evaluation

For comparing systems on multiple languages we have defined an overall score (6.4) as the harmonic mean of the F-measures for the individual languages.

We use the harmonic mean (and not the standard average) because we prefer systems with consistent results across languages.

$$Overall = \frac{1}{\frac{1}{4} * (\frac{1}{F_{en}} + \frac{1}{F_{es}} + \frac{1}{F_{nl}} + \frac{1}{F_{cz}})} \tag{6.4}$$

All the presented results were acquired on the test part of the corpora. The Pearson correlation between all the results on the validation data and all the results on the test data is higher than 0.97, so the test data results should give a very good idea of the validation data.

In the following sections we will firstly introduce all the corpora used in this work. We will follow with a description of our experiments. The last section will discuss the results of all the experiments.

6.2.4 Corpora

In this paper, we use English, Spanish and Dutch CoNLL corpora. All the corpora (Tjong Kim Sang, 2002) are in the same format and have similar sizes: approximately 250,000 tokens. The NEs are classified into four categories: persons, organizations, locations, and miscellaneous. Multi-word entities are encoded using the BIO format.

Additional resources were provided with these corpora. Part of speech tags are available for all tree corpora that we used. Chunk tags were provided for English. Gazetteers were provided for English and Dutch. This information is not used in our system, so as to preserve its full language independence.

For Czech, we used the CoNLL format version (Konkol and Konopík, 2013) of the Czech Named Entity corpus (Ševčíková et al., 2007). It contains approximately 150,000 tokens and uses 7 classes of Named Entities: time, geography, person, address, media, institution, and other.

To train the LDA and semantic spaces, larger unlabeled corpora are needed. For English, we used the Reuters corpus RCV1[2]; for Spanish, the Reuters corpus RCV2[2]; for Dutch, the Twente News Corpus[3]; and for Czech,

[2] http://trec.nist.gov/data/reuters/reuters.html

[3] http://hmi.ewi.utwente.nl/TwNC

	English	Spanish	Dutch	Czech	Overall
Baseline (Words)	84.19	79.86	76.19	68.21	76.65
Baseline (Stems)	84.00	79.73	77.09	69.29	77.14
Baseline (Words + Stems)	84.80	79.71	77.18	69.25	77.32

Table 6.2: Results (in F-measure) for baseline and stem features.

the Czech Press Agency corpus. Corpora with approximately 80 million tokens were used for all languages to ensure similar conditions.

6.2.5 Baseline and stem features

We use the system briefly described in Section 6.2.1 as a baseline. We then changed the feature set by incorporating stems. In the first experiment, we used stems instead of words in the features *word*, *bag of words*, and *n-grams*. In the second experiment, we used the original features based on words and added the features based on stems. The results of these experiments are shown in Table 6.2. The results show that using both words and stems yields the best overall performance. It is thus used as the starting point for the subsequent experiments.

6.2.6 Semantic spaces

We test five different semantic spaces: BEAGLE, COALS, HAL, PP and RI. For each semantic space, we try different numbers of clusters: 100, 500, 1000 and 5000. The numbers of clusters were chosen to scale approximately logarithmically. All the tests are carried out both with and without stemming. The results are shown in Table 6.3.

All semantic spaces are implemented in the S-Space library (Jurgens and Stevens, 2010). For all the semantic spaces, we used the parameters recommended by their authors. For HAL, COALS and RI, we used a context window size equal to 4 in both directions, for P&P and BEAGLE it is 5. The co-occurrence matrix created by HAL has 50,000 columns, for COALS 14,000. The reduction using singular value decomposition was not used for COALS, based on the experience of (Brychcín and Konopík, 2014). RI uses vectors with dimension 1,024. The dimension D for BEAGLE was set to

(a) English (baseline + stem 84.80)

# clusters		100	500	1000	5000
Words	BEAGLE	86.16	86.46	87.09	86.24
	COALS	86.15	85.67	85.68	85.30
	HAL	**87.82**	87.29	87.02	85.75
	PP	84.24	84.49	84.90	84.38
	RI	86.24	86.11	86.19	84.61
Stems	BEAGLE	85.29	85.73	86.09	86.21
	COALS	85.69	85.67	85.34	85.32
	HAL	86.52	**86.57**	86.27	86.02
	PP	84.84	84.97	84.97	84.89
	RI	85.89	85.89	85.93	85.70

(b) Spanish (baseline + stem 79.71)

# clusters		100	500	1000	5000
Words	BEAGLE	81.11	81.11	80.72	80.55
	COALS	80.99	80.49	80.14	79.87
	HAL	**81.70**	81.15	80.93	81.08
	PP	80.02	80.20	79.95	80.10
	RI	80.59	80.49	80.63	80.28
Stems	BEAGLE	80.34	80.75	80.20	79.87
	COALS	80.63	80.66	80.10	79.74
	HAL	**81.20**	80.69	80.47	80.24
	PP	79.84	79.66	79.57	79.64
	RI	80.33	80.05	80.41	80.40

(c) Dutch (baseline + stem 77.18)

# clusters		100	500	1000	5000
Words	BEAGLE	79.28	79.23	79.43	79.16
	COALS	80.57	78.96	78.63	77.75
	HAL	**80.72**	80.62	79.53	78.44
	PP	77.41	77.46	77.46	77.05
	RI	79.44	78.51	77.87	78.51
Stems	BEAGLE	77.63	78.20	78.49	78.08
	COALS	**79.39**	79.27	78.68	77.74
	HAL	78.80	78.97	78.42	78.18
	PP	76.92	77.14	76.96	77.66
	RI	78.55	78.90	78.18	78.49

(d) Czech (baseline + stem 69.25)

# clusters		100	500	1000	5000
Words	BEAGLE	70.51	70.40	70.34	70.83
	COALS	71.56	70.77	70.17	70.12
	HAL	**71.98**	71.92	71.07	71.02
	PP	69.49	69.31	69.49	69.63
	RI	70.50	71.05	70.68	70.14
Stems	BEAGLE	69.66	69.17	70.44	70.17
	COALS	70.00	**70.89**	70.07	70.26
	HAL	70.10	70.38	70.47	69.76
	PP	69.57	69.53	70.14	69.64
	RI	69.57	70.05	69.10	70.16

Table 6.3: Results (in F-measure) for different semantic spaces for (a) English, (b) Spanish, (c) Dutch and (d) Czech. There are results for semantic spaces trained on both words and stems.

1,024, the mean value to 0, and the variance to $1/D$. P&P uses 3 meanings for each word.

6.2.7 Latent Dirichlet allocation

We tested LDA with various numbers of topics. The following numbers of topics were chosen: $\{20, 50, 100, 200, 300, 400, 500\}$.

Each experiment incorporating LDA features was repeated five times, because the vector of topics generated by LDA is a random variable. The results are shown in Table 6.4, where Avg represents the average of all five experiments and σ represents the standard deviation.

# topics	en		es		nl		cz		Overall
	Avg	σ	Avg	σ	Avg	σ	Avg	σ	
20	85.19	0.03	**80.55**	0.07	76.99	0.15	69.80	0.09	77.72
	84.83	0.05	80.54	0.05	77.33	0.04	**70.61**	0.23	**77.98**
50	85.05	0.07	80.50	0.02	77.54	0.08	70.27	0.07	77.96
	84.88	0.18	80.51	0.04	77.18	0.07	70.29	0.10	77.84
100	85.25	0.12	80.33	0.04	77.48	0.02	70.02	0.07	77.87
	85.22	0.02	80.14	0.03	77.22	0.09	70.15	0.04	77.79
200	**85.29**	0.03	80.22	0.06	**77.59**	0.06	69.92	0.05	77.85
	85.11	0.02	79.93	0.05	77.31	0.05	70.53	0.06	77.86
300	85.01	0.02	80.07	0.04	77.17	0.13	69.73	0.08	77.59
	84.96	0.02	79.89	0.07	77.41	0.07	70.14	0.10	77.73
400	84.98	0.02	79.94	0.05	77.37	0.06	69.69	0.03	77.59
	84.79	0.05	79.93	0.03	77.30	0.06	70.44	0.07	77.76
500	85.06	0.06	79.96	0.04	77.20	0.20	70.02	0.09	77.67
	84.88	0.03	79.91	0.04	77.21	0.08	70.02	0.04	77.63
Baseline + Stem	84.80		79.71		77.18		69.25		77.32

Table 6.4: LDA results (in F-measure). The results on the top of the row are for standard LDA. The bottom results are for S-LDA.

We used the LDA implementation from the Mallet library (McCallum, 2002). We used Gibbs sampling with 1,000 iterations for inference. The hyperparameters α and β of the Dirichlet distribution were set according to recommendations given by (Griffiths and Steyvers, 2004). β was set to 0.1 and α to $1/K$, where K is the number of topics.

6.2.8 Combinations

It is intractable to test all possible combinations of our features (as we experiment with 54 single models for each language). In this section, we will describe our procedure for selecting the best performing combinations. In all, we tested approximately 200 combinations. Due to space requirements, we have chosen only the interesting results (Table 6.5).

We started with experiments that combined multiple variations of a single method (various numbers of topics and clusters). The results show that it is always advantageous to use all variations of clusters, resp., topics. Therefore we always use all variations combined in subsequent experiments and denote them with the name of the method, e.g. HAL as the combination of HAL-100, HAL-500, HAL-1000 and HAL-5000. This reduced the number of models to 12 combined models (one for each method) for each language.

The goal of our subsequent experiments was to choose the optimal combination of the proposed features. We chose a different (the best) combination for each language, and one extra combination based on the overall improvement. We used a standard heuristic for choosing the best combination. We started with the baseline + stem feature set and iteratively added more features. In each iteration, new features are evaluated on the validation set and the best feature is added to the resulting feature set. The algorithm stops if the improvement of the best feature is less than or equal to zero. Furthermore, we followed multiple paths if the results were almost equal for two features.

	en	es	nl	cz	Overall
Baseline	84.19	79.86	76.19	68.21	76.65
All word clusters	87.92	**83.08**	81.86	72.34	80.89
All clusters	89.32	82.10	82.04	72.82	81.14
HAL-100	87.82	81.70	80.72	71.98	80.15
HAL	88.62	82.06	81.95	72.74	80.94
LDA-50	85.05	80.50	77.54	70.27	77.96
LDA	85.92	81.52	79.04	70.49	78.83
S-HAL	87.41	81.44	79.89	70.76	79.41
All word clusters + LDA	88.33	**83.08**	82.08	73.11	81.27
All clusters + LDA	**89.44**	82.43	82.21	73.58	81.52
HAL + COALS + S-COALS + LDA	89.18	82.74	**83.01**	**74.08**	**81.89**
(Lin and Wu, 2009)	**90.90**	—	—	—	
(Lin and Wu, 2009) w/o phrase clusters	88.34	—	—	—	
(Florian et al., 2003)	88.76	—	—	—	
(Carreras et al., 2002, 2003a)	85.00	81.39	77.05	—	
(Curran and Clark, 2003)	84.89	—	**79.63**	—	
(Ferrández et al., 2006)	—	**83.37**	—	—	
(Konkol and Konopík, 2013)	83.24	81.39	75.97	**74.08**	

Table 6.5: Results (in F-measure) for combinations of different clusters and LDA models.

6.2.9 Discussion

We will start by discussing the unsupervised stemming. Table 6.2 reveals that adding the stem features is better than replacing word features. Adding stem features improved the results of all languages except Spanish by approximately 1 (absolute improvement in the F-measure). The performance for Spanish was lower, but only by 0.15. The highest improvement (1.04) was achieved for Czech, Dutch being only slightly worse (0.99). The improvement for English was 0.61. The experiment confirmed our expectation, that the improvement would be higher for more inflectional languages, but we expected a higher improvement for highly inflectional Czech compared to weakly inflectional English.

All the subsequent experiments used the baseline + stem feature set and their results are compared with the results of this feature set.

The results of semantic spaces are in Table 6.3. We see that the best performing model is HAL using 100 clusters. It is also evident that all the semantic spaces except PP improved the performance of the baseline + stem. PP was the worst performing model and in some cases produced worse results than the baseline. The results for the stemmed models are not so clear. The best model is not obvious, but we can say that HAL and COALS are the top performing methods.

Table 6.3 also shows the relation between the number of clusters and performance. For word based HAL and COALS, the optimal value is 100 for almost all cases. For their stem based versions, the optimum is unclear, but seems to be between 100 and 500. The optimum for RI and BEAGLE is not obvious. The stem based versions of the models are worse than the word based ones for all languages. We believe this is because the semantic models we use also work with morphological information that is lost at stemming.

The results of our LDA experiments are shown in Table 6.4. It is very important that the deviations between the tests of one model are relatively small. The LDA feature improves the baseline + stem in general. The difference between the word and stem versions of LDA for individual languages reveals that the word version is clearly better for English and Spanish, is indecisive for Dutch, and the stem version is better for Czech. This shows that stemming is more important for highly inflectional languages—an expected behavior. The optimal number of topics is not clearly visible even for individual languages, but for more than 200 topics the performance drops.

The best overall score was surprisingly achieved using the stem based LDA with 20 topics (77.96). Word based LDA with 50 topics has almost equal results (77.96).

We tested approximately 200 combinations of features and some of the results are shown in Table 6.5. As we mentioned earlier, combinations of various sizes (number of clusters, resp., topics) are beneficial and improve the results of single models for both semantic spaces and LDA.

We have improved the results of our NER system by 5.24 in the overall F-measure from 76.65 (baseline) to 81.89. The best result overall was achieved using a combination of models: HAL, COALS, S-COALS and LDA.

6.2.10 Comparison with the state of the art

If we compare our best results with the state-of-the-art publications (Table 6.5), we can see that our system has a very good performance. In comparison with the best English NER system (Lin and Wu, 2009), our results are worse by 1.46. The best English system uses phrase clusters, but we have insufficient data to test phrase clusters with our methods (their corpus has 700bn tokens, while ours has only 89 million). If we compare our system with the results of the best English system without phrase clusters (i.e., compare their word clusters with our clusters) we have improved the results by 0.98 even though they used 1000 times more data. Our approach also outperformed the external knowledge features from (Ratinov and Roth, 2009).

We can also compare our system with the best performing English system from the CoNLL-2003 (Florian et al., 2003) and we outperform it by 0.68. This system used gazetteers and language dependent preprocessing (part-of-speech, chunks, lemmatizer).

We are worse by 0.29 than the best Spanish NER system (Ferrández et al., 2006), but the Spanish system is heavily language-dependent. It is based on a combination of a machine learning and a rule based approach. The machine learning system is used as an input for the rule based part, which made the final decisions.

The best performing CoNLL-2002 Spanish system is outperformed by 1.69. This system also used language-dependent features (part-of-speech, gazetteers).

The best Dutch system (Curran and Clark, 2003) is outperformed by 3.38.

For Czech, we compare our system with the system of (Konkol and Konopík, 2013), which is also a language dependent system (lemmatizer, gazetteers). The results end up to be exactly equal.

7 Segment representations

“Everything we hear is an opinion,
not a fact.
Everything we see is a perspective,
not the truth.”
Marcus Aurelius

This chapter is based on the paper (Konkol and Konopík, 2015). Our goal in this paper was to explore the commonly overlooked aspect of NER, the segment representations. The idea for this paper comes from the paper (Ratinov and Roth, 2009), where the authors provided a comparison of the BIO and BILOU model and concluded that the BILOU model is better the the BIO model. Multiple papers then adopted this model because of these results. In our opinion, the conclusion was only weakly supported by the data. Thus, our goal was to compare multiple segment representations (not just BIO and BILOU) in a statistically sound way.

7.1 Related work

Most of the related work was already mentioned in Section 4.3. In this section, we only mention two more studies, that are relevant to our experiments.

An interesting study concerning segment representations in NER is provided in (Cho et al., 2013). The authors present method for using multiple segment representation together in a single system. They also provide a comparison of multiple segment representations on the biomedical domain. The biomedical domain has different properties than the standard (news) corpora used in NER and cannot be compared with our results.

A similar research (comparison of segment representations) has been done for a different task – text chunking (Shen and Sarkar, 2005). To our best knowledge, there are no other articles comparing segment representations in NER.

7.2 NER system

In this paper, we use two standard machine learning systems. The first one is based on maximum entropy (ME) classifier and follows the description in (Borthwick, 1999). The second one is based on conditional random fields (CRF), similar to the baseline system in (Lin and Wu, 2009). We use the Brainy ML library (Konkol, 2014) for this purpose.

Both methods use the same feature set which consists of common NER features. The features are the following: words, bag of words, n-grams, orthographic features, orthographic patterns, and affixes.

7.3 Corpora

Our experiments are done on four languages – English, Spanish, Dutch and Czech. We use one corpus for each language.

For English, Spanish and Dutch we use the corpora from CoNLL-2002 and CoNLL-2003 shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). These corpora have approximately 300,000 tokens and use four entity types – person (PER), organization (ORG), location (LOC) and miscellaneous (MISC).

For Czech we use the CoNLL format version of Czech Named Entity Corpus 1.1 (Konkol and Konopík, 2013; Ševčíková et al., 2007). This corpus is smaller than the CoNLL corpora and has approximately 150,000 tokens. It uses 7 classes – time (T), geography (G), person (P), address (A), media (M), institution (I) and other (O).

All corpora use the BIO segment representation for the data. The English corpus (CoNLL-2003) uses the BIO-1 representation of segments. The rest the BIO-2. The segment representation of the corpora does not play any role in the training or evaluation as we firstly load the corpora to inner, corpus-independent representation and then transform it into training (or validation or test) data with proper segment representation for the given experiment.

7.4 Experiments

In all the experiments, we use the standard CoNLL evaluation with precision, recall and F-measure. We present only the F-measure because of space requirements. In the following sections we show two sets of experiments. The discussion of our results is in a separate section.

7.4.1 Standard partitioning

The first set of experiments is evaluated on the original partitioning of the corpora – training, validation and test set. For our experiments, we do not need to set any parameters based on the results on validation set. The results on the validation set thus provide the same information as on the test set. This follows the same procedure as in (Ratinov and Roth, 2009).

For each combination (segment representation, ML approach) we train a model on the training data and evaluate it on the validation and test data. The results of these experiments are shown in table 7.1.

7.4.2 Significance tests

The results of the first experiments are in many respects indecisive. For many representation pairs it is impossible to choose the better one (one is better on the test set, the other one on the validation set). Thus, we decided to perform a 10 fold cross-validation to obtain more consistent results computed on much larger data. The advantage is that our tests do not depend on a short portion of data created by manual corpus division.

The data are prepared by the following procedure. Firstly, we concatenate all the data sets for each language (ordered: training, validation, test) into the data set D_{All} and number all the sentences (s denotes the index of a sentence). For fold i, $i = 0, \ldots, 9$, the test set is $D_{Test} = \{s : s \mod 10 = i\}$

	ME		CRF	
	val	test	val	test
IO	86.89	78.66	88.98	83.64
IOU	**87.16**	**79.94**	88.75	83.60
BIO-1	86.89	78.27	88.98	83.61
BIO-2	86.86	79.15	88.08	83.74
BIOU	87.01	79.82	88.92	83.96
IEO-1	86.79	78.54	89.02	83.79
IEO-2	86.99	79.53	**89.25**	**84.16**
IEOU	86.91	79.85	89.10	83.62
BIEO	86.55	78.88	88.90	83.82
BILOU	86.42	79.50	88.84	83.47

(a) English

	ME		CRF	
	val	test	val	test
IO	64.59	70.45	74.02	79.66
IOU	63.98	70.93	73.95	79.33
BIO-1	64.46	70.35	74.38	79.80
BIO-2	63.80	**71.37**	**74.56**	79.54
BIOU	64.04	71.27	74.15	79.18
IEO-1	**65.13**	71.03	74.27	**79.86**
IEO-2	63.12	70.33	74.45	79.50
IEOU	63.39	70.76	74.44	78.96
BIEO	63.30	70.54	74.46	79.55
BILOU	63.42	70.71	74.37	79.37

(b) Spanish

	ME		CRF	
	val	test	val	test
IO	67.25	70.09	74.31	76.34
IOU	69.28	71.85	74.39	76.62
BIO-1	67.49	70.06	**74.81**	76.53
BIO-2	68.31	70.45	74.37	76.23
BIOU	**69.56**	**72.53**	74.59	76.56
IEO-1	68.84	71.07	74.54	76.13
IEO-2	68.49	70.91	74.07	**77.17**
IEOU	69.23	72.34	73.63	76.79
BIEO	68.43	70.68	74.68	76.51
BILOU	68.78	72.08	73.82	76.82

(c) Dutch

	ME		CRF	
	val	test	val	test
IO	56.93	53.48	**68.64**	68.41
IOU	57.26	54.33	68.12	68.05
BIO-1	56.16	53.45	68.50	68.90
BIO-2	56.96	54.99	68.44	69.11
BIOU	58.11	55.86	68.54	**70.26**
IEO-1	56.75	55.42	68.30	69.34
IEO-2	56.98	55.61	68.22	70.08
IEOU	58.21	56.64	67.92	69.55
BIEO	58.40	**57.21**	67.58	69.61
BILOU	**58.60**	56.73	67.41	69.21

(d) Czech

Table 7.1: The results of our experiments on the standard partitioning of corpora.

	IO	IOU	BIO-2	BIO-1	BIOU	IEO-2	IEO-1	IEOU	BIEO	BILOU
IO	====	====	>===	==≥=	==<<	==<=	<<=<	==≤≤	>===	=≥≤≤
IOU	====	====	>===	==>≥	==≤=	==<=	==>=	====	>===	>===
BIO-1	==≤=	==<≤	>=<=	====	==<<	==<=	=<=<	==<<	>===	=≥<<
BIO-2	<===	<===	====	<=>=	<=≤<	<≥<=	<=><	<==≤	>>==	=>=≤
BIOU	==>>	==≥=	>=≥>	==>>	====	===>	==>=	≤===	>=>≥	≥===
IEO-1	>>=>	==<=	>=<>	=>=>	==<=	=><>	====	==<=	>>=≥	≥><=
IEO-2	==>=	==>=	>≤>=	==>=	===<	====	=<><	=≤≥<	>=>=	>===
IEOU	==≥≥	====	>==≥	==>>	≥===	=≥≤>	==>=	====	>≥=≥	>>==
BIEO	<===	<===	<<==	<===	<=<≤	<=<=	<<=≤	<≤=≤	====	≤=<<
BILOU	=≤≥≥	<===	=<=≥	=≤>>	≤===	<===	≤<>=	<<==	≥=>>	====

Table 7.2: The significance tests for various segment representations using ME. Detailed description is provided in Section 7.4.2.

and training set $D_{Train} = D_{All} - D_{Test}$. This procedure assures uniform distribution of sentences.

Each combination (segment representation, ML approach) is then tested on each fold. We compare the different combinations using the paired Student's t-test. The results are shown in Table 7.2 for ME and in Table 7.3 for CRF. We use two confidence levels $\alpha = 0.1, 0.05$. The null hypothesis H_0 is that there is no difference between segment representations. The alternative hypothesis H_1 is that one segment representation is significantly better than the other segment representation. Each cell contains four symbols, one for each language in the order English, Spanish, Dutch, and Czech.

- The symbol $<$ (resp. $>$) means, that the row segment representation is significantly worse (resp. better) than the column representation. The H_0 hypothesis is rejected at both levels $\alpha = 0.05, 0.1$.
- The symbol $\leq$ (resp. $\geq$) means, that the row representation is significantly worse (resp. better) than the column representation. The H_0 hypothesis is rejected at the level $\alpha = 0.1$, but we fail to reject it at the level $\alpha = 0.05$.
- The symbol $=$ is used for representations which are not significantly better or worse. We fail to reject hypothesis H_0.

	IO	IOU	BIO-2	BIO-1	BIOU	IEO-2	IEO-1	IEOU	BIEO	BILOU
IO	====	>>=>	=<=>	≤<<≤	===>	=<<<	<==≤	==<=	=<=≥	>==≥
IOU	<<=<	====	<<=<	<<≤<	≤<≤<	<<<<	<==<	=<<<	=<=<	==<<
BIO-1	≥>>≥	>>≥>	>==>	====	>>=>	===≤	==>=	>>==	><>>	>==≥
BIO-2	=>=<	>>=>	====	<==<	>>==	==<<	<==<	≥>≤<	>===	>===
BIOU	===<	≥>≥>	<<==	<<=<	====	≤<<<	<==<	==<<	=<==	>===
IEO-1	>==≥	>==>	>==>	==<=	>==>	==<≤	====	>=<=	>==>	>==>
IEO-2	=>>>	>>>>	==>>	===≥	≥>>>	====	==>≥	≥>≥>	=<>>	>=>>
IEOU	==>=	=>>>	≤<≥>	<<==	==>>	≤<≤<	<=>=	====	=<>>	==>>
BIEO	=>=≤	=>=>	<===	<><<	=>==	=><<	<==<	=><<	====	=≥==
BILOU	<==≤	==>>	<===	<==≤	<===	<=<<	<==<	==<<	=≤==	====

Table 7.3: The significance tests for various segment representations using CRF. Detailed description is provided in Section 7.4.2.

7.4.3 Discussion

We start our discussion with the comparison to results of Ratinov and Roth (2009). They compared BIO-1 and BILOU representations on the English CoNLL corpus using CRF. Our experiments have similar results. The BIO-1 representation was better on the test set, while the BILOU representation was better on the validation set. The differences slightly favor the BILOU representation, but it is unclear, if it is just a coincidence or the BILOU representation is better. This conclusion is also supported by the fact that in (Ratinov and Roth, 2009), the BILOU representation was better on the test set and worse on the validation set (in our case, better on the validation set, worse on the test set). The rest of the results has similar problems. For many representation pairs, it is impossible to pick the better one.

We were not satisfied with the results of the first set of tests, because it does not compare the representations rigorously. Thus, we proposed another approach for segment representations comparison described in Section 7.4.2. It is based on paired Student's test and gives well-defined comparisons.

The results of the significance tests are much more convincing. On one hand, the results provide evidence, that some segment representations are better than others. On the other hand, we are still unable to decide for many representations pairs, i.e. we must treat them as equal. Given these limitations, we can create a group of representations for each language, which are at the same or better level than all the other representations. These groups are (the bold representation has the highest average F-measure):

English, ME: IOU, BIO-1, IEO-1, **IEO-2**, IEOU

English, CRF: BIO-1, **IEO-1**, IEO-2

Spanish, ME: IOU, BIO-2, BIOU, **IOE-1**, IEOU

Spanish, CRF: BIO-2, IEO-1, **BIEO**

Dutch, ME: BIOU, **IEO-2**, BILOU

Dutch, CRF: BIO-1, **IEO-2**

Czech, ME: BIOU, **IEO-1**, IEOU, BILOU

Czech, CRF: IEO-2

Surprisingly, the IOE representations perform quite good. IOE-2 is even significantly better than the rest for Czech using CRF. The BILOU representation, generally considered as the best choice, performed rather poorly. We can say, that the optimal segment representation depends on both language and algorithm. We also expect it to be dependent on the feature set.

7.5 Conclusion

In this chapter, we provide a rigorous study of segment representations for named entities. We experiment with ten different segment representations on the English, Spanish, Dutch and Czech corpora using two machine learning approaches – maximum entropy and conditional random fields.

We performed two sets of experiments. The first one was based on the standard partitioning of CoNLL corpora. The second one exploited 10 fold cross-validation and evaluation using the paired Student's t-test. The second test provides more accurate results.

Our experiments provide an interesting evidence. The BILOU representation ended up as the worst for English using CRF, even though it was considered better than the commonly used BIO-1 by Ratinov and Roth (2009) and it is generally considered as one of the best representations. The results presented in Ratinov and Roth (2009) were similar to the results of our first set of experiments, but the second set of experiments disproved this hypothesis. The IOE-1 and IOE-2 representations seem to be the best or at least reasonable choice for almost all languages and methods. Surprisingly, these representations have not been used in NER yet.

We show that choosing the optimal segment representation for named entities is a complex problem. The optimal representation depends on the language (corpus), on the approach, and very likely on the feature set. We propose a well-defined procedure for finding the optimal representation.

Thus, the impact of the article is two fold. First, we propose a new procedure for segment representation evaluation. Second, we recommend the use of IOE-1 and IOE-2 as they provide the most promising results in our tests.

In the future, we would like to experiment with multiple feature sets and their relation to optimal segment representation. The relation of the data size and the optimal representation could be also interesting.

8 Morphology in NER

"Better is the enemy of good."
Voltaire

In this section, we describe and discuss our experiments (Konkol and Konopík, 2014) with various approaches for stemming and lemmatization. Stemming and lemmatization proved to be an integral part of NER for highly inflectional languages like Czech. In this study, we compare different approaches from the simplest to very complex and compare they results. The main goal was to find out, if it is worth to use very complex methods (with much higher demands) instead of the simple ones and if there is a big difference between language independent and language dependent methods.

8.1 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.

8.1.1 OpenOffice based lemmatizer

This lemmatizer (proposed by authors of this paper) is inspired by the approach from (Kanis and Skorkovská, 2010). 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.

8.1.2 HPS stemmer

The high precision stemmer (Brychcín and Konopík, 2015) (HPS) is an unsupervised stemmer. It works in two phases. In the first phase, lexically similar words are clustered using MMI clustering (Brown et al., 1992). 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.

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.

8.1.3 HMM tagger

The HMM tagger represents a standard (pure) statistical approach to lemmatization (Kupiec, 1992). Transition probabilities in HMM are estimated using 3-gram Kneser-Ney smoothing (Chen and Goodman, 1998). This approach can be easily reproduced using common machine learning libraries. It is trained on the PDT 2.0 data (Hajič et al., 2006).

8.1.4 PDT 2.0 lemmatizer

The PDT 2.0 lemmatizer (Hajič et al., 2006) uses the most complex approach. The system as a whole is a hybrid system[1]. 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

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

derivation rules. The tagger is statistical (feature-based). The system also contains a statistical guesser for out-of-dictionary words.

8.1.5 Majka

Majka (Šmerk, 2009) 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.

8.1.6 MorphoDiTa

MorphoDiTa[2] (Straková et al., 2014) is a state-of-the-art tool for morphological analysis, which is based on the averaged perceptron algorithm (Collins, 2002a). 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.

8.2 NER system

Our NER system is based on Conditional Random Fields (CRF) (Lafferty et al., 2001a), which are considered as the best method for NER by many authors. We use Brainy (Konkol, 2014) implementation of CRF.

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

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

[2] https://ufal.mff.cuni.cz/morphodita

Lemma – The lemmatization approaches are described in section 8.1. 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 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 `-` (Ciaramita and Altun, 2005b).

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 Development

8.3 Experiments and discussion

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. The experiments are done on the CNEC 1.1 and 2.0 corpora. For all experiments, we use the feature set described in section 8.2. It consists of frequently used features with default parameters and should work very well in all our experiments.

The evaluation is done using the strict (CoNLL) and lenient metrics described in section 2.5. The results of our experiments are shown in Tables 8.1 and 8.2. 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 and less language dependent 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.

		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 8.1: Results for the CNEC 1.1.

		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

Table 8.2: Results for the CNEC 2.0.

9 Named Entity Disambiguation

“Start by doing what’s necessary;
then do what’s possible;
and suddenly
you are doing the impossible.”
Francis of Assisi

Multiple tasks are related to the named entity disambiguation (NED). Most often, NED is defined by three subtasks: entity linking, NIL detection and NIL clustering. Using this definition, the input data is a text with annotated entity mentions and a knowledge base. The knowledge base contains real world entities (which may be ambiguous) and a description for each entity. Wikipedia is very often used as the knowledge base. Entity linking tries to link all entity mentions to the correct entities (entries in the knowledge base). NIL detection tries to decide whether the entity is known (i.e. refers the entity mention really to the most similar knowledge base entry) or not. NIL clustering groups the unknown entities in such a way, that each group refers to a single entity. Later, we use *full named entity disambiguation* for this definition. This definition is used in the knowledge base population (KBP) task of the Text Analysis Conference (TAC) (Simpson et al., 2010). New data set was created (and later extended) for this task. There are also data for cross-lingual entity linking (Ji et al., 2011).

The web person search task (Artiles et al., 2010) is also related to NED. The task is to cluster web pages returned by a search engine for a person name query so that each cluster refers to a different person (with the same name). This task is closely related (or identical) to the NIL clustering task.

NED addresses two problems at the same time. The *ambiguity* problem is when there are multiple entities (knowledge base entries) with the same surface form (entity mention). The *variety* problem occurs if there are multiple surface forms for one entity. The higher is the ambiguity and variety the harder is the NED task.

In the following experiment, we address a special case of entity linking, where the knowledge base does not contain any description of the entities. The only option for entity linking is thus the use of string similarity metrics. This task arise very often in the research as well as in the commercial practice. We have two main goals. First, we want to propose a method, which solves this problem. Second, we need to create a new corpus in order to evaluate the proposed approach.

This experiment is the first step in Czech entity linking. The next step (not addressed here) is to create a corpus for full named entity disambiguation. We are going to use the proposed method in order to choose the best (highly ambiguous) entity mentions for annotation.

9.1 Corpus creation

The corpus is based on press releases from the Czech News Agency and a list of known entities. It is important to note, that with a list of known entities we only try to solve the variety problem and not the ambiguity problem. We have chosen to use only the person names, because they have (according to our estimates) higher frequency and variety in the news domain than organizations and locations.

Each entity was assigned to the corresponding entry in the list of known entities if such an entry exists. There are situations, in which the entity can be assigned to multiple entries or we are not able to decide certainly, e.g. for entity mention “Doe”, we cannot decide if it is “John Doe” or “Jack Doe” from the list of know entities and even if there is only the entry “John Doe”, we cannot be certain that the document is about “John Doe” and not some other “Doe”. For this purpose, two types of links are defined: certain and possible. A certain link is used for entities that can be linked certainly to the list given the document, e.g. “Johnnie” can be certainly linked to “John Doe” only if it is obvious from the document (without external knowledge), that “Johnnie” refers to “John Doe”.

We have annotated 77 documents with 879 entity mentions. The list of known entities contains 21648 entries. From the 879 found entity mentions, 316 are linked to a known entity and 563 are not linked. Certain link was assigned to 253 of the linked entities and possible link to 63 entity mentions. There were 213 possible links in total, what makes the average of more than 3 possible links for the 63 entity mentions.

The certainly linked entity mentions referenced 96 distinct entries in the dictionary. Each linked entity mention is a surface form of the particular known entity. There were 38 known entities referenced by more than one surface form, so the variety is approximately 39.5%. On average the referenced known entities have 1.9 surface forms. The most surface forms (9) and links (15) were found for "Ehud Olmert", an Israel politician and former prime minister. There are multiple documents in the corpus dealing with Israel politics. Figures 9.1 and 9.2 show the histograms of the number of surface forms and links.

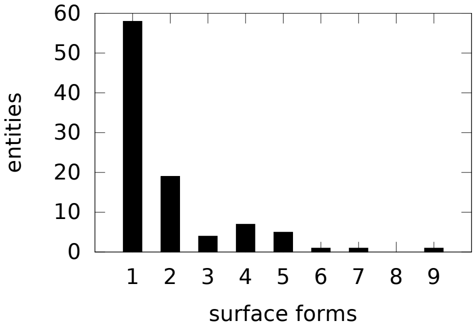


Figure 9.1: Histogram of surface forms per entity.

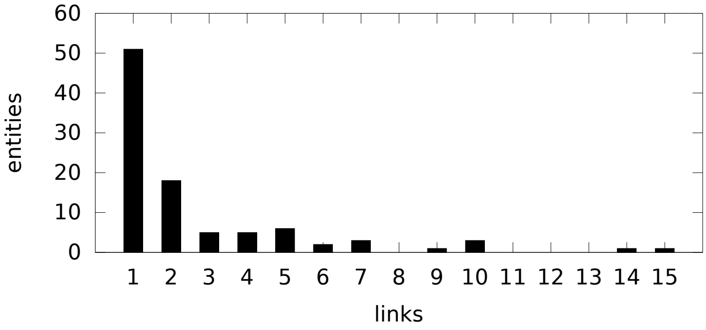


Figure 9.2: Histogram of links per entity.

9.2 Similarity metrics

In this section, we introduce the string similarity metrics used in our experiments. Before their introduction, we need to define our mathematical notation. We compare strings a and b. The length of the string a is denoted as $|a|$. We say that A is a set of n-grams (and their counts) contained in string a. The sum of counts of all n-grams in this set is denoted as $|A|$. The union $A \cup B$ contains all n-grams of A and B, where the count of a shared n-gram is the maximum of their original counts. The intersection $A \cap B$ contains all shared n-grams and their counts are minimums of their original counts. E.g. we have $a =$ 'aaab' and $b =$ 'aabc', the sets are $A = \{(\text{'aa'}, 2), (\text{'ab'}, 1)\}$ and $B = \{(\text{'aa'}, 1), (\text{'ab'}, 1), (\text{'bc'}, 1)\}$, $|A| = 3$ and $|B| = 3$, the union $A \cup B = \{(\text{'aa'}, 2), (\text{'ab'}, 1), (\text{'bc'}, 1)\}$ and the intersection $A \cap B = \{(\text{'aa'}, 1), (\text{'ab'}, 1)\}$.

The *Levenshtein distance* is probably the most commonly used string distance metric. It computes the minimal edit distance using three edit operations – delete, add, and substitute. The Levenshtein metric is defined as (9.1), where $\mathbf{1}_{a_i \neq b_i} = 1$ if $a_i \neq b_i$ and 0 otherwise. The Levenshtein distance is converted to similarity using (9.2).

$$D_L(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} D_L(i-1,j) + 1 \\ D_L(i,j-1) + 1 \\ D_L(i-1,j-1) + \mathbf{1}_{a_i \neq b_j} \end{cases} & \text{otherwise} \end{cases} \tag{9.1}$$

$$S_L = 1 - \frac{D_L(|a|, |b|)}{\max\{|a|, |b|\}} \tag{9.2}$$

The *Levenshtein-Damerau distance* extends the set of operations in the Levenshtein distance by the transposition of adjacent characters. It is defined by (9.3) and converted to similarity using the same approach as for Levenshtein distance.

$$D_{LD}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} D_{LD}(i-1,j)+1 \\ D_{LD}(i,j-1)+1 \\ D_{LD}(i-1,j-1)+\mathbf{1}_{a_i \neq b_j} \\ D_{LD}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} D_{LD}(i-1,j)+1 \\ D_{LD}(i,j-1)+1 \\ D_{LD}(i-1,j-1)+\mathbf{1}_{a_i \neq b_j} \end{cases} & \text{otherwise} \end{cases} \tag{9.3}$$

The *Jaro distance* was designed for the person names comparison and is defined by (9.4), where m is the number of matching characters and t is the number of transpositions. The characters are considered as matching, if they are the same and their position differs by a maximum of k characters (9.5). The transpositions happen if the matching characters are in different order.

$$S_J = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{9.4}$$

$$k = \left\lfloor \frac{\max\{|a|,|b|\}}{2} - 1 \right\rfloor \tag{9.5}$$

The *Jaro-Winkler distance* is an improvement of the original Jaro distance. It gives higher weight to n first characters and is defined as (9.6). If we denote c the length of a common prefix of a and b, then $l = \max\{c, n\}$. The weight of the first characters is denoted as p, $0 \leq p \leq \frac{1}{n}$. In our experiments we use common settings $p = 0.1$ and $n = 4$. Both these metrics are named “distances”, but in fact they are similarities, i.e. $0 \leq S_{J(W)} \leq 1$ and higher values are assigned to more similar strings.

$$S_{JW} = S_J + lp(1 - S_J) \tag{9.6}$$

The *Jaccard similarity*, *Overlap similarity*, and *Soerensen-Dice similarity*are defined by (9.7), (9.8), and (9.9), respectively. These similarities were not originally proposed for string similarity, but can be used for this purpose.

$$S_{Jac} = \frac{|A \cap B|}{|A \cup B|} \tag{9.7}$$

$$S_O = \frac{|A \cap B|}{\min\{|A|, |B|\}} \tag{9.8}$$

$$S_{SD} = \frac{2|A \cap B|}{|A| + |B|} \tag{9.9}$$

The *common prefix similarity* is simply a ration between the length of a common prefix c and the length of one of the strings. We can choose both minimal or maximal length of a and b (9.10), the is denoted in parentheses in the experiments.

$$S_{CP_{max}} = \frac{c}{\max\{|a|, |b|\}} \quad \text{or} \quad S_{CP_{min}} = \frac{c}{\min\{|a|, |b|\}} \tag{9.10}$$

The *longest common subsequence similarity* is the ratio of the length of the longest common subsequence lcs and the length of one of the strings. We can again use minimal or maximal length of a and b (9.11).

$$S_{LCS_{max}} = \frac{lcs}{\max\{|a|, |b|\}} \quad \text{or} \quad S_{LCS_{min}} = \frac{lcs}{\min\{|a|, |b|\}} \tag{9.11}$$

9.3 Proposed combination

The proposed system is based on the maximum entropy classifier. We use the implementation of this algorithm from the Brainy library (Konkol, 2014).

We use the similarities from the previous section as features, but not directly. We firstly tokenize the entity mention and the list entry, then we align the tokens to maximize the overall similarity. For this purpose we use a (suboptimal) greedy algorithm, which seems to be sufficient for the person names. The Hungarian algorithm (Kuhn, 1955) can be used for the optimal alignment, but has higher complexity.

A missing token (one entity has more tokens than the other) is aligned to `null` token and the similarity is set to a constant M. Furthermore, if one of the tokens is an acronym and it can represent the other token, we set the similarity to a constant R. Both R and M are parameters of the system. Using the development data, we have set these parameters to $M = 0.5$ and $R = 0.65$.

The final similarity S is the arithmetic mean of similarities between all tokens. We use the following features for all the similarity metrics:

- Similarity
- Dissimilarity $(1 - S)$
- Intervals of length 0.1 (e.g $0.3 \leq S \leq 0.4$)
- Lesser than a threshold (0.1 step, e.g $S \leq 0.4$)
- Greater or equal than a threshold (0.1 step, e.g $S \geq 0.4$)

9.4 Experiments

The experiments are based on *queries*, similarly to the KBP entity linking task. Each query contains a document with all entity annotations and one annotation (or entity mention) is chosen as the query. Each query is associated with the correct answer, i.e. the correct entry in the list of known entities or indication of unknown entity. There is a limited amount of positive examples (e.g. the entity mention matches the list entry), but very high number of negative examples (e.g. entity mention does not match the list entry). We have decided to use all positive examples and to select three times more negative examples, i.e. the positive examples forms $\frac{1}{4}$ of examples. We have tried to choose the hardest negative examples, where the entity mention is most similar to a wrong entry. The similarity was measured by the Levenshtein metric. This choice penalizes the Levenshtein metric when compared to other metrics as the negative examples are the hardest for Levenshtein metric, but they may be easy for other metrics.

The first experiment was proposed to explore the data and to see the limits of similarity metrics. We compute a similarity s between the entity mention (query) and the list entry using each similarity metric and compare it with threshold t. If $s \geq t$, then we say that the mention matches the entry. Figure 9.3 shows the relation between the chosen threshold and the accuracy. The values in parentheses are choices for the given metric (e.g. order of n-grams).

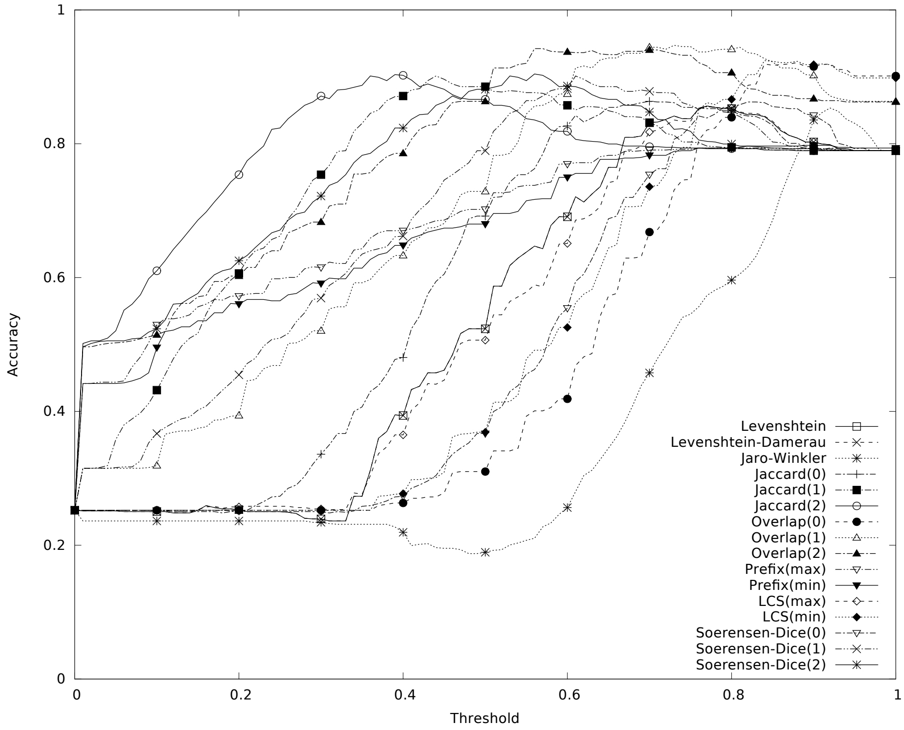


Figure 9.3: Similarity metrics accuracy for various threshold settings.

Our second experiment is done using a 10-fold cross-validation. For each fold, the data are divided in the ratio 80 : 10 : 10 between the training, development and test data, respectively. For each similarity metric, we estimate the optimal threshold using the training data and we apply it on the test data. For the machine learning combination of similarity metrics, we use the training data to find the optimal parameters of the maximum entropy classifier, the development data to find optimal hyperparameters of the model (e.g. the optimal compensation for missing words), and we apply the best model on the test data. The results are shown in Table 9.1.

We can see, that it is possible to achieve accuracy over 90% using a simple similarity metric. The highest score using similarity metric (93.52%) was achieved with Overlap similarity using bigrams. The proposed algorithm

	Accuracy		
Model	Training	Development	Test
Levenshtein	86.37%	84.45%	83.75%
Levenshtein-Damerau	86.37%	84.45%	83.75%
Jaro-Winkler	85.66%	84.95%	83.85%
Jaccard(0)	86.96%	85.84%	85.74%
Jaccard(1)	91.07%	88.33%	89.33%
Jaccard(2)	91.07%	89.13%	86.94%
Overlap(0)	92.71%	91.03%	90.43%
Overlap(1)	95.06%	93.82%	93.52%
Overlap(2)	94.95%	92.42%	92.22%
Prefix(max)	79.20%	79.36%	79.36%
Prefix(min)	79.55%	79.66%	79.66%
LCS(max)	86.37%	84.75%	84.65%
LCS(min)	92.71%	91.72%	91.63%
Soerensen-Dice(0)	86.96%	85.54%	85.64%
Soerensen-Dice(1)	91.07%	88.33%	89.33%
Soerensen-Dice(2)	91.19%	89.43%	87.04%
ML combination	99.79%	97.21%	97.11%

Table 9.1: Results for similarity metrics and their machine learning combination on the training, development, and test data.

further improves the accuracy to 97.11%. These results highly surpassed our expectations.

9.5 Discussion

We have manually created a Czech corpus for a simplified entity linking task and provided the necessary statistics. The data show a rather high variety (39.5%), which can be explained by the rich morphology of Czech.

We have carried out experiments with well-known similarity metrics. The best similarity metric in our experiments was Overlap similarity with accuracy 93.52%. We also propose a classifier based combination of these similarity metrics, which achieved accuracy 97.11%.

10 NER system design

"I have been impressed with the urgency of doing.
Knowing is not enough; we must apply.
Being willing is not enough; we must do."
Leonardo da Vinci

In this chapter, we will briefly introduce the design of our NER systems. At the beginning, we have decided to use the GATE system (Cunningham et al., 2011) as the basic building block for our system. There are two main reasons for this choice. First, GATE is a de facto standard in the NLP community; it is well known and widely used. Our work can be easily shared and the work of others can be easily used in our systems. Second, the design of GATE was already tested by time. The first stable version was released 20 years ago in 1995. The current stable version was released in 2015.

GATE is a modular system and provides interfaces for all common components of NLP applications. There are three component types: *language resources*, *processing resources* and *visual resources*. The language resources represent data of various types (corpora, annotations, ontologies, etc.) and formats.

The processing resources represent the NLP tools (or tasks). Each tool is represented by a separate module in GATE, e.g. a basic module used in almost all applications is tokenizer. The input of a module is the text and annotations provided by other modules; the output of a module are usually new annotations. The application is then defined by a pipeline controlling the order, in which modules are run.

The visual resources are related to GUI and are not very important from our point of view.

10.1 Created modules

We have created a total of 27 GATE modules. All the modules are related to the common NER pipeline and can be categorized into the following categories.

Core NE Modules for named entity recognition and disambiguation.

Segmentation Modules for identifying words or sentences.

Morphology Modules for lemmatization, tagging, and stemming.

Data formats Modules responsible for loading data from different sources.

Evaluators Modules implementing various evaluation metrics.

Utilities A wide category of modules, e.g. filtering annotations.

There are 5 core NE modules (which are most related to this work). We have three modules for NER based on rules, classification (e.g. SVM), and sequential classification (e.g. CRF). One module for named entity normalization based on morphology. And finally, a module for basic named entity disambiguation based on string similarity metrics (see Section 9). All the modules are parametrisable, e.g. the NER machine learning algorithm can be changed by a module parameter.

The implementation of machine learning algorithms are mainly provided by our machine learning library called Brainy (Konkol, 2014).

10.2 Wrappers

We have created three wrappers to simplify the use of GATE pipelines (or applications). The purpose of these wrappers is to shield the user from the complexities of GATE and to provide easy ways of using NLP applications. The *console wrapper* is mainly used for experiments, because it can be easily run remotely on a cluster. The *application wrapper* is used to incorporate GATE pipeline into other application. The *web application wrapper* is used to provide the GATE pipeline as a web service.

We define an XML format for describing GATE pipeline. This XML file is then used as an input for the wrappers. This allows us to easily experiment, because each experiment is defined only by a set of XML files. It is also easy to manage a large number of experiments and reproduce them. The best system can be easily transferred from experiments to practical use in the application and web application wrappers.

The application and web application wrappers have been already used in multiple university projects and commercial systems.

10.3 Discussion

Our choice of GATE as the base for our system brings some advantages and also disadvantages. As already said, the main advantages are reusability and compatibility. The main disadvantages are complexity (size of the project) and computational overhead. The complexity of the system can be (partially) hidden for the end user by the wrappers, but developers still need to be familiar with GATE. The computational overhead is the curse of generality. A single purpose system is always more effective than a general system that covers a lot of possibilities. Fortunately, the computational overhead in our system is reasonable.

11 Conclusions

> *"Now this is not the end. It is not*
> *even the beginning of the end. But it is,*
> *perhaps, the end of the beginning."*
> *Winston Churchill*

In this thesis, we present our contribution in the named entity recognition and disambiguation tasks.

Our experiments in named entity recognition cover all the important aspects of named entity recognition. We have experimented with semantic features, machine learning algorithms, segment representations, preprocessing, etc. If we look at our work as a whole, our main contribution is the focus on multilinguality and other languages than English. We have shown that the multilingual systems can compete with language dependent systems and that languages like Czech bring new challenges to the named entity recognition task. Regarding Czech, the performance was improved by more than 10% during the work on this thesis, which is a big step forward.

In named entity disambiguation, we focus on bringing this task to Czech language. Our experiment can be seen as a first step in this direction and helps us with the creation of the new Czech named entity disambiguation corpus.

11.1 Future work

In named entity recognition, we would like to find a way, which would allow us to optimize the semantic model for named entity recognition during training. In other words, we would like to create a model focused on the semantic information related to named entity recognition instead of a general model.

In named entity disambiguation, we currently prepare a new Czech named entity disambiguation corpus, which will open the way for further development in this task.

11.2 Fulfilment of the thesis goals

In the following paragraphs, we summarize our contribution according to the thesis goals.

Develop new recognition methods and features to improve performance for Czech and other languages. All the publications listed in Appendix A are directly or indirectly related to this point. In (Konkol et al., 2015), we explore the unsupervised semantic features which highly improve the results. In (Konkol and Konopík, 2014), we study the effect of various stemming and lemmatization approaches. In (Konkol and Konopík, 2013), we consolidate the previous research on Czech and propose new NER system. In (Konkol and Konopík, 2015) we explore various segment representation. In (Konkol and Konopík, 2011) we report results of our initial experiments with semantic features. The rest of the publications are related indirectly.

Propose semi-supervised approaches to improve the adaptability of NER. We use unsupervised semantic features, which effectively substitute language dependent features and thus improve the adaptability of our system. In (Konkol et al., 2015), we present a language-independent system, that performed on a similar level or better than language-dependent systems for multiple languages. The first attempts in this direction were done in (Konkol and Konopík, 2011). In (Konkol and Konopík, 2015), we try to experimentally find best segment representation for multiple languages. In (Konkol and Konopík, 2014), we can see the differences between language dependent and independent approaches for stemming and lemmatization.

Experiment with disambiguation on small subset of selected named entities. We have done the first steps in named entity disambiguation in Czech (Konkol, 2015a). Our experiment covered an important task of linking entity mentions in text to a list of known entities. We are currently preparing a new Czech corpus for full named entity disambiguation, which will allow us to experiment with more sophisticated methods.

Create quality and reusable NER system, which will provide standard interfaces. We have implemented multiple NER systems based on various approaches as plugins to the well-known GATE system (Section 10). We have also created many utility plugins such as tokenizers, lemmatizers, stemmers, etc. Currently, the whole pipeline necessary for a successful NER system is available through the GATE plugin system. We have created simple NER systems based on dictionaries and rules as well as state-of-the-art machine learning systems. We have implemented our own machine learning library for this purpose (Konkol, 2014). Our NER system was successfully used in commercial projects.

A Author's publications

"In vain have you acquired knowledge
if you do not impart it to others."
Deuteronomy Rabbah

[c9] M. Konkol. First steps in czech entity linking. In P. Král and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 489–496. Springer International Publishing, 2015a. ISBN 978-3-319-24032-9. doi: 10.1007/978-3-319-24033-6_55. URL `<http://dx.doi.org/10.1007/978-3-319-24033-6_55>`

[c8] M. Konkol and M. Konopík. Segment representations in named entity recognition. In P. Král and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 61–70. Springer International Publishing, 2015. ISBN 978-3-319-24032-9. doi: 10.1007/978-3-319-24033-6_7. URL `<http://dx.doi.org/10.1007/978-3-319-24033-6_7>`

[c7] M. Konkol. Fuzzy agglomerative clustering. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, Lecture Notes in Computer Science. Springer International Publishing, 2015b

[j1] M. Konkol, T. Brychcín, and M. Konopík. Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470 – 3479, 2015. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2014.12.015. URL `<http://www.sciencedirect.com/science/article/pii/S0957417414007933>`

[c6] T. Brychcín, M. Konkol, and J. Steinberger. UWB: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL `<http://www.aclweb.org/anthology/S14-2145>`

[c5] M. Konkol and M. Konopík. Named entity recognition for highly inflectional languages: Effects of various lemmatization and stemming approaches. In P. Sojka, A. Horák, I. Kopeček, and K. Pala, editors, *Text, Speech and Dialogue*, volume 8655 of *Lecture Notes in Computer Science*, pages 267–274. Springer International Publishing, 2014. ISBN 978-3-319-10815-5. doi: 10.1007/978-3-319-10816-2_33. URL `<http://dx.doi.org/10.1007/978-3-319-10816-2_33>`

[c4] J. Steinberger, T. Brychcín, and M. Konkol. Aspect-level sentiment analysis in czech. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 24–30, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `<http://www.aclweb.org/anthology/W14-2605>`

[c3] M. Konkol. Brainy: A machine learning library. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer, 2014. ISBN 978-3-319-07175-6. doi: 10.1007/978-3-319-07176-3_43. URL `<http://dx.doi.org/10.1007/978-3-319-07176-3_43>`

[c2] M. Konkol and M. Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In I. Habernal and V. Matoušek, editors, *Text, Speech and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 153–160. Springer Berlin Heidelberg, 2013

[c1] M. Konkol and M. Konopík. Maximum entropy named entity recognition for czech language. In I. Habernal and V. Matoušek, editors, *Text, Speech and Dialogue*, volume 6836 of *Lecture Notes in Computer Science*, pages 203–210. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23537-5. doi: 10.1007/978-3-642-23538-2_26. URL `<http://dx.doi.org/10.1007/978-3-642-23538-2_26>`

Bibliography

Proc. 7-th Message Understaning Conference, 1998. URL <http://aclweb.org/anthology/M/M98/>.

J. An, S. Lee, and G. G. Lee. Automatic acquisition of named entity tagged corpus from world wide web. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, ACL '03, pages 165–168, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. ISBN 0-111-456789. doi: 10.3115/1075178.1075207. URL <http://dx.doi.org/10.3115/1075178.1075207>.

J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

F. Béchet, A. Nasr, and F. Genet. Tagging unknown proper names using decision trees. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 77–84, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075229. URL <http://dx.doi.org/10.3115/1075218.1075229>.

Y. Benajiba, M. Diab, and P. Rosso. Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 284–293, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1613715.1613755>.

O. Bender, F. J. Och, and H. Ney. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages

148–151, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119196. URL <http://dx.doi.org/10.3115/1119176.1119196>.

D. Benikova, C. Biemann, M. Kisselew, and S. Padó. Germeval 2014 named entity recognition shared task: Companion paper. In *Konvens 2014, GermEval workshop*, Hildesheim, Germany, 2014.

A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22:39–71, March 1996. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=234285.234289>.

D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. doi: 10.3115/974557.974586. URL <http://dx.doi.org/10.3115/974557.974586>.

D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what&lsquo;s in a name. *Mach. Learn.*, 34(1-3):211–231, Feb. 1999. ISSN 0885-6125. doi: 10.1023/A:1007558221122. URL <http://dx.doi.org/10.1023/A:1007558221122>.

W. Black, F. Rinaldi, and D. Mowatt. Facile: Description of the ne system used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. URL <http://aclweb.org/anthology/M98-1014>.

D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. URL <http://aclweb.org/anthology/M98-1018>.

A. E. Borthwick. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, New York, NY, USA, 1999. AAI9945252.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18

(4):467–479, Dec. 1992. ISSN 0891-2017. URL <`http://dl.acm.org/citation.cfm?id=176313.176316`>.

T. Brychcín and I. Habernal. Unsupervised improving of sentiment analysis using global target context. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 122–128, Hissar, Bulgaria, September 2013. INCOMA Ltd. Shoumen, BULGARIA. URL <`http://www.aclweb.org/anthology/R13-1016`>.

T. Brychcín and M. Konopík. Semantic spaces for improving language modeling. *Computer Speech & Language*, 28(1):192 – 209, 2014. ISSN 0885-2308. doi: http://dx.doi.org/10.1016/j.csl.2013.05.001. URL <`http://www.sciencedirect.com/science/article/pii/S0885230813000387`>.

T. Brychcín and M. Konopík. Hps: High precision stemmer. *Information Processing & Management*, 51(1):68 – 91, 2015. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2014.08.006. URL <`http://www.sciencedirect.com/science/article/pii/S0306457314000843`>.

T. Brychcín, M. Konkol, and J. Steinberger. UWB: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL <`http://www.aclweb.org/anthology/S14-2145`>.

C. Burgess and K. Lund. Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12:177–210, 1997.

X. Carreras, L. Màrquez, and L. Padró. Named entity extraction using adaboost. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118857. URL <`http://dx.doi.org/10.3115/1118853.1118857`>.

X. Carreras, L. Màrquez, and L. Padró. A simple named entity extractor using adaboost. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 152–155. Edmonton, Canada, 2003a.

X. Carreras, L. Màrquez, and L. Padró. A simple named entity extractor using adaboost. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 152–155. Edmonton, Canada, 2003b.

W. G. Charles. Contextual correlates of meaning. *Applied Psycholinguistics*, 21(04):505–524, 2000.

S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. Technical report, Computer Science Group, Harvard University, 1998.

H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 160–163. Edmonton, Canada, 2003.

H.-C. Cho, N. Okazaki, M. Miwa, and J. Tsujii. Named entity recognition with multiple segment representations. *Information Processing & Management*, 49(4):954 – 965, 2013. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2013.03.002. URL `<http://www.sciencedirect.com/science/article/pii/S0306457313000368>`.

M. Ciaramita and Y. Altun. Named-entity recognition in novel domains with external lexical knowledge. *Proceedings of the NIPS Workshop on Advances in Structured Learning for Text and Speech Processing*, (Section 00): 0–3, 2005a. URL `<http://www.cis.upenn.edu/~crammer/workshop_material/ciaramita_altun_structlearn.pdf>`.

M. Ciaramita and Y. Altun. Named-Entity Recognition in Novel Domains with External Lexical Knowledge. 2005b. URL `<http://www.cis.upenn.edu/\~{}crammer/workshop\_material/ciaramita\_altun\_structlearn.pdf>`.

N. Collier and K. Takeuchi. Comparison of character-level and part of speech features for name recognition in biomedical texts. *Journal of Biomedical Informatics*, 37(6):423 – 435, 2004. ISSN 1532-0464. doi: http://dx.doi.org/10.1016/j.jbi.2004.08.008. URL `<http://www.sciencedirect.com/science/article/pii/S1532046404000887>`. Named Entity Recognition in Biomedicine.

M. Collins. 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, pages 1–8, Stroudsburg, PA, USA, 2002a. Association for Computational Linguistics. doi: 10.3115/1118693.1118694. URL `<http://dx.doi.org/10.3115/1118693.1118694>`.

M. Collins. Ranking algorithms for named entity extraction: Boosting and the votedperceptron. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Philadelphia, Pennsylvania, USA, July 2002b. Association for Computational Linguistics. doi: 10.3115/1073083.1073165. URL `<http://www.aclweb.org/anthology/P02-1062>`.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. URL `<http://doi.acm.org/10.1145/1390156.1390177>`.

C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL `<http://dx.doi.org/10.1023/A:1022627411411>`.

J. Cowie. Crl/nmsudescription of the crl/nmsu systems used for muc-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL `<http://aclweb.org/anthology/M95-1013>`.

S. Cucerzan and D. Yarowsky. Language independent ner using a unified model of internal and contextual evidence. In *Proceedings of CoNLL-2002*, pages 171–174. Taipei, Taiwan, 2002.

H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6).* 2011. ISBN 978-0956599315. URL `<http://tinyurl.com/gatebook>`.

J. R. Curran and S. Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 164–167, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119200. URL `<http://dx.doi.org/10.3115/1119176.1119200>`.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):pp. 1470–1480, 1972. ISSN 00034851. URL `<http://www.jstor.org/stable/2240069>`.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

H. Demir and A. Ozgur. Improving named entity recognition for morphologically rich languages using word embeddings. In *The 13th International Conference on Machine Learning and Applications (ICMLA'14)*, Detroit, Michigan, USA, 2014.

B. Desmet and V. Hoste. Dutch named entity recognition using ensemble classifiers. In E. Westerhout, T. Markus, and P. Monachesi, editors, *Computational Linguistics in the Netherlands 2010 : selected papers from the twentieth CLIN meeting (CLIN 2010)*, pages 29–41. Landelijke Onderzoeksschool Taalwetenschap (LOT), 2010. ISBN 9789460930478.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *Proceedings of LREC 2004*, pages 837–840, 2004.

D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson. Sri international fastus systemmuc-6 test results and analysis. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL `<http://aclweb.org/anthology/M95-1019>`.

A. Ekbal and S. Bandyopadhyay. Bengali named entity recognition using support vector machine. In *IJCNLP-08 Workshop on NER for South and South East Asian Languages*, pages 51–58, 2008.

A. Ekbal and S. Saha. Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. *International Journal on Document Analysis and Recognition*, pages 1–24. ISSN 1433-2833. URL `<http://dx.doi.org/10.1007/s10032-011-0155-7>`. 10.1007/s10032-011-0155-7.

A. Ekbal and S. Saha. Classifier ensemble selection using genetic algorithm for named entity recognition. *Research on Language & Computation*, 8: 73–99, 2010. ISSN 1570-7075. URL `<http://dx.doi.org/10.1007/s11168-010-9071-0>`. 10.1007/s11168-010-9071-0.

A. Ekbal and S. Saha. A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in indian languages as case studies. *Expert Systems with Applications*, 38(12):14760 – 14772, 2011. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2011.05.004. URL `<http://www.sciencedirect.com/science/article/pii/S0957417411007871>`.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134, June 2005. ISSN 0004-3702. doi: 10.1016/j.artint.2005.03.001. URL `<http://dx.doi.org/10.1016/j.artint.2005.03.001>`.

M. Faruqui and S. Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010.

O. Ferrández, A. Toral, and R. Muñoz. Fine tuning features and post-processing rules to improve named entity recognition. In *Proceedings of the 11th international conference on Applications of Natural Language to Information Systems*, NLDB'06, pages 176–185, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-34616-3, 978-3-540-34616-6. doi: 10.1007/11765448_16. URL `<http://dx.doi.org/10.1007/11765448_16>`.

L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE, Sept. 2005.

J. R. Finkel and C. D. Manning. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 141–150, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL `<http://dl.acm.org/citation.cfm?id=1699510.1699529>`.

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219885. URL `<http://dx.doi.org/10.3115/1219840.1219885>`.

J. R. Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

D. Fisher, S. Soderland, J. McCarthy, F. Feng, and W. Lehnert. Description of the umass system as used for muc-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL <http://aclweb.org/anthology/M95-1011>.

R. Florian. Named entity recognition as a house of cards: classifier stacking. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118863. URL <http://dx.doi.org/10.3115/1118853.1118863>.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.

G. Fu and K.-K. Luke. Chinese named entity recognition using lexicalized hmms. *SIGKDD Explor. Newsl.*, 7(1):19–25, June 2005. ISSN 1931-0145. doi: 10.1145/1089815.1089819. URL <http://doi.acm.org/10.1145/1089815.1089819>.

J. Fukumoto, F. Masui, M. Shimcheta, and M. Saski. Description of the oki system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. URL <http://aclweb.org/anthology/M98-1004>.

J. Gao, M. Li, A. Wu, and C.-N. Huang. Chinese word segmentation and named entity recognition: A pragmatic approach. *Comput. Linguist.*, 31(4):531–574, Dec. 2005. ISSN 0891-2017. doi: 10.1162/089120105775299177. URL <http://dx.doi.org/10.1162/089120105775299177>.

G. Georgiev, P. Nakov, K. Ganchev, P. Osenova, and K. Simov. Feature-rich named entity recognition for bulgarian using conditional random fields. In *Proceedings of the International Conference RANLP-2009*, pages 113–117, Borovets, Bulgaria, September 2009. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/R09-1022>.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, Apr. 2004. ISSN 0027-8424. doi: 10.1073/pnas.0307752101. URL <http://dx.doi.org/10.1073/pnas.0307752101>.

R. Grishman and B. Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/992628.992709. URL `<http://dx.doi.org/10.3115/992628.992709>`.

Y. Gui, Z. Gao, R. Li, and X. Yang. Hierarchical text classification for news articles based-on named entities. In S. Zhou, S. Zhang, and G. Karypis, editors, *Advanced Data Mining and Applications*, volume 7713 of *Lecture Notes in Computer Science*, pages 318–329. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35526-4. doi: 10.1007/978-3-642-35527-1_27. URL `<http://dx.doi.org/10.1007/978-3-642-35527-1_27>`.

S. Guiasu and A. Shenitzer. The principle of maximum entropy. *The Mathematical Intelligencer*, 7:42–48, 1985. ISSN 0343-6993. URL `<http://dx.doi.org/10.1007/BF03023004>`. 10.1007/BF03023004.

H. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 281–289, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL `<http://dl.acm.org/citation.cfm?id=1620754.1620795>`.

I. Habernal and M. Konopík. Swsnl: Semantic web search using natural language. *Expert Systems with Applications*, 40(9):3649 – 3664, 2013. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.12.070. URL `<http://www.sciencedirect.com/science/article/pii/S0957417412013115>`.

Y. Hahm, J. Park, K. Lim, Y. Kim, D. Hwang, and K.-S. Choi. Named entity corpus construction using wikipedia and dbpedia ontology. In N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

J. Hajič, J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtský, and M. Ševčíková Razímová. Prague dependency treebank 2.0 (PDT 2.0), 2006. URL `<http://hdl.handle.net/11858/00-097C-0000-0001-B098-5>`.

Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

U. Hermjakob, K. Knight, and H. Daumé III. Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397. Association for Computational Linguistics, 2008. URL <http://aclweb.org/anthology/P08-1045>.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

F. Huang. *Multilingual Named Entity Extraction and Translation from Text and Speech*. PhD thesis, Pittsburgh, PA, USA, 2006. AAI3232646.

L. Iwanska, M. Croll, T. Yoon, and M. Adams. Wayne state university: Description of the uno natural language processing system as used for muc-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL <http://aclweb.org/anthology/M95-1021>.

H. Ji, R. Grishman, and H. Dang. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*, 2011.

M. N. Jones and D. J. K. Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114: 1–37, 2007.

D. Jurgens and K. Stevens. The s-space package: An open source package for word space models. *In System Papers of the Association of Computational Linguistics*, 2010.

M. Kabadjov, J. Steinberger, and R. Steinberger. Multilingual statistical news summarization. In T. Poibeau, H. Saggion, J. Piskorski, and R. Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 229–252. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-28568-4. doi: 10.1007/978-3-642-28569-1_11. URL <http://dx.doi.org/10.1007/978-3-642-28569-1_11>.

J. Kanis and L. Skorkovská. Comparison of different lemmatization approaches through the means of information retrieval performance. *Lecture Notes in Artificial Intelligence*, 2010:93–100, 2010. ISSN 0302-9743. URL <http://www.kky.zcu.cz/en/publications/JakubKanis_2010_Comparisonof>.

G. Karypis. Cluto - a clustering toolkit. 2003. URL `<www.cs.umn.edu/~karypis/cluto>`.

J. Kazama and K. Torisawa. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007. URL `<http://www.aclweb.org/anthology-new/D/D07/D07-1073.pdf>`.

J. Kazama and K. Torisawa. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 407–415. The Association for Computer Linguistics, 2008. ISBN 978-1-932432-04-6. doi: http://www.aclweb.org/anthology/P08-1047.

M. Konkol. Named entity recognition. Technical Report DCSE/TR-2012-04, University of West Bohemia, Pilsen, Czech Republic, June 2012.

M. Konkol. Brainy: A machine learning library. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. Springer, 2014. ISBN 978-3-319-07175-6. doi: 10.1007/978-3-319-07176-3_43. URL `<http://dx.doi.org/10.1007/978-3-319-07176-3_43>`.

M. Konkol. First steps in czech entity linking. In P. Král and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 489–496. Springer International Publishing, 2015a. ISBN 978-3-319-24032-9. doi: 10.1007/978-3-319-24033-6_55. URL `<http://dx.doi.org/10.1007/978-3-319-24033-6_55>`.

M. Konkol. Fuzzy agglomerative clustering. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, Lecture Notes in Computer Science. Springer International Publishing, 2015b.

M. Konkol and M. Konopík. Maximum entropy named entity recognition for czech language. In I. Habernal and V. Matoušek, editors, *Text, Speech and Dialogue*, volume 6836 of *Lecture Notes in Computer Science*, pages 203–210. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23537-5. doi: 10.1007/978-3-642-23538-2_26. URL `<http://dx.doi.org/10.1007/978-3-642-23538-2_26>`.

M. Konkol and M. Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In I. Habernal and V. Matoušek, editors, *Text, Speech and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 153–160. Springer Berlin Heidelberg, 2013.

M. Konkol and M. Konopík. Segment representations in named entity recognition. In P. Král and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 61–70. Springer International Publishing, 2015. ISBN 978-3-319-24032-9. doi: 10.1007/978-3-319-24033-6_7. URL `<http://dx.doi.org/10.1007/978-3-319-24033-6_7>`.

M. Konkol and M. Konopík. Named entity recognition for highly inflectional languages: Effects of various lemmatization and stemming approaches. In P. Sojka, A. Horák, I. Kopeček, and K. Pala, editors, *Text, Speech and Dialogue*, volume 8655 of *Lecture Notes in Computer Science*, pages 267–274. Springer International Publishing, 2014. ISBN 978-3-319-10815-5. doi: 10.1007/978-3-319-10816-2_33. URL `<http://dx.doi.org/10.1007/978-3-319-10816-2_33>`.

M. Konkol, T. Brychcín, and M. Konopík. Latent semantics in named entity recognition. *Expert Systems with Applications*, 42(7):3470 – 3479, 2015. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2014.12.015. URL `<http://www.sciencedirect.com/science/article/pii/S0957417414007933>`.

Z. Kozareva, O. Ferrández, A. Montoyo, R. Muñoz, A. Suárez, and J. Gómez. Combining data-driven systems for improving named entity recognition. *Data & Knowledge Engineering*, 61(3): 449 – 466, 2007. ISSN 0169-023X. doi: 10.1016/j.datak.2006.06.014. URL `<http://www.sciencedirect.com/science/article/pii/S0169023X06001157>`. <ce:title>Advances on Natural Language Processing</ce:title> <ce:subtitle>NLDB 05</ce:subtitle>.

P. Král. Features for Named Entity Recognition in Czech Language. In *KEOD*, pages 437–441, 2011.

P. Král. Named entities as new features for Czech document classification. In *15th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2014)*, volume 8404 LNCS, pages 417–427, Kathmandu, Nepal, 6-12 April 2014. ISBN 978-3-642-54902-1. doi: 10.1007/978-3-642-54903-8_35. URL `<http://link.springer.com/chapter/10.1007/978-3-642-54903-8\_35>`.

J. Kravalová and Z. Žabokrtský. Czech named entity corpus and svm-based recognizer. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pages 194–201, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-57-2. URL <http://dl.acm.org/citation.cfm?id=1699705.1699748>.

J. Kravalová and Z. Žabokrtský. Czech named entity corpus and SVM-based recognizer. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pages 194–201, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-57-2. URL <http://dl.acm.org/citation.cfm?id=1699705.1699748>.

H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. doi: 10.1002/nav.3800020109.

A. Kumar and T. M. Sebastian. Sentiment analysis on twitter. *IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012*, 9(4):372 – 378, 2012. ISSN 1694-0814.

J. Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225 – 242, 1992. ISSN 0885-2308. doi: http://dx.doi.org/10.1016/0885-2308(92)90019-Z. URL <http://www.sciencedirect.com/science/article/pii/088523089290019Z>.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001a. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL <http://dl.acm.org/citation.cfm?id=645530.655813>.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001b. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

D. Lin. Using collocation statistics in information extraction. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference*

Held in Fairfax, Virginia, April 29 - May 1, 1998, 1998. URL <http://aclweb.org/anthology/M98-1006>.

D. Lin and X. Wu. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1030–1038, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL <http://dl.acm.org/citation.cfm?id=1690219.1690290>.

B. Liu and C. Li. An efficient feature selection method using named entity recognition for chinese text categorization. In *Machine Learning and Cybernetics, 2009 International Conference on*, volume 6, pages 3527–3531, July 2009. doi: 10.1109/ICMLC.2009.5212749.

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45:503–528, December 1989. ISSN 0025-5610. doi: 10.1007/BF01589116. URL <http://portal.acm.org/citation.cfm?id=81100.83726>.

X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 359–367, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL <http://dl.acm.org/citation.cfm?id=2002472.2002519>.

K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods Instruments and Computers*, 28(2):203–208, 1996.

R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118871. URL <http://dx.doi.org/10.3115/1118853.1118871>.

X. Mao, W. Xu, Y. Dong, S. He, and H. Wang. Using Non-Local Features to Improve Named Entity Recognition Recall. volume 21. The Korean Society for Language and Information (KSLI), 2007. URL <http://aclweb.org/anthology/Y/Y07/Y07-1031.pdf>.

A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191. Edmonton, Canada, 2003.

A. McCallum, D. Freitag, and F. C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2. URL <http://portal.acm.org/citation.cfm?id=645529.658277>.

A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

D. S. McNamara. Computational methods to extract meaning from text and advance theories of human cognition. *Topics in Cognitive Science*, 3(1): 3–17, 2011. ISSN 1756-8765. doi: 10.1111/j.1756-8765.2010.01117.x.

A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 1–8, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. doi: 10.3115/977035.977037. URL <http://dx.doi.org/10.3115/977035.977037>.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel, and t. Annotation Group. Bbn: Description of the sift system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. URL <http://aclweb.org/anthology/M98-1009>.

A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 641–648, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273577. URL <http://doi.acm.org/10.1145/1273496.1273577>.

D. Molla, M. van Zaanen, and D. Smith. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia, November 2006. URL <http://www.aclweb.org/anthology/U06-1009>.

D. Mollá, M. van Zaanen, and S. Cassidy. Named entity recognition in question answering of speech data. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 57–65, Melbourne, Australia, December 2007. URL <http://www.aclweb.org/anthology/U07-1010>.

S. Montalvo, R. Martínez, A. Casillas, and V. Fresno. Multilingual document clustering: An heuristic approach based on cognate named entities. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 1145–1152, Stroudsburg, PA, USA, 2006a. Association for Computational Linguistics. doi: 10.3115/1220175.1220319. URL <http://dx.doi.org/10.3115/1220175.1220319>.

S. Montalvo, R. Martínez, A. Casillas, and V. Fresno. Multilingual news document clustering: Two algorithms based on cognate named entities. In P. Sojka, I. Kopeček, and K. Pala, editors, *Text, Speech and Dialogue*, volume 4188 of *Lecture Notes in Computer Science*, pages 165–172. Springer Berlin Heidelberg, 2006b. ISBN 978-3-540-39090-9. doi: 10.1007/11846406_21. URL <http://dx.doi.org/10.1007/11846406_21>.

A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In S. McDonald and J. Tait, editors, *Advances in Information Retrieval*, volume 2997 of *Lecture Notes in Computer Science*, pages 181–196. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-21382-6. doi: 10.1007/978-3-540-24752-4_14. URL <http://dx.doi.org/10.1007/978-3-540-24752-4_14>.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007. doi: doi:10.1075/li.30.1.03nad. URL <http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002>.

D. Nadeau, P. D. Turney, and S. Matwin. Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In *Proceedings of the 19th international conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence*, AI'06, pages 266–277, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-34628-7, 978-3-540-34628-9. doi: 10.1007/11766247_23. URL <http://dx.doi.org/10.1007/11766247_23>.

S. Narayanan and S. Harabagiu. Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1220355.1220455. URL <http://dx.doi.org/10.3115/1220355.1220455>.

V. Nikoulina, A. Sandor, and M. Dymetman. Hybrid adaptation of named entity recognition for statistical machine translation. In *Proceedings of the Second Workshop on Applying Machine Learning Techniques to Optimise the Division of Labour in Hybrid MT*, pages 1–16, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL <http://www.aclweb.org/anthology/W12-5701>.

C. Nobata, S. Sekine, H. Isahara, and R. Grishman. Summarization system integrated with named entity tagging and ie pattern discovery. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. European Language Resources Association (ELRA), 2002. URL <http://aclweb.org/anthology/L02-1119>.

C. Nobata, S. Sekine, and H. Isahara. *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, chapter Evaluation of Features for Sentence Extraction on Different Types of Corpora. 2003. URL <http://aclweb.org/anthology/W03-1204>.

J. Nocedal. Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782, 1980. URL <http://www.jstor.org/stable/2006193>.

J. Nothman, J. R. Curran, and T. Murphy. Transforming wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132, Hobart, Australia, December 2008. URL <http://www.aclweb.org/anthology/U08-1016>.

J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, (0):–, 2012. ISSN 0004-3702. doi: 10.1016/j.artint.2012.03.006. URL <http://www.sciencedirect.com/science/article/pii/S0004370212000276>.

S. Pal, K. S. Naskar, P. Pecina, S. Bandyopadhyay, and A. Way. *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*, chapter Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation, pages 46–54. Coling 2010 Organizing Committee, 2010. URL <http://aclweb.org/anthology/W10-3707>.

G. Paliouras, V. Karkaletsis, G. Petasis, and C. D. Spyropoulos. Learning Decision Trees for Named-Entity Recognition and Classification. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, ECAI 2000, August 20–25 2000. URL <http://www.ellogon.org/petasis/bibliography/ECAI2000/ECAI-2000.pdf>.

M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, AAAI'06, pages 1400–1405. AAAI Press, 2006. ISBN 978-1-57735-281-5. URL <http://dl.acm.org/citation.cfm?id=1597348.1597411>.

T. Poibeau and L. Kosseim. Proper Name Extraction from Non-Journalistic Texts. *Language and Computers*, pages 144–157, Dec. 2001. ISSN 0921-5034. URL <http://www.ingentaconnect.com/content/rodopi/lang/2001/00000037/00000001/art00011>.

A. Purandare and T. Pedersen. Word sense discrimination by clustering contexts in vector and similarity spaces. *Proceedings of 8th Conference on Computational Natural Language Learning*, pages 41–48, 2004.

L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9. URL <http://dl.acm.org/citation.cfm?id=1596374.1596399>.

L. Rau. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume i, pages 29–32, Feb 1991. doi: 10.1109/CAIA.1991.120841.

A. E. Richman, P. Schone, and F. G. G. Meade. Mining wiki resources for multilingual named entity recognition. *Computational Linguistics*, (June):1–9, 2008. URL <https://docs.google.com/viewer?url=http://www.mt-archive.info/ACL-2008-Richman.pdf>.

B. Riordan and M. N. Jones. Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science*, 3(2):303–345, 2011. ISSN 1756-8765. doi: 10.1111/j.1756-8765.2010.01111.x.

D. L. T. Rohde, L. M. Gonnerman, and D. C. Plaut. An improved method for deriving word meaning from lexical co-occurrence. *Cognitive Psychology*, 7:573–605, 2004.

H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, Oct. 1965. ISSN 0001-0782.

S. K. Saha, S. Sarkar, and P. Mitra. A hybrid feature set based maximum entropy hindi named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, 2008.

M. Sahlgren. An Introduction to Random Indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, 2005.

M. Sahlgren, A. Holst, and P. Kanerva. Permutations as a means to encode order in word space. *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 1300–1305, 2008.

R. Sasano and S. Kurohashi. Japanese named entity recognition using structural natural language processing. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008. URL <http://aclweb.org/anthology/I08-2080>.

S. Sekine, K. Sudo, and C. Nobata. Extended named entity hierarchy. In M. G. Rodríguez and C. P. S. Araujo, editors, *Proceedings of* 3^{rd} *International Conference on Language Resources and Evaluation (LREC'02)*, pages 1818–1824, Canary Islands, Spain, May 2002.

K. Shaalan. A survey of arabic named entity recognition and classification. *Comput. Linguist.*, 40(2):469–510, June 2014. ISSN 0891-2017. doi: 10.1162/COLI_a_00178. URL <http://dx.doi.org/10.1162/COLI_a_00178>.

H. Shen and A. Sarkar. Voting between multiple data representations for text chunking. In B. Kégl and G. Lapalme, editors, *Advances in Artificial Intelligence*, volume 3501 of *Lecture Notes in Computer Science*, pages 389–400. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-25864-3. doi: 10.1007/11424918_40. URL <http://dx.doi.org/10.1007/11424918_40>.

H. Simpson, S. Strassel, R. Parker, and P. McNamee. Wikipedia and the web of confusable entities: Experience from entity linking query creation for tac 2009 knowledge base population. In N. C. C. Chair), K. Choukri,

B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

M. Souza and R. Vieira. *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, chapter Entity-centric Sentiment Analysis on Twitter data for the Potuguese Language. 2013. URL `<http://aclweb.org/anthology/W13-4821>`.

J. Steinberger, T. Brychcín, and M. Konkol. Aspect-level sentiment analysis in czech. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 24–30, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `<http://www.aclweb.org/anthology/W14-2605>`.

R. Steinberger, M. Ehrmann, J. Pajzs, M. Ebrahim, J. Steinberger, and M. Turchi. Multilingual media monitoring and text analysis – challenges for highly inflected languages. In I. Habernal and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 22–33. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40584-6. doi: 10.1007/978-3-642-40585-3_3. URL `<http://dx.doi.org/10.1007/978-3-642-40585-3_3>`.

J. Straková, M. Straka, and J. Hajič. A new state-of-the-art czech named entity recognizer. In I. Habernal and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 68–75. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40584-6. doi: 10.1007/978-3-642-40585-3_10. URL `<http://dx.doi.org/10.1007/978-3-642-40585-3_10>`.

J. Straková, M. Straka, and J. Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `<http://www.aclweb.org/anthology/P/P14/P14-5003.pdf>`.

J. Sun, T. Wang, L. Li, and X. Wu. Person name disambiguation based on topic model. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 2010.

B. M. Sundheim. Overview of results of the muc-6 evaluation. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 13–31, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics. ISBN 1-55860-402-2. doi: 10.3115/1072399.1072402. URL `<http://dx.doi.org/10.3115/1072399.1072402>`.

K. Takeuchi and N. Collier. Use of support vector machines in extended named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118853.1118882. URL `<http://dx.doi.org/10.3115/1118853.1118882>`.

P. P. Talukdar, T. Brants, M. Liberman, and F. Pereira. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 141–148, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL `<http://dl.acm.org/citation.cfm?id=1596276.1596303>`.

E. F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan, 2002.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119195. URL `<http://dx.doi.org/10.3115/1119176.1119195>`.

A. Tkachenko, T. Petmanson, and S. Laur. *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, chapter Named Entity Recognition in Estonian, pages 78–83. Association for Computational Linguistics, 2013. URL `<http://aclweb.org/anthology/W13-2412>`.

M. Tkachenko and A. Simanovsky. Named entity recognition: Exploring features. In J. Jancsary, editor, *Proceedings of KONVENS 2012*, pages 118–127. OGAI, September 2012. URL `<http://www.oegai.at/konvens2012/proceedings/17_tkachenko12o/>`. Main track: oral presentations.

A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. *EACL 2006*, 2006.

A. Toral, E. Noguera, F. Llopis, and R. Muñoz. Improving question answering using named entity recognition. In *Proceedings of the 10th International Conference on Natural Language Processing and Information Systems*, NLDB'05, pages 181–191, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-26031-5, 978-3-540-26031-8. doi: 10.1007/11428817_17. URL <http://dx.doi.org/10.1007/11428817_17>.

J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1858681.1858721>.

P. D. Turney and P. Pantel. From frequency to meaning : Vector space models of semantics. *Journal of Artificial Intelligence Research*, pages 141–188, 2010.

H. van Halteren, W. Daelemans, and J. Zavrel. Improving accuracy in word class tagging through the combination of machine learning systems. *Comput. Linguist.*, 27(2):199–229, June 2001. ISSN 0891-2017. doi: 10.1162/089120101750300508. URL <http://dx.doi.org/10.1162/089120101750300508>.

D. Varga and E. Simon. Hungarian named entity recognition with a maximum entropy approach. *Acta Cybern.*, 18(2):293–301, 2007. URL <http://www.inf.u-szeged.hu/actacybernetica/edb/vol18n2/Varga_2007_ActaCybernetica.xml>.

M. Ševčíková, Z. Žabokrtský, and O. Krůza. Named entities in Czech: annotating data and developing NE tagger. In *Proceedings of the 10th international conference on Text, speech and dialogue*, TSD'07, pages 188–195, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-74627-7, 978-3-540-74627-0. URL <http://dl.acm.org/citation.cfm?id=1776334.1776362>.

P. Šmerk. Fast morphological analysis of czech. In *Proceedings of the Raslan Workshop 2009*, Brno, 2009. Masarykova univerzita. ISBN 978-80-210-5048-8.

M. V. Zaanen and D. Mollá. A named entity recogniser for question answering. In *Proceedings PACLING 2007*, 2007.

Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, Department of Computer Science, University of Minnesota, Minneapolis, 2002.

G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 473–480, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073163. URL `<http://dx.doi.org/10.3115/1073083.1073163>`.