Expert Systems with Applications 42 (2015) 3470–3479

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Latent semantics in Named Entity Recognition

Michal Konkol [*], Tomáš Brychcín, Miloslav Konopík

Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia in Pilsen, Univerzitní 8, 306 14 Plzeň, Czech Republic

ARTICLE INFO

Article history:
Available online 20 December 2014

Keywords:
Named Entity Recognition
Information extraction
Stemming
Semantic analysis
Semantic spaces
Latent Dirichlet allocation

ABSTRACT

In this paper, we propose new features for Named Entity Recognition (NER) based on latent semantics. Furthermore, we explore the effect of unsupervised morphological information on these methods and on the NER system in general. The newly created NER system is fully language-independent thanks to the unsupervised nature of the proposed features. We evaluate the system on English, Spanish, Dutch and Czech corpora and study the difference between weakly and highly inflectional languages. Our system achieves the same or even better results than state-of-the-art language dependent systems. The proposed features proved to be very useful and are the main reason of our promising results.

1. Introduction

Named Entity Recognition (NER) systems search for important phrases (such as cities, personal names, or dates) in a given text. In this way, an NER system can serve as a valuable component for many expert systems, ranging from the standard Natural Language Processing tasks, such as question answering (Álvaro Rodrigo, Pérez-Iglesias, Peñas, Garrido, & Araujo, 2013), machine translation (Chen, Zong, & Su, 2013), social media analysis (Jung, 2012), semantic search (Habernal & Konopík, 2013), or summarization (Baralis, Cagliero, Jabeen, Fiori, & Shah, 2013; Glavaš & Šnajder, 2014; Kabadjov, Steinberger, & Steinberger, 2013) to biomedical domain (Atkinson & Bull, 2012).

The state-of-the-art NER systems are based on machine learning techniques. Many different machine learning methods have been used for NER so far. The most common examples are Hidden Markov Models (Zhou & Su, 2002), Decision Trees (Carreras, Màrquez, & Padró, 2003), Maximum Entropy (Borthwick, 1999), Support Vector Machines (Isozaki & Kazawa, 2002) and Conditional Random Fields (McCallum & Li, 2003). It has been shown that various combinations of these methods yield better results (Ekbal & Saha, 2011; Florian, Ittycheriah, Jing, & Zhang, 2003). All these methods are of the type known as supervised learning, which is the most common learning paradigm in NER. There have also been experiments with semi-supervised and unsupervised systems (Collins & Singer, 1999), but the results of such systems are significantly worse.

The NER task was defined at MUC-6 (Grishman & 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. For the well known CoNLL-2002 (Tjong Kim Sang, 2002) and CoNLL-2003 (Tjong Kim Sang & 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).

In this paper, we present a machine learning based system that can be used without any change on a variety of languages with an available NE corpus and a large unlabeled corpus.

We focus on the use of semantic features. A typical semantic feature used in NER is a gazetteer (Carreras, Màrquez, & Padró, 2002; Florian et al., 2003; Konkol & Konopík, 2013), a list of named entities of the same type. Many systems use gazetteers made by human experts for a given language and domain and thus the system loses its independence to some extent. The first step in the direction of language independent semantic features were experiments with the automatic creation of gazetteers. Some approaches using both semi-supervised and unsupervised methods (Kozareva, 2006) have been published. Lin and Wu (2009) and Tkachenko and Simanovsky (2012) used word and phrase clusters, which can be seen as a substitute for a gazetteer.

[*] Corresponding author. Tel.: +420 377 632 491.
E-mail addresses: konkol@kiv.zcu.cz (M. Konkol), brychcin@kiv.zcu.cz (T. Brychcín), konopik@kiv.zcu.cz (M. Konopík).

http://dx.doi.org/10.1016/j.eswa.2014.12.015
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

In this paper, we further extend this idea and exploit word similarity based on *semantic spaces* to cluster words. These clusters are then used to represent the local semantic information. We also experiment with *topic models*. They are used to represent the global semantic information. Our features are then enriched by a language-independent unsupervised stemming. We study the effects of stemming on both sources of semantic information (semantic spaces and topic models), as well as its effects on weakly and highly inflectional languages.

This research has the following goals:

- Compare our features exploiting latent semantics with other similar features.
- Explore the effects of unsupervised stemming method on both semantic features and NER system in general.
- Study the differences between various languages.

The rest of this article is organized as follows. We start with a brief introduction of latent semantics (Section 2). We follow (in Section 3) with a recapitulation of the previous work about semantic features in NER. Section 4 provides information about our NER system and the way we incorporated the novel features. Section 5 describes our experiments and also shows and discusses their results. And finally Section 6 contains our conclusions and ideas for the future work.

2. Latent semantics

In this paper, we use various methods for modeling latent semantics to improve the quality of our NER system. The basic idea behind these methods is based on distributional hypothesis (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) and Charles (2000), where authors carry out empirical tests on humans.

The computational models (that exploit this hypothesis) usually gather statistics on contexts for each word. These statistics are used to create high-dimensional vectors each representing the meaning for one word. 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 & 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) and 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 modeling the local context – HAL (Section 2.1), COALS (Section 2.2), RI (Section 2.3), BEAGLE (Section 2.4) and P&P (Section 2.5). 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, Dumais, Furnas, Landauer, & Harshman, 1990), PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999) and LDA (Latent Dirichlet Allocation) (Section 2.6). In this paper, we use only LDA as it represents the current state-of-the-art model 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 used in this paper.

2.1. HAL

Hyperspace Analogue to Language (HAL) (Burgess & Lund, 1997; Lund & Burgess, 1996) 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.

2.2. COALS

Correlated Occurrence Analogue to Lexical Semantic (or COALS) (Rohde, Gonnerman, & Plaut, 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.

2.3. Random Indexing

Random Indexing (RI) (Sahlgren, 2005) is based on a different approach from the previously introduced methods. The previous methods created the co-occurrence matrix from the data and context vectors were rows and columns from this matrix. The RI begins already with some initial context vectors and incrementally tries to refine them in a way that ensures similar vectors for similar contexts.

The first step in the RI method is the creation of the initial context vectors. A context vector is randomly generated for each word. Values are usually from the interval $(-1, +1)$. The dimension of the vectors is chosen empirically and usually is in thousands. The high dimension of the vectors and the random initialization makes the probability of similar vectors for two distinct words very small. These vectors are called index vectors.

In the second step the algorithm goes through the training data and updates the context vector for current word by summing up all index vectors for co-occurring words.

The dimension reduction is not needed, because the dimension is set at the beginning and therefore should be reasonable.

An extension of the Random Indexing method is introduced in Sahlgren, Holst, and Kanerva (2008). It allows RI to take word order information into account. It is inspired by the BEAGLE method (Section 2.4), but it uses permutation of vector coordinates instead of convolution, because of the lower computational cost.

2.4. BEAGLE

Bound Encoding of the AggreGate Language Environment (BEAGLE) (Jones & Mewhort, 2007) is a model similar to Random Indexing (Section 2.3).

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.

2.5. Purandare and Pedersen

The Purandare and Pedersen (P&P) (Purandare & 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.

2.6. Latent Dirichlet allocation

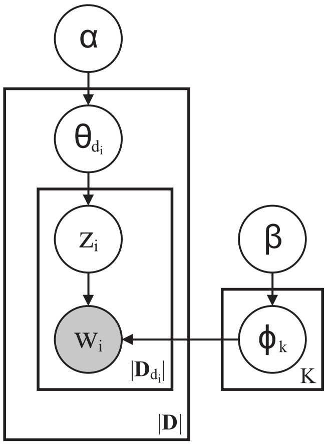

Fig. 1. Graphical model representation of LDA.

Latent Dirichlet Allocation (LDA) (Blei, Ng, Jordan, & Lafferty, 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 Fig. 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).

3. Semantic features in NER

We are not aware of any work that uses LDA for NER in a similar way to our approach. Chrupala (2011) used LDA to produce word clusters, which is a completely different approach. LDA was also used in the task of Named Entity Recognition in Queries (Guo, Xu, Cheng, & Li, 2009). The Named Entity Recognition in Queries is related to NER, but the tasks are different and usually use different approaches.

The work most closely related to our use of semantic spaces is (Lin & Wu, 2009). They improved the results from 83.78 to 88.34[1] by adding word clusters and then further to 90.90 by using phrase clusters. The results were measured on the English CoNLL corpus using the standard evaluation. The method they used can be seen as simplified HAL. Firstly, a co-occurrence matrix is created with a context window of size 1 or 3. Then the vectors (representing words) are clustered by (a soft-clustering version of) the K-Means algorithm.

Ratinov and Roth (2009) used clusters created using the Brown algorithm (Brown, deSouza, Mercer, Pietra, & Lai, 1992). They experimented with non-local features and external knowledge and achieved 90.57. External knowledge was represented by gazetteers and Brown clusters and achieved 88.55 without non-local features. This result is related to our system as we also experiment with clusters and want to compare semantic spaces with Brown clusters. Unfortunately, there are no separate results for the Brown clusters and gazetteers, and it is unclear how much the results are improved by the Brown clusters and how much by the gazetteers.

Turian, Ratinov, and Bengio (2010) tested three different methods for word representation: Brown clustering (Brown et al., 1992), C&W embeddings (Collobert & Weston, 2008), and HLBL embeddings (Mnih & Hinton, 2007). The final system performed well with 90.36. The system used gazetteers and the best result without gazetteers was 89.35.

Tkachenko and Simanovsky (2012) explored many features and combined them into a final system. Their system achieved an F-measure of 91.02 on the English CoNLL corpus. They tested Brown clusters, Clark clusters, and LDA clusters. There are results for the individual features, but they are not in the context of a state-of-

[1] All the results presented in this paper use the F-measure. See Section 5.

the-art baseline system as in (Lin & Wu, 2009; Ratinov & Roth, 2009; Turian et al., 2010). In our opinion, the presented results do not show well the contributions of the individual features or their redundancy.

4. NER system

The structure of our system is depicted in Fig. 2. The system works in two phases – training and test. The training phase uses the training data, natural language text data annotated with entities. The training data are then transformed into feature vectors (representation of words) and labels (representation of entities), one vector and label for each word. The training algorithm estimates parameters for the trained model using these data. The test phase corresponds to normal use of the NER system. Test data (unannotated) are supplied and transformed into feature vectors. The trained model is applied and outputs the annotations for the test data.

The performance of the system is based primarily on the machine leaning method and selected features. We use Conditional Random Fields as the machine learning method and we briefly introduce them in the next section. Then we follow with description of the used features.

4.1. Conditional random fields

Conditional Random Fields (CRFs) are regarded as the best self-standing model for NER. CRFs are an undirected graphical model for conditional probability distribution $p(\mathbf{s}|\mathbf{x})$, where $\mathbf{s}$ are output states and $\mathbf{x}$ are input states. For the standard NER task, we use a model whose output states form a simple chain. The input states are a sequence $\mathbf{x} = x_1, \ldots, x_t$ of tokens and their contexts. The output states are a sequence $\mathbf{s} = s_1, \ldots, s_t$, where state is an n-tuple of NE categories similarly to Hidden Markov Models.

The probability distribution $p(\mathbf{s}|\mathbf{x})$ is equivalent to the distribution $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} = y_1, \ldots, y_t$ is a sequence of labels. This distribution is then given as

$$p(\mathbf{y}|\mathbf{x}) \equiv p(\mathbf{s}|\mathbf{x};\Lambda) = \frac{1}{Z_{\mathbf{x}}} \exp \sum_{j=1}^{t}\sum_{k=1}^{m} \lambda_k f_k(s_j, s_{j-1}, x_j) \tag{1}$$

where $\Lambda = \lambda_1, \ldots, \lambda_m$ are parameters of the model, t is number of tokens in a sequence, m is number of features and $Z_{\mathbf{x}}$ is a normalizing factor, which sums the scores of all possible sequences and is given by (2). Feature functions $f_k(s_j, s_{j-1}, x_j)$ can express a wide variety of evidence. An example of a feature function is given by (3).

$$Z_{\mathbf{x}} = \sum_{\mathbf{s}} \exp \sum_{j=1}^{t}\sum_{k=1}^{m} \lambda_k f_k(s_j, s_{j-1}, x_j) \tag{2}$$

$$f_k(s_j, s_{j-1}, x_j) = \begin{cases} 1 & x_j = \text{'Peter'} \quad \wedge \\ & s_{j-1} = \text{'O'} \quad \wedge \\ & s_j = \text{'B-PER'} \\ & \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The parameters Λ are found similarly to Maximum Entropy classifiers by maximizing log-likelihood. Usually, the log-likelihood function is extended with a regularization term, which helps to prevent overfitting. In our experiments, we have used a Gaussian prior for regularization. The final function (4) was minimized using L-BFGS method.

$$L(\Lambda) = -\sum_{i=1}^{n} \log(p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)};\Lambda)) - \sum_{k=1}^{m} \frac{\lambda_k^2}{2\sigma^2} \tag{4}$$

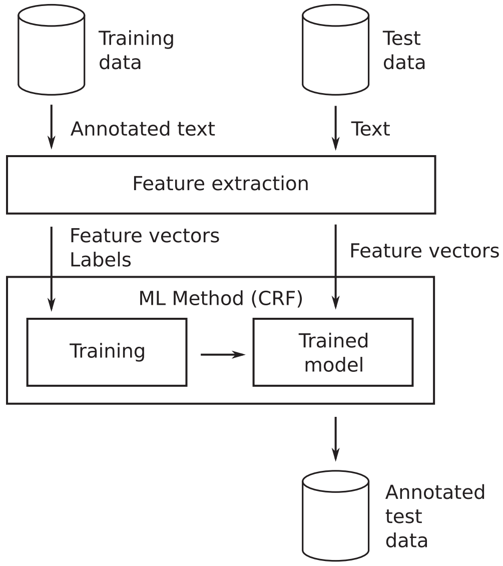

Fig. 2. Structure of our NER system.

4.2. Baseline feature set

The baseline feature set consists of commonly used language independent features and forms a starting point for our experiments.

Word – The word feature is a basic feature used by almost all NER systems. A list of words is created during the training phase. This list is then mapped to a binary feature vector, where one is only at the position assigned to a currently processed word or at the out-of-dictionary position. Each word in a $[-2, 2]$ window is used as a feature and has a separate binary vector.

Bag of words – The bag of words feature is very similar to the word feature, but uses only one binary vector for all words in a $[-2, 2]$ window. In contrast to the word feature, the bag of words feature removes the information about the position in the window and each value in the vector then occurs more often in the data.

N-grams – The *n*-gram feature is used similarly to the word feature, but the word is replaced by an *n*-tuple of words. An *n*-gram is constructed for each word in the $[-2, 2]$ window and the word is always at the last position of the newly created *n*-gram. A word feature is equal to a unigram feature. We use only bigrams (and not higher order *n*-grams), because of the size of NER corpora.

Orthographic features – We use the following set of orthographic features for each word in the $[-2, 2]$ window: *all letters upper*, *first letter upper*, *mixed capitalization*, *contains digit*, *contains apostrophe*, *contains dot*, *contains hyphen*, *contains other symbol*, *contains letter and number*, *contains letter and hyphen*, *contains letter and apostrophe*, *acronym*, *initial*, and *single letter*.

Orthographic patterns – An orthographic pattern works similarly to a word feature, but the letters are transformed by the following rules.

- Upper case letter to `‘A’`.
- Lower case letter to `‘a’`.
- Digit to `‘1’`.
- White spaces to `‘ ’`.
- Other symbols to `‘-’`.

These patterns can be compressed by removing all subsequent characters of the same type. We also use conjunctions of these patterns over a $[-2,2]$ context window.

Affixes – The affix feature is a feature for common beginnings and endings of words. Prefixes and suffixes of length from 4 to 2 are extracted from the words in the $[-2,2]$ window.

4.3. Stemming features

We use the High Precision Stemmer[2] (HPS) (Brychcín & 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 low maximal mutual information loss. 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*.

4.4. Latent Dirichlet allocation

We incorporate LDA (see Section 2.6) 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 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 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.

4.5. 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 than 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 & Karypis, 2002) from the CLUTO library (Karypis, 2003), that has been already used in language modeling (Brychcín & 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.

5. Experiments

All the experiments use a standard CoNLL evaluation with precision (5), recall (6) and *F*-measure (7). These metrics are based on comparison of the NER system output with data annotated by humans. Based on this comparison we can classify the results into four categories:

- True positive (*tp*) – System found an entity and it was also marked by human annotator.
- True negative (*tn*) – System did not find an entity and it was not marked by human annotator.
- False positive (*fp*) – System found an entity, but it was not marked by human annotator.
- False negative (*fn*) – System did not find an entity, that was marked by human annotator.

The CoNNL metric is very strict compared to other metrics. The entity is accepted as correct, if and only if the system correctly marks both the type and the span of the entity. Other metrics often give some kind of a partial score, if the system guesses only one of the entity attributes.

For comparing systems on multiple languages we have defined an overall score (8) 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.

$$P = \frac{tp}{tp + fp} \tag{5}$$

$$R = \frac{tp}{tp + fn} \tag{6}$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

$$Overall = \frac{1}{\frac{1}{4} * \left(\frac{1}{F_{en}} + \frac{1}{F_{es}} + \frac{1}{F_{nl}} + \frac{1}{F_{cz}}\right)} \tag{8}$$

All the presented results were acquired on the test part of the corpora. We have omitted the results on the validation (heldout) data because of space requirements. 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.

5.1. 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

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

Table 1
Results (in *F*-measure) for baseline and stem features.

	English	Spanish	Dutch	Czech	Overall
Baseline	84.19	79.86	76.19	68.21	76.65
Baseline – words replaced by stems	84.00	79.73	77.09	69.29	77.14
Baseline + Stem	84.80	79.71	77.18	69.25	77.32

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 & Konopík, 2013) of the Czech Named Entity corpus (Ševčíková, Žabokrtský, & Krůza, 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[3]; for Spanish, the Reuters corpus RCV2[3]; for Dutch, the Twente News Corpus[4]; and for Czech, the Czech Press Agency corpus. Corpora with approximately 80 million tokens were used for all languages to ensure similar conditions.

5.2. Baseline and stem features

We created a reasonable baseline system with the features described in Section 4.2. 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 1. 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.

5.3. 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 2.

All semantic spaces are implemented in the S-Space library (Jurgens & 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 1,024, the mean value to 0, and the variance to $1/D$. P&P uses 3 meanings for each word.

5.4. Latent Dirichlet allocation

We tested LDA with various numbers of topics. We chose the following numbers of topics based on our experience with LDA: $\{20, 50, 100, 200, 300, 400, 500\}$.

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 3, where Avg represents the average of all five experiments and σ represents the standard deviation.

Table 2
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.

# Clusters		100	500	1000	5000
(a) English (baseline + stem 84.80)					
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)					
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)					
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)					
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

The bold numbers mark the best performing models.

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

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

Table 3
LDA results (in *F*-measure). The results on the top of the row are for standard LDA. The bottom results are for S-LDA.

# 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

The bold numbers mark the best performing models.

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.

5.5. 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 4).

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.

5.6. Discussion

We will start by discussing the unsupervised stemming. Table 1 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.

Table 4
Results (in *F*-measure) for combinations of different clusters and LDA models.

	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) and Carreras et al. (2003)	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**	

The bold numbers mark the best performing models.

The results of semantic spaces are in Table 2. 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 2 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 3. 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 4. 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.

5.7. Comparison with the state of the art

If we compare our best results with the state-of-the-art publications (Table 4), we can see that our system has a very good performance. In comparison with the best English NER system (Lin & 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, Toral, & Muñoz, 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 & 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.

6. Conclusions and future work

We have created a fully language independent NER system based on Conditional Random Fields. Our system works very well for all tested languages, namely, English, Spanish, Dutch and Czech. We have experimented with two sources of semantic information: LDA and semantic spaces. We have also tested the effect of an unsupervised language-independent stemmer. We achieved 89.44 in *F*-measure for English, 83.08 for Spanish, 83.01 for Dutch, and 74.08 for Czech.

The main contribution of this article consists in a successful design of features based on semantic spaces and LDA. To the best of our knowledge, such features have not yet been tested in NER. We have shown that our approach to the creation of clusters yields better results than the approaches in related work. It is also important to note that the results of our fully language-independent NER system are at the same level or sometimes even better than the language-dependent state-of-the-art systems.

In the future, we will study the relation between the size of the training data for the unsupervised features and the NER performance improvement. We would also like to use the phrase clusters created using the semantic spaces. We also believe that there are better ways of incorporating LDA into NER than clusters (Chrupala, 2011) and our current approach. It would be also interesting to study the effect of various multi-word entity encodings (Cho, Okazaki, Miwa, & Tsujii, 2013), because our preliminary experiments show, that the optimal encoding is highly dependent on the feature set. Another possible improvement of our system are the global context features (Fersini, Messina, Felici, & Roth, 2014; Finkel, Grenager, & Manning, 2005).

Acknowledgements

This work was supported by Grant No. SGS-2013-029 Advanced computing and information systems, by the European Regional Development Fund (ERDF) and by project "NTIS - New Technologies for Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090. Access to the MetaCentrum computing facilities provided under the program "Projects of Large Infrastructure for Research, Development, and Innovations" LM2010005, funded by the Ministry of Education, Youth, and Sports of the Czech Republic, is highly appreciated. The access to the CERIT-SC computing and storage facilities provided under the programme Center CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, reg. No. CZ. 1.05/3.2.00/08.0144 is acknowledged.

References

Álvaro Rodrigo Pérez-Iglesias, J., Peñas, A., Garrido, G., & Araujo, L. (2013). Answering questions about european legislation. *Expert Systems with Applications, 40*, 5811–5816. http://dx.doi.org/10.1016/j.eswa.2013.05.008<http://www.sciencedirect.com/science/article/pii/S0957417413002947>.

Atkinson, J., & Bull, V. (2012). A multi-strategy approach to biological named entity recognition. *Expert Systems with Applications, 39*, 12968–12974. http://dx.doi.org/10.1016/j.eswa.2012.05.033<http://www.sciencedirect.com/science/article/pii/S0957417412007397>.

Baralis, E., Cagliero, L., Jabeen, S., Fiori, A., & Shah, S. (2013). Multi-document summarization based on the yago ontology. *Expert Systems with Applications, 40*, 6976–6984. http://dx.doi.org/10.1016/j.eswa.2013.06.047<http://www.sciencedirect.com/science/article/pii/S0957417413004429>.

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

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

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics, 18*, 467–479<http://dl.acm.org/citation.cfm?id=176313.176316>.

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

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

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

Carreras, X., Màrquez, L., & Padró, L. (2003). A simple named entity extractor using adaboost. In W. Daelemans, M. Osborne (Eds.), *Proceedings of CoNLL-2003, Edmonton, Canada* (pp. 152–155).

Carreras, X., Màrquez, L., & Padró, L. (2002). Named entity extraction using adaboost. *Proceedings of the 6th conference on natural language learning* (Vol. 20, pp. 1–4). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1118853.1118857<http://dx.doi.org/10.3115/1118853.1118857>.

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

Chen, Y., Zong, C., & Su, K. Y. (2013). A joint model to identify and align bilingual named entities. *Computational linguistics, 39*, 229–266. http://dx.doi.org/10.1162/COLI_a_00122<http://dx.doi.org/10.1162/COLI_a_00122>.

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

Chrupala, G. (2011). Efficient induction of probabilistic word classes with LDA. In *Proceedings of 5th international joint conference on natural language processing, Asian federation of natural language processing, Chiang Mai, Thailand* (pp. 363–372). URL: <http://www.aclweb.org/anthology-new/I/I11/I11-1041.bib>.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora* (pp. 100–110).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1390156.1390177<http://doi.acm.org/10.1145/1390156.1390177>.

Curran, J. R., & Clark, S. (2003). Language independent ner using a maximum entropy tagger. *Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003* (Vol. 4, pp. 164–167). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1119176.1119200<http://dx.doi.org/10.3115/1119176.1119200>.

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

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

Ferrández, O., Toral, A., & Muñoz, R. (2006). 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* (pp. 176–185). Berlin, Heidelberg: Springer-Verlag. http://dx.doi.org/10.1007/11765448_16<http://dx.doi.org/10.1007/11765448_16>.

Fersini, E., Messina, E., Felici, G., & Roth, D. (2014). Soft-constrained inference for named entity recognition. In *Information processing* & *management* URL: <http://www.sciencedirect.com/science/article/pii/S0306457314000338>, doi: http://dx.doi.org/10.1016/j.ipm.2014.04.005.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 363–370). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1219840.1219885http://dx.doi.org/10.3115/1219840.1219885.

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

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

Glavaš, G., & Šnajder, J. (2014). Event graphs for information retrieval and multi-document summarization. *Expert Systems with Applications, 41*, 6904–6916. http://dx.doi.org/10.1016/j.eswa.2014.04.004<http://www.sciencedirect.com/science/article/pii/S0957417414001985>.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the national academy of sciences of the United States of America, 101*, 5228–5235. http://dx.doi.org/10.1073/pnas.0307752101<http://dx.doi.org/10.1073/pnas.0307752101>.

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. *Proceedings of the 16th conference on computational Linguistics* (Vol. 1, pp. 466–471). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/992628.992709<http://dx.doi.org/10.3115/992628.992709>.

Guo, J., Xu, G., Cheng, X., & Li, H. (2009). Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 267–274). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1571941.1571989<http://doi.acm.org/10.1145/1571941.1571989>.

Habernal, I., & Konopík, M. (2013). Swsnl: Semantic web search using natural language. *Expert Systems with Applications, 40*, 3649–3664. http://dx.doi.org/10.1016/j.eswa.2012.12.070<http://www.sciencedirect.com/science/article/pii/S0957417412013115>.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of 15th conference on uncertainty in artificial intelligence* (pp. 289–296).

Isozaki, H., & Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. *Proceedings of the 19th international conference on computational linguistics* (Vol. 1, pp. 1–7). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1072228.1072282<http://dx.doi.org/10.3115/1072228.1072282>.

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

Jung, J. J. (2012). Online named entity recognition method for microtexts in social networking services: A case study of twitter. *Expert Systems with Applications, 39*, 8066–8070. http://dx.doi.org/10.1016/j.eswa.2012.01.136.

Jurgens, D., & Stevens, K. (2010). The s-space package: An open source package for word space models. In *System papers of the association of computational linguistics*.

Kabadjov, M., Steinberger, J., & Steinberger, R. (2013). Multilingual statistical news summarization. In T. Poibeau, H. Saggion, J. Piskorski, & R. Yangarber (Eds.), *Multilingual information extraction and summarization. Theory and applications of natural language processing* (Vol. 2013, pp. 229–252). Berlin, Heidelberg: Springer. http://dx.doi.org/10.1007/978-3-642-28569-1_11<http://link.springer.com/chapter/10.1007>.

Karypis, G. (2003). Cluto – A clustering toolkit. URL: <www.cs.umn.edu/karypis/cluto>.

Konkol, M., & Konopík, M. (2013). Crf-based czech named entity recognizer and consolidation of czech ner research. In I. Habernal & V. Matoušek (Eds.), *Text, speech and dialogue* (pp. 153–160). Berlin, Heidelberg: Springer.

Kozareva, Z. (2006). Bootstrapping named entity recognition with automatically generated gazetteer lists. In *Proceedings of the eleventh conference of the European chapter of the association for computational linguistics: Student research workshop* (pp. 15–21). Stroudsburg, PA, USA: Association for Computational Linguistics<http://dl.acm.org/citation.cfm?id=1609039.1609041>.

Lin, D., & Wu, X. (2009). Phrase clustering for discriminative learning. *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* (Vol. 2, pp. 1030–1038). Stroudsburg, PA, USA: Association for Computational Linguistics<http://dl.acm.org/citation.cfm?id=1690219.1690290>.

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

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

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

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

Mnih, A., & Hinton, G. (2007). Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on machine learning* (pp. 641–648). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1273496.1273577<http://doi.acm.org/10.1145/1273496.1273577>.

Purandare, A., & Pedersen, T. (2004). Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of 8th conference on computational natural language learning* (pp. 41–48).

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning* (pp. 147–155). Stroudsburg, PA, USA: Association for Computational Linguistics<http://dl.acm.org/citation.cfm?id=1596374.1596399>.

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

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

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM, 8*, 627–633.

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE 2005*.

Sahlgren, M., Holst, A., & Kanerva, P. (2008) . Permutations as a means to encode order in word space. *Proceedings of the 30th annual conference of the cognitive science society* (pp. 1300–1305).

Ševčíková, M., Žabokrtský, Z., & Krůza, O. (2007). Named entities in Czech: Annotating data and developing NE tagger. In *Proceedings of the 10th international conference on text, speech and dialogue* (pp. 188–195). Berlin, Heidelberg: Springer-Verlag<http://dl.acm.org/citation.cfm?id=1776334.1776362>.

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

Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003* (Vol. 4, pp. 142–147). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1119176.1119195<http://dx.doi.org/10.3115/1119176.1119195>.

Tkachenko, M., & Simanovsky, A. (2012). Named entity recognition: Exploring features. In J. Jancsary (Ed.), *Proceedings of KONVENS 2012, OGAI* (pp. 118–127). main track: Oral presentations. URL: <http://www.oegai.at/konvens2012/proceedings/17_tkachenko12o/>.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394). Stroudsburg, PA, USA: Association for Computational Linguistics<http://dl.acm.org/citation.cfm?id=1858681.1858721>.

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

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

Zhou, G., & Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 473–480). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dx.doi.org/10.3115/1073083.1073163<http://dx.doi.org/10.3115/1073083.1073163>.