Latent semantics in language models

Tomáš Brychcín[a,b,*], Miloslav Konopík[a,b]

[a] *Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic*

[b] *NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic*

Abstract

This paper investigates three different sources of information and their integration into language modelling. Global semantics is modelled by Latent Dirichlet Allocation and brings long range dependencies into language models. Word clusters given by semantic spaces enrich these language models with short range semantics. Finally, our own stemming algorithm is used to further enhance the performance of language modelling for inflectional languages.

Our research shows that these three sources of information enrich each other and their combination dramatically improves language modelling. All investigated models are acquired in a fully unsupervised manner.

We show the efficiency of our methods for several languages such as Czech, Slovenian, Slovak, Polish, Hungarian, and English, proving their multilingualism. The perplexity tests are accompanied by machine translation tests that prove the ability of the proposed models to improve the performance of a real-world application.

Keywords: Language Models, Latent Dirichlet Allocation, Semantic spaces, Stemming, HAL, COALS, Random Indexing, HPS, LDA, Machine translation, Moses

1. Introduction

Language modelling is an essential part in many tasks of natural language processing (NLP). Speech recognition, machine translation, optical character recognition, and many other disciplines strongly depend on the language model and thus every improvement in language modelling can also improve the performance of the whole system.

In this paper we explore fully unsupervised methods for language modelling (which require no labelled data and no information about language itself). To prove their multilingualism we experiment with several languages including highly inflectional as well as low-inflection languages. We incorporate three different families of languages (Slavic, Uralic, and Germanic) into our experiments. As representatives of Slavic languages we experiment with Czech, Slovenian, Slovak, and Polish. Uralic languages are represented by Hungarian, and Germanic languages by English. All languages we investigate in this paper except English are characterized by a high level of inflection and relatively free word order (from the syntactic point of view, the words in a sentence can usually be reordered in several ways to carry a slightly different meaning). Properties of these languages complicate the language modelling task. The great number of word forms and large number of possible word sequences lead to a much higher number of n-grams. Data sparsity is a common problem of language models, but for highly inflected languages this problem is even more evident.

The highly inflected languages in this paper belong rather among non-mainstream languages, for which the language modelling task has not gained as much attention as it has for English, for example. We thus

*Corresponding author

Email addresses: brychcin@kiv.zcu.cz (Tomáš Brychcín), konopik@kiv.zcu.cz (Miloslav Konopík)

believe there is considerable potential for improvements. However we provide experiments even for English for mutual comparison with the state of the art.

In this paper we extend our work on the application of semantic spaces in language modelling [Brychcín and Konopík, 2014], where we have achieved significant improvements in perplexity and in machine translation task especially with HAL, COALS and RI models. Thus, these models are investigated more deeply in this paper.

We attempt to improve language modelling by adding long-range semantic dependencies. We choose Latent Dirichlet Allocation (LDA) [Blei et al., 2003] for that task, because it has already been shown by many researches that LDA improves language modelling (see, e.g., [Tam and Schultz, 2005, 2006; Watanabe et al., 2011]).

The performance of these language models is further enhanced by our unsupervised stemming algorithm called High Precision Stemmer (HPS)[1] introduced in [Brychcín and Konopík, 2015]. We have already tested our stemmer in language modelling tasks and the results indicate that HPS performs best compared to other unsupervised stemmers.

To the best of our knowledge we are first to try to combine these three sources of information (i.e. local semantics, global semantics, and morphology).

2. State of the art in latent semantics

In the context of this article we work with various methods for modelling semantic relations between words and use them to improve our language models. The backbone principle of methods for discovering hidden meaning from a plain text is the formulation of Distributional Hypothesis in [Firth, 1957] that says *"a word is characterized by the company it keeps"*. The direct implication of this hypothesis is that the word meaning is related to the context where it usually occurs and thus it is possible to compare the meanings of two words by statistical comparisons of their contexts. This implication was confirmed by empirical tests carried out on human groups in [Charles, 2000].

Several authors have made huge efforts to give an overview of the current state of the art in computational methods for extracting meaning from text [Turney and Pantel, 2010; Riordan and Jones, 2011; McNamara, 2011].

All the methods for extracting meaning can be approximately summarized in two categories. Authors [Riordan and Jones, 2011; McNamara, 2011] categorize these methods into *context-word* and *context-region* approaches. In this paper we use the notation *local context* and *global context*, respectively, because we think this notation describes the principle of meaning extraction better. These two categories are briefly described in the following subsections: 2.1 and 2.2. Additionally, to give a better idea of how these two approaches differ, Figure 1 shows an example of global context and local context semantics of words.

`hockey, Pittsburgh, Jágr, goal, 68`	`hockey, football, soccer, basketball, tennis`
`win, champion, medal, gold, best`	`win, lose, play, wager, defeat`
(a) Global semantics	(b) Local semantics

Figure 1: Examples of semantically similar words. Each row represents semantically similar words according to (a) global semantics with long range dependencies (words in the same line are likely to occur in similar contexts, but not at the same position) and (b) local semantics with short range dependencies (words in the same line should be mutually substitutable at the same position in the appropriate context).

Models based upon the distributional hypothesis usually represent the meaning as a point in a multi-dimensional space. Thus, one meaning is represented as a single vector. These models are then referred to as the vector-space models (VSM). The vector representation enables an easy comparison of word meanings by computing distances between the vectors.

[1] A description of the algorithm and its implementation is available at `http://liks.fav.zcu.cz/HPS`.

2.1. Global context

Global semantics models assume that words that are close in meaning will occur in similar pieces of text (documents). These methods are usually based on the bag-of-words hypothesis, which assumes the word order has no meaning.

LSA (Latent Semantic Analysis) [Deerwester et al., 1990] is a model for discovering global semantic relationships between words. A term-document matrix is constructed and then the SVD (Singular Value Decomposition) is used to reduce the dimension of the matrix and to smooth the values of the matrix.

In [Hofmann, 1999] the PLSA (Probabilistic Latent Semantic Analysis) model was introduced, which is in fact the Bayesian version of LSA. PLSA assumes that the document is a mixture of topics and a topic is simultaneously a mixture of words. The probabilistic output makes this model more easily applicable in many tasks.

PLSA was later extended to LDA [Blei et al., 2003], which places the Dirichlet prior to the document-topic distribution as well as the topic-word distribution. LDA is thus the proper model even for unseen documents that has often been criticized in the case of PLSA.

The motivation for using such methods in language modelling is the fact that text can continuously change in domain, topics, writing style, and so on and it is not possible to recognize these changes only from a short history of words (let us say three words if we use four-gram language models). A much larger history is needed to observe these changes, where of course the data sparsity problem is much more evident. Inhibition of word order thus leads to far fewer possible combinations of histories.

LDA is used for boosting the probabilities of words that are likely to co-occur in the same document (as will be described in Subsection 4.3). This is independent of the position of words in the document (the document is a bag of words). These word probabilities only depend on the document itself (global context). This brings long-range dependencies, and language models are thus adapted to the current domain of text.

2.2. Local context

The second approach to modelling the semantics of words is to use only their local context. Local semantic models assume that the meaning of a word is related to the short context around the word. Methods based on this assumption usually use a small context window (let us say four words in both directions). These methods do not require text that is naturally divided into documents, which is an undisputed advantage over the methods mentioned in the previous section (LSA, PLSA, LDA).

Because of the short context, these methods can take word order into account, so the methods model semantic as well as syntactic relations between words. There is a lot of methods for deriving word meaning according to local context. We have already experimented with five of them in [Brychcín and Konopík, 2014]. In this paper we continue our research and use only the three best performing methods in language modelling (HAL, COALS, and RI).

HAL (Hyperspace Analogue to Language) [Lund and Burgess, 1996] is a very simple method for building semantic space. HAL goes through the corpus and records the co-occurring words around the target word (in some small context window – typically four words). Co-occurring words are weighted inversely to the distance from the target word. This results in the co-occurrence matrix $\mathbb{M} = |W| \times |W|$, where $|W|$ is the vocabulary size. Finally, the row and column vectors of the matrix $\mathbb{M}$ represent the co-occurrence information of words appearing before and after the target word, respectively.

COALS (Correlated Occurrence Analogue to Lexical Semantics) [Rohde et al., 2004] is an extension of the HAL model and starts almost identically to HAL. After recording the co-occurrence information, the raw counts of the matrix $\mathbb{M}$ are converted into the Pearson's correlations. Negative values are zeroed and other values are replaced by their square roots. The optional final step, inspired by LSA [Deerwester et al., 1990], is to apply the SVD reduction to the matrix $\mathbb{M}$, resulting in the smoothing of values and also the discovery of latent semantic relationships between words.

RI (Random Indexing) [Sahlgren, 2005] uses a completely different approach to recording co-occurrence statistics compared to HAL and COALS. For each word in the vocabulary, RI starts by creating high-dimensional index vectors randomly filled with few 1 and -1. The dimension is typically of the order of thousands. Such vectors are very sparse and thus unlikely to overlap. The algorithm then iterates over the

corpus and for each target word it sums all the co-occurring words' index vectors. These sums are recorded in the matrix $\mathbb{M}$. Even though, RI performs a little worse than the other two methods in language modelling, we use it again because it is computationally very undemanding.

In [Brychcín and Konopík, 2014] we also tested BEAGLE [Jones and Mewhort, 2007] and P&P model [Purandare and Pedersen, 2004], but these models did not perform as well in language modelling as the above mentioned methods.

There are several interesting methods which we have not investigated in language modelling yet, but which are certainly worth mentioning.

Similarly to [Purandare and Pedersen, 2004], the authors of [Reisinger and Mooney, 2010a,b] address the common problem of VSMs, where each word is often represented with a single vector, which clearly fails to capture homonymy and polysemy. In their approach, contexts for a single word are clustered to create several meanings of the word. Similar studies, where the word meaning is disambiguated according to the context, can be found in [Dinu and Lapata, 2010; Erk and Padó, 2010; Huang et al., 2012].

Recently, the neural-network models for learning word representations get attention from many researchers. Artificial neural networks hold an implicit ability to store all seen data (words) in their weights. In theory, it should be enough to infer word meanings. However, many researcher specifically design network architectures to support inference of semantic information. For example, recurrent neural networks can use a memory to see sequences and the context.

In [Mikolov et al., 2013], authors introduce skip-gram and continuous bag-of-words (CBOW) models based on a simple single-layer architecture. They proved that even such a simple neural-network architecture can bring promising results. Huang et al. [2012] introduced a model based on neural network, which uses both local and global context via a joint training objective for modelling semantics of the words. They outperform models that use only local context on the word similarity tasks.

In other papers, words are usually regarded to as an independent entities without any relationship between morphologically related word forms. Luong et al. [2013] came with an idea to represent words as a composition of morphemes using the recursive neural network (RNN). The word semantics is than learned by neural language models (NLM).

3. State of the art in language modelling

In the last years, great attention has been concentrated on the exploration of semantic information in language modelling. Further, discovering latent semantic relations between words is more interesting because there are many methods that work in an unsupervised manner. These methods are usually based on assumptions introduced in the previous section (i.e. the distributional hypothesis and the bag-of-words hypothesis). In the context of this article we distinguish three directions of improvements in language modelling (using global context semantics: Subsection 3.1; using local context semantics: Subsection 3.2; and using morphological information: Subsection 3.3).

3.1. Global semantics language models

The use of global semantics in language modelling is motivated by the assumption that documents (long contexts) may differ in domains, topics, and writing styles. This also means that they have a different probability distribution of n-grams. This assumption is used for adapting language models to the long context (domain, topic, style of particular documents). A method such as LSA, PLSA, or LDA is used to find out which documents (which global contexts) are similar and which are not. This long-context information is added to standard n-gram models to adapt them to the global context. The group of language models that benefits from this idea is sometimes called *topic-based language models*.

An important study on the application of LSA to language modelling was presented in [Bellegarda, 2000]. Significant reductions in perplexity (down to 33%) and improvements in speech recognition of English [word error rate (WER) was decreased by 16%] were achieved in this paper. Several authors later obtained good results with PLSA [Gildea and Hofmann, 1999; Wang et al., 2003] and LDA [Tam and Schultz, 2005, 2006] approaches.

Topic Tracing Language Models (TTLM) are investigated in [Watanabe et al., 2011]. These models are based on LDA and PLSA and integrate the ability to dynamically track changes in topics. The tracking is based upon focused text information and previously estimated topics. This research proves that TTLM significantly improves speech recognition of English.

The language models based on semantic composition are described in [Mitchell and Lapata, 2009]. Word vectors are constructed from LDA (word distribution over topics) or SSM (simple semantic space based on word co-occurrence statistics). Different vector compositions are investigated to represent the history of upcoming words in the language model. Their composition models reduce the perplexity of English corpora when combined with a baseline.

3.2. Local semantics language models

The second direction is to use local context semantics for language modelling, where usually class-based language models or similar architectures are used. Individual words are clustered into much smaller number of classes, which reduces the data sparsity problem that the language models try to tackle.

One of the pilot studies in unsupervised language modelling methods was [Brown et al., 1992], where class-based language models of English were introduced. The clustering algorithm builds clusters in a way that will minimize the entropy of the bigram class-based language model. This problem was reformulated so as to maximize the average mutual information between word clusters in whole training corpora [Maximum Mutual Information (MMI) clustering]. To achieve this, classes keep the words that are most probable in the given context (one word window in both directions), which is essentially similar to the distributional hypothesis on which the methods investigated in this paper are based (words occurring in similar contexts are likely to have similar meanings). The MMI algorithm gives very satisfactory results but its computational complexity is very problematic.

This approach was later extended by [Martin et al., 1998] in order to improve the complexity and to work with trigrams (not only bigrams as in Brown's MMI clustering). This clustering algorithm was also used to create class-based language models of Russian and English in [Whittaker and Woodland, 2003]. Linear interpolation with a baseline model improved the perplexity of Russian by 23% and that of English by 7.9%. The authors also present a 2.2% relative reduction of WER in speech recognition of English.

In [Deschacht et al., 2012], the Latent words language model (LWLM) was introduced. This model uses a very similar idea to the methods in [Brown et al., 1992; Martin et al., 1998], but the solutions differ. LWLM represents the word clusters as a latent variable in a graphical model and these clusters consist of the words that are most probable in the given context window. Gibbs sampling or the Expectation Maximization (EM) algorithm is used for inference. LWLM improved the perplexity of language modelling by 14–18% on three English corpora. LWLM groups words according to their local contexts and thus models the semantic relationships between words. LWLM provides an efficient framework that is able to work with a wider context than the MMI approach and with lower complexity.

In [Brychcín and Konopík, 2014] we were the first to apply semantic spaces (working with a small context window) to language models. Our clustering method based on semantic spaces build clusters of words that are likely to occur in similar contexts, which is again a similar idea to the above described methods, but gives a different solution. We experimented with modelling of Czech, Slovak, and English and achieved perplexity reductions ranging between 10 and 18%. These language models were also able to significantly improve the BLEU score in machine translation tests. We also tested the same approach to clustering based on semantic spaces in different applications such as sentiment analysis [Habernal and Brychcín, 2013], where we significantly improved the classification f-measure for Czech and English.

Neural networks are becoming more attractive for language modeling in recent years [Bengio et al., 2003; Schwenk, 2007; Mikolov et al., 2010]. They reach the state-of-the-art performance and successfully compete with n-grams. Neural networks can be designed to capture words contextual meaning which enables them to estimate similarity between words. An unseen word sequence can be then better estimated using a seen word sequence composed of similar words. Given virtually unlimited word combinations such an ability can dramatically increase model performance.

Some researchers also experiment with enrichment of neural networks with an external source of information. For example in [Mikolov and Zweig, 2012], an approximation of LDA topics is feed into the network input alongside with words to add global semantic information.

3.3. Morphology-based language models

A third important direction for improving language modelling is to use morphological information about language. Many authors have already proved that this kind of information can be very useful for the modelling of inflectional languages, which, as stated above, are important for this paper. These approaches usually use supervised methods (lemmatization, part-of-speech tagging, etc.), but unsupervised methods of stemming also exist.

In [Oikonomidis and Digalakis, 2003], the authors used stems for language modelling of Greek. Class-based language models and maximum entropy language models were investigated. The authors present small but significant improvements in the WER of speech recognition.

Another approach to the modelling of Arabic was investigated in [Kirchhoff et al., 2006]. Several different approaches to incorporating morphological information (stems, roots, affixes, morphemes) into language models were tested, with a special focus on factored language models (FLMs) as an architecture. These models successfully improved the performance of speech recognition of Arabic.

Language modelling and speech recognition of Turkish using stems, endings, and morphemes was also investigated in [Arsoy et al., 2006]. The authors present significant improvements in WER by application of their combined model, which uses information about the morphology of Turkish.

In [Oparin, 2008], experiments with morphological random forests (using information about stems, lemmas, parts-of-speech, etc.) in the Czech and Russian languages were shown, with the conclusion that they can be used effectively for inflectional languages.

In [Brychcín and Konopík, 2011] we studied language modelling of Czech and Slovak. We used lemmatization and part-of-speech tagging to derive word clusters for class-based language models and achieved perplexity improvements ranging between 10 and 30% for both languages depending on the amount of training data.

We outlined three main directions (i.e. local semantics, global semantics, and morphology) in language modelling on which researches often focus their attention. To put our research into the context of the state-of-the-art we can state that our method is based on all these sources of information. As will be shown later, these sources of information enrich each other; moreover, their combination dramatically improves language modelling.

The rest of the article is organized as follows. Section 4 describes how the latent semantics is discovered in unlabelled corpora and how it is incorporated into the language models. Our experiments are described in Section 5, which is followed by discussion of the results and the conclusion.

4. Our language models

This section describes our language models and how these models are acquired from an unannotated corpus. The baseline is introduced in the following subsection 4.1. The next subsections present various sources of information, such as the morphology (Subsection 4.2), global semantics (Subsection 4.3), and local semantics (Subsection 4.4), which are used to improve language modelling. Finally, Subsection 4.5 describes how these sources of information are combined together with the baseline.

4.1. Baseline

We are using the Modified Kneser-Ney interpolation (introduced in [Chen and Goodman, 1998]), which is the state-of-the-art approach for smoothing methods. The formula for smoothing of word probabilities is

$$P(w_i|w_{i-n+1}^{i-1}) = \frac{cnt\left(w_{i-n+1}^{i}\right) - D\left(cnt\left(w_{i-n+1}^{i}\right)\right)}{\sum_{w_i} cnt\left(w_{i-n+1}^{i}\right)} + \gamma\left(w_{i-n+1}^{i-1}\right) P\left(w_i|w_{i-n+2}^{i-1}\right), \quad (1)$$

where $P()$ is the probability given by the Modified Kneser-Ney interpolation model and $cnt()$ is the count of the n-gram. The goal of discounting function $D(cnt)$ is to save some probability mass for lower-order models. The normalization function $\gamma(w_{i-n+1}^{i}) \in (0,1)$ makes the probability distribution sum up to 1. The definitions and derivations of these functions can be found in the original paper.

The main advantage of the Modified Kneser-Ney smoothing is the clever way in which it calculates the unigram probability distribution

$$P\left(w_i\right) = \frac{N_{1+}\left(\bullet w_i\right)}{N_{1+}\left(\bullet\bullet\right)}, \tag{2}$$

where symbol $\bullet$ means an arbitrary word (class) and $N_r(w_{i-n+1}^{i})$ is the number of n-grams with frequency r (i.e. the number of such n-grams, where $cnt(w_{i-n+1}^{i}) = r$). In other words, the unigram probability of w_i is given by the number of different bigrams ending in w_i divided by the total number of different bigrams.

4.2. Stem-based language model

We use HPS (see [Brychcín and Konopík, 2015]) as a stemming algorithm. HPS is a new unsupervised stemming algorithm that uses both lexical and semantic information about words to decide how to stem a particular word. The idea of HPS is to split the stemming into two stages. The first stage is based upon a clustering where stemming candidates are selected. The second stage uses a maximum entropy classifier with stemming-specific features that is trained on these candidates.

In our case, stemming is a mapping function $m^S : w \rightarrow s$ that maps words $w \in W$ to stems $s \in S$. Note that stemming is contextually independent (in any context the same word always receives the same stem).

We suppose the stemming should be very helpful for languages with rich morphology. We use three ways of incorporating stemming information into the language modelling. The first way is to use class-based language models with stems representing classes, and the other two ways are used to improve global semantic (Subsection 4.3) and local semantic language models (Subsection 4.4).

Class-based language models are the state-of-the-art approaches to language modelling. The main task of the approach is to replace the statistical dependencies between words with dependencies among a much lower number of word classes, thus reducing the data sparsity problem. In our case the class consists of all words with the same stem.

The probability estimation of a word w_i conditioned by its history w_{i-n+1}^{i-1} (where n is the length of the n-gram) is given by the following formula

$$P^S\left(w_i|w_{i-n+1}^{i-1}\right) = P\left(w_i|s_i\right) P\left(s_i|s_{i-n+1}^{i-1}\right). \tag{3}$$

The probability $P\left(s_i|s_{i-n+1}^{i-1}\right)$ is calculated in the same way as in formula 1, but words are replaced with stems. To calculate the probability $P\left(w_i|s_i\right)$, we use Good-Touring smoothing.

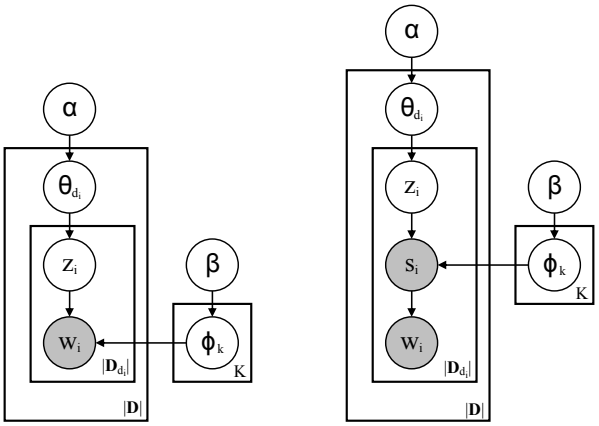
4.3. Global semantics language model

For modelling global context properties we use the well-known topic model LDA [Blei et al., 2003] and our extension of LDA enriched with stem information: stem-based LDA (S-LDA).

LDA models documents as a mixture of topics where each topic is simultaneously a mixture of words. Assume we have a set of documents $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_M\}$ each containing a sequence of words. Let w_i denote a word at position i in a corpus, d_i is a document index to which this word belongs and z_i is a hidden topic label for this word that we try to discover. The graphical model representation is depicted in Figure 2a. The generative process of a word corpus in LDA is as follows:

1. For each document $\boldsymbol{D}_m \in \boldsymbol{D}$, sample a distribution $\boldsymbol{\theta}_m \sim \ Dirichlet\left(\boldsymbol{\alpha}\right)$ over topics $1 \leq z_i \leq K$, where $\boldsymbol{\alpha}$ is a vector of hyper-parameters of Dirichlet distribution.
2. For each topic, sample the word distribution $\boldsymbol{\phi}_k \sim \ Dirichlet\left(\boldsymbol{\beta}\right)$ over words $1 \leq w_i \leq |W|$, where $\boldsymbol{\beta}$ is a vector of hyper-parameters of Dirichlet distribution.
3. For each position i in a corpus:
   (a) Sample a topic label z_i from the distribution $\boldsymbol{\theta}_{d_i}$.

(b) Sample the word w_i from the distribution $\boldsymbol{\phi}_{z_i}$.



(a) LDA (b) S-LDA

Figure 2: Graphical model representation of (a) LDA and (b) S-LDA (stem-based LDA). Note that $|\boldsymbol{D}|$ denotes the number of documents and $|\boldsymbol{D}_{d_i}|$ denotes the size of actual document (the number of word tokens).

For inference of topic assignments, we use Gibbs sampling, which needs to compute $P\left(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right)$, the probability of topic assignment at position i in the corpus given all other topic assignments for all words. According to [Griffiths and Steyvers, 2004] this leads to a simple formula

$$P\left(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) \propto \frac{cnt_{\neg i,k}^{(w_i)} + \beta_{w_i}}{\sum_{j=1}^{|W|}\left[cnt_{\neg i,k}^{(j)} + \beta_j\right]} \cdot \frac{cnt_{\neg i,k}^{(d_i)} + \alpha_k}{\sum_{l=1}^{K}\left[cnt_{\neg i,l}^{(d_i)} + \alpha_l\right]}, \tag{4}$$

where $cnt_{\neg i,k}^{(w_i)}$ is the number of times the topic k has been assigned to a word w_i, except the position i in the corpus. The $cnt_{\neg i,k}^{(d_i)}$ denotes the count of how many times the topic k occurs in the document d_i again except for the position i in the corpus.

From the topic assignments we can easily obtain estimates of $\boldsymbol{\theta}_m$ and $\boldsymbol{\phi}_k$:

$$P(w_i = j|z_i = k, \boldsymbol{\beta}) = \phi_k^{(j)} \approx \frac{cnt_k^{(j)} + \beta_j}{\sum_{j=1}^{|W|}\left[cnt_k^{(j)} + \beta_j\right]} \tag{5}$$

$$P(z_i = k|d_i, \boldsymbol{\alpha}) = \theta_k^{(d_i)} \approx \frac{cnt_k^{(d_i)} + \alpha_k}{\sum_{l=1}^{K}\left[cnt_l^{(d_i)} + \alpha_l\right]}. \tag{6}$$

Finally, to derive the probability of a word w_i in the context of the whole document (global context) we need to marginalize out the topic variable

$$P^{LDA}\left(w_i|d_i\right) = \sum_{k=1}^{K} P\left(w_i|z_i = k\right) P\left(z_i|d_i\right) = \sum_{k=1}^{K} \phi_k^{(w_i)} \theta_k^{(d_i)}. \tag{7}$$

In the second part of this subsection we describe our extension of LDA called S-LDA (shown in Figure 2b). The generative process of S-LDA starts in the same way as the LDA does. Firstly, the topics z_i are

generated. For each z_i the stem s_i is sampled from the Dirichlet distribution and finally the word form w_i is selected.

In many languages, especially those with rich morphology, the suffixes contain morpho-syntactic information of the word. In topic models such as LDA based on the bag-of-words approach, the word order has no meaning and the syntactic information is inhibited. We assume the base forms of the words (approximated by stems) contain a satisfactory amount of information to infer topics. Moreover, taking these properties into account, we suppose that this model will deal with the data sparsity problem better.

Because the variables s_i and w_i are both observed in a corpus and $P(w_i|s_i)$ is constant during sampling z_i, the inference process is almost the same as for LDA (in formulas 4 and 5, the variables $\boldsymbol{w}$, w_i, and W are only replaced with $\boldsymbol{s}$, s_i, and S, respectively, where $\boldsymbol{s}$ denote stems on all positions and S is a set of different stems).

Finally, the unigram probability according to S-LDA is given by

$$P^{S-LDA}(w_i|d_i) = P(w_i|s_i)\sum_{k=1}^{K} P(s_i|z_i=k)\,P(z_i|d_i) = P(w_i|s_i)\sum_{k=1}^{K}\phi_k^{(s_i)}\theta_k^{(d_i)}, \tag{8}$$

where $P(w_i|s_i)$ is calculated in the same way as in Subsection 4.2.

4.4. Local semantics language models

According to our previous research [Brychcín and Konopík, 2014], the well-performing semantic spaces in language modelling were discovered to be:

- HAL (Hyperspace Analogue to Language) [Lund and Burgess, 1996]
- COALS (Correlated Occurrence Analogue to Lexical Semantic) [Rohde et al., 2004]
- RI (Random Indexing) [Sahlgren, 2005]

Since words in these semantic spaces are represented as real-valued vectors, we can apply clustering methods. The main assumption is that words within the same cluster should be semantically substitutable (i.e. they make sense at the same position in the appropriate context).

The selection of a suitable clustering algorithm is crucial for such a task. According to the study in [Zhao and Karypis, 2002], we selected the Repeated Bisection algorithm because of its efficiency and acceptable computational requirements. We use the implementation from the CLUTO software package [Karypis, 2003]. As a measure of the similarity between two words, we use the cosine similarity of word vectors, calculated as the cosine of the angle between corresponding vectors:

$$S_{\cos}\left(\vec{a},\vec{b}\right) = \cos(\theta) = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\cdot\left\|\vec{b}\right\|} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2}\sqrt{\sum b_i^2}}. \tag{9}$$

Since we already have word clusters, we can easily define the mapping function $m : w \to c$, where $w \in W$ denotes a word and $c \in C$ denotes a word cluster. Class-based language models seem to be a suitable architecture for applying this kind of information to the language models.

$$P\left(w_i|c_{i-n+1}^{i-1}\right) = P(w_i|c_i)\,P\left(c_i|c_{i-n+1}^{i-1}\right). \tag{10}$$

The class (cluster) at position i is given by $c_i = m(w_i)$. The probability $P\left(c_i|c_{i-n+1}^{i-1}\right)$ is calculated in the same way as in formula 1, but words are replaced with word classes. To calculate the probability $P(w_i|c_i)$ we use Good-Touring smoothing. The mapping function for particular semantic spaces will be denoted as m_{HAL}, m_{COALS}, and m_{RI}.

Similarly to the previous subsection, we also extend these local semantic models with stemming information. During the building of semantic space we can simply use stems instead of word forms and the mapping function becomes $m : w \to s \to c$, where $w \in W$, $s \in S$, and $c \in C$ and everything else remain unchanged. The mapping functions using semantic spaces together with stemming will be denoted as m_{S-HAL}, $m_{S-COALS}$, and m_{S-RI}.

4.5. Combination of language models

In this paper we work with two ways of combining language models: linear interpolation (see Subsection 4.5.1) and its extension, bucketed linear interpolation (see Subsection 4.5.2).

4.5.1. Linear interpolation

As in our previous works we use a simple but very effective linear interpolation to combine different language models

$$P^{LI}\left(w_i|w_{i-n+1}^{i-1}\right) = \sum_{k=1}^{K} \lambda_k \cdot P_k\left(w_i|w_{i-n+1}^{i-1}\right), \tag{11}$$

where λ_k is the weight of the k-th language model $P_k()$. We use the *Expectation Maximization (EM)* algorithm described in [Dempster et al., 1977] to calculate optimal weights λ_k, which maximizes the likelihood of the held-out data. Using linear interpolation it is quite straightforward to combine different sources of information such as our global and local context language models.

4.5.2. Bucketed linear interpolation

The linear interpolation can be extended to a method called bucketed linear interpolation, where weights become the function of the frequency of word history [Bahl et al., 1983]. The main idea is that the weights λ_k should be different for words with histories of varying frequencies. For example we expect that the word n-gram language model would produce the best probability estimates (receive the highest weight) for very often frequented histories. The formula of linear interpolation is transformed to

$$P^{BLI}\left(w_i|w_{i-n+1}^{i-1}\right) = \sum_{k=1}^{K} \lambda_k\left(w_{i-n+1}^{i-1}\right) \cdot P_k\left(w_i|w_{i-n+1}^{i-1}\right). \tag{12}$$

The weights $\lambda_k()$ certainly cannot be different for each possible frequency of history because of data sparsity. Instead, the whole frequency spectrum is divided into buckets, where each bucket holds some range of frequencies. Histories with frequencies in the same bucket receive the same weights. The number of buckets can be tuned but it generally depends on the amount of training data available. The more training data are available, the more buckets can be used.

In our previous research [Brychcín and Konopík, 2014], it was shown that bucketed linear interpolation produces slightly better results compared to simple linear interpolation when combining several language models.

5. Experimental results

In this section we describe various results of our experiments in detail. Firstly, the corpora we use in our experiments are introduced in Subsection 5.1. Perplexity results, as the most commonly used metric for language models, are shown in the next subsection, 5.2. Finally, machine translation tests are shown in Subsection 5.3. We use six languages for our experiments: Czech (CZ), Slovenian (SL), Slovak (SK), Polish (PL), Hungarian (HU), and English (EN).

In all language models, the vocabulary consists of words occurring at least five times in the training data (and is kept fixed for all tests) and n-grams are smoothed by Modified Kneser-Ney interpolation. These language models will be denoted as follows in our experiments:

- Word and stem-based language models:
  - **BL:** Four-gram word-based language model (baseline).
  - **HPS:** Four-gram class-based language model, where classes are stems given by HPS.
- Global semantics language models:

- **LDA:** Global semantic language models using LDA.
- **S-LDA:** Global semantic language models using a stemmed version of LDA.

- Local semantics language models:
  - **HAL:** Four-gram class-based language model, where classes are given by clustering HAL.
  - **S-HAL:** Four-gram class-based language model, where classes are given by clustering HAL preprocessed by HPS.
  - **COALS:** Four-gram class-based language model, where classes are given by clustering COALS.
  - **S-COALS:** Four-gram class-based language model, where classes are given by clustering COALS preprocessed by HPS.
  - **RI:** Four-gram class-based language model, where classes are given by clustering RI.
  - **S-RI:** Four-gram class-based language model, where classes are given by clustering RI preprocessed by HPS.

LDA as well as the semantic spaces (HAL, COALS, and RI as well as their stemmed versions) works with the same vocabulary as the language models do (words occurring at least five times in the training part of the corpus). We use LDA implementation from the MALLET [McCallum, 2002] software package. For each experiment we always train the LDA with 1,000 iterations of Gibbs sampling. The hyperparameters of Dirichlet distributions were initially set to $\alpha = 50/K$, where K is the number of topics and $\beta = 0.1$. This setting is recommended by [Griffiths and Steyvers, 2004]. The number of topics is ranging between 20 and 1,000 to find an optimal configuration for each language.

Implementation of the HAL, COALS, and RI algorithms is available in an open source package S-Space [Jurgens and Stevens, 2010]. The parameters of these semantic spaces are set as follows. For each semantic space we use a four-word context window (in both directions). HAL uses a matrix consisting of 50,000 columns, which keeps the largest amount of information. COALS uses a matrix with only 14,000 columns (as recommended by the authors of the algorithm). The SVD reduction was not used in our experiments (according to our previous research [Brychcín and Konopík, 2014], COALS with SVD reduction performed worse). RI uses vectors with a dimension of 1024.

For clustering of words, the Repeated Bisection algorithm with the cosine similarity metric is used. We take the implementation from the CLUTO software package [Karypis, 2003]. For all semantic spaces the word vectors are clustered into four different numbers of clusters: 1,000, 5,000, 10,000, and 20,000. The use of more clusters is meaningless as they will be too sparse and too close to the baseline word model (as is clear from statistics on corpora in Subsection 5.1). From these clusters, appropriate class-based language models are constructed.

For stemming we use our own implementation of HPS available at `http://liks.fav.zcu.cz/HPS`. Already trained stemming models for languages used in this paper are also publicly available there.

We use 50 buckets in linear interpolation for combining our language models.

5.1. Corpora

In our experiments we use the Europarl[2] corpora, version 7, provided by [Koehn, 2005]. These corpora consist of parallel texts in many languages extracted from the proceedings of the European parliament. The texts of each language are aligned with the text in English on sentence level.

We chose to use these corpora for two reasons. Besides the perplexities, we test our language models in a machine translation task (where the parallel corpora are needed to train and evaluate the machine translation system). The second reason is that the Europarl corpora contain texts within the same domain (the same texts on the same topics) in several languages, enabling us to make comparisons of our models among different languages.

[2]Available at `http://www.statmt.org/europarl`.

Corpora are tokenized with our simple language-independent tokenizer based upon regular expressions. The casing of all word tokens in corpora is normalized using the *true casing* method available within the Moses framework (see Section 5.3). The *true casing* method uses casing statistics collected from a raw text corpus. The statistics contain information about the most frequent casing of all words.

Parallel corpora are used for training machine translation systems (more information is given in Subsection 5.3). Some statistics on these parallel corpora are shown in Table 1.

Table 1: Statistics on parallel corpora after preprocessing and alignment with appropriate English texts for a machine translation. The number of distinct words in corpora is denoted as *words min. 1*. The number of distinct words occurring at least five times in corpora is denoted as *words min. 5*. The full size of the corpora in terms of number of tokens is denoted as *tokens*. The number of the sentences is denoted as *sentences*.

	tokens	sentences	words min. 1	words min. 5
CZ (CZ-EN)	14,971,635	646,605	176,545	63,455
SL (SL-EN)	14,479,624	623,490	145,307	56,352
SK (SK-EN)	14,865,039	640,715	175,676	65,443
PL (PL-EN)	14,685,556	632,565	182,734	67,242
HU (HU-EN)	14,431,768	624,934	327,374	85,802
EN (CZ-EN)	17,430,472	646,605	67,119	27,494

Texts available in the Europarl corpus are not divided into documents that are required for the training of our global context language models based on LDA. There are, however, publicly available source texts of Europarl corpora that are not already mutually aligned on sentence level (monolingual texts). Fortunately, these source texts are annotated with the tag *SPEAKER*, which indicates particular speakers of the texts (the text is about one topic just discussed in parliament). We assume that texts spoken by one speaker at one moment can be taken as documents. To find the boundaries of the documents we trace the points of change of the *SPEAKER* tag. Every change introduces a new document. We use these documents from these monolingual corpora for training all our language models. Statistics on these monolingual corpora are shown in Table 2. We can see that there are more tokens for all languages, especially for English.

Table 2: Statistics on monolingual corpora after preprocessing. The number of distinct words in corpora is denoted as *words min. 1*. The number of distinct words occurring at least 5 times in corpora is denoted as *words min. 5*. Full size of corpora (the number of tokens) is denoted as *tokens*. The number of distinct documents is denoted as *documents*.

	tokens	documents	words min. 1	words min. 5
CZ	15,100,585	69,927	177,362	63,843
SL	14,547,176	66,600	145,582	56,496
SK	14,967,656	68,669	176,374	65,790
PL	14,873,966	68,444	184,065	67,769
HU	14,548,536	67,160	329,527	86,320
EN	61,260,516	207,946	136,907	49,144

Both parallel and monolingual corpora are split into the training, development (held-out), and test sets in proportions of approximately 70, 10, and 20%, respectively. The held-out set and the test set for both types of corpora are chosen in such a way they contain the same sentences and these sentences are previously unseen for language models as well as for the machine translation model.

5.2. Perplexity results

This subsection presents various experiments with our language models using perplexity as an evaluation measure. Perplexity is the most often used measure of the quality of a language model. The perplexities in the tables below are always calculated on the test part of appropriate corpora (see Section 5.1 about corpora), which is previously unseen for language models.

Baseline perplexities (BL) are shown in all tables in this subsection. Numbers in brackets shown on the right of perplexities of our models denote the relative improvements in perplexity compared to the baseline. We believe that these numbers are more important than the perplexities themselves. Bold numbers represent the best results for the current language. If it is not written explicitly, then the combination of models is always done by linear interpolation (for more information see Subsection 4.5.1).

Our first way of incorporating morphological information (stemming) into language models is to use stem-based language models (class-based languages model with stems used as classes) and to interpolate them with baseline models. The perplexities are shown in Table 3. We can see that there are significant improvements for all languages except English, even with this simple approach. The stem-based model performs similarly for all Slavic languages, that is, for Czech, Slovenian, Slovak, and Polish. For Hungarian, the representative of Uralic languages, the improvements given by stemming are more significant. The almost complete lack of improvement for English as a representative of Germanic languages is not surprising, as word normalization (stemming) is meaningless for languages with almost no inflection.

Table 3: Table represents baseline perplexities (denoted as BL) and also perplexities of linear interpolation of baseline with stem-based model (denoted as BL+HPS).

	BL	BL + HPS
CZ	217.0	201.6 (-7.11%)
SL	168.8	157.4 (-6.77%)
SK	200.7	186.3 (-7.17%)
PL	228.0	209.9 (-7.94%)
HU	287.4	252.5 (-12.17%)
EN	65.2	64.3 (-1.42%)

Local semantics language models are investigated in the next group of experiments, where we use semantic spaces (HAL, COALS, and RI) together with their stemmed versions (S-HAL, S-COALS, and S-RI) clustered into four different depths (1,000, 5,000, 10,000, and 20,000 clusters). Class-based language models given by appropriate clusters are always interpolated with a baseline. Results are shown in Table 4. The last column (where a combination of all class-based language models is shown) in the table is especially interesting. We can see that the semantic spaces clustered into different depths enrich each other and are able to improve language models more than interpolations with only a stand-alone class-based language model.

For all languages, the best semantic space for 1,000, 5,000, and 10,000 clusters, respectively, is conclusively HAL without stemming. In the case of 20,000 clusters, S-HAL performed better for all languages except Hungarian (HAL performed best) and English (the RI model performed best, which we suppose is due to chance rather than its properties). The stemmed version of semantic spaces worked well only in the case of COALS, where S-COALS was almost always better, and also for sparse clusters (20,000 clusters) where the stemmed model was almost always better than the unstemmed one. From these results we can state that the HAL model is most suitable for language modelling of all languages. Perplexity improvements by HAL (baseline interpolated with all four class-based language models created from HAL) are 13% on average for inflectional languages. For English, the improvement is not as big (approximately 7%).

Table 4: Perplexities of local semantics language models. Columns, denoted as 1k, 5k, 10k, 20k, respectively, represent linear interpolation of baseline with the class-based language model given by a particular semantic space clustered into corresponding number of clusters. Last column, denoted as BL+{1k–20k}, represents the linear interpolation of baseline with four class-based language models created from all clusters (1k, 5k, 10k, 20k) of appropriate semantic space. Row denoted as BL represents the perplexity of baseline.

(a) CZ					
BL	217.0				
	Number of classes				
	1k	5k	10k	20k	BL+{1k–20k}

HAL	**199.8 (-7.97%)**	**199.6 (-8.05%)**	**200.4 (-7.67%)**	203.2 (-6.39%)	**191.0 (-12.00%)**
S-HAL	203.9 (-6.05%)	201.9 (-6.99%)	200.7 (-7.53%)	**200.1 (-7.79%)**	195.4 (-9.97%)
COALS	206.9 (-4.66%)	206.6 (-4.83%)	203.5 (-6.26%)	202.6 (-6.65%)	198.7 (-8.47%)
S-COALS	208.3 (-4.05%)	205.0 (-5.54%)	201.4 (-7.21%)	201.2 (-7.30%)	196.6 (-9.41%)
RI	204.9 (-5.61%)	203.5 (-6.25%)	203.4 (-6.28%)	205.5 (-5.33%)	197.1 (-9.21%)
S-RI	206.1 (-5.02%)	204.5 (-5.80%)	203.3 (-6.36%)	202.5 (-6.69%)	198.7 (-8.47%)
(b) SL					
BL	168.8				
	Number of classes				BL+{1k–20k}
	1k	5k	10k	20k	
HAL	**153.7 (-8.99%)**	**154.4 (-8.56%)**	**154.6 (-8.40%)**	157.0 (-6.98%)	**146.7 (-13.09%)**
S-HAL	156.5 (-7.32%)	155.4 (-7.93%)	154.8 (-8.30%)	**154.3 (-8.63%)**	149.7 (-11.33%)
COALS	159.9 (-5.29%)	160.8 (-4.78%)	159.0 (-5.85%)	156.7 (-7.17%)	154.3 (-8.59%)
S-COALS	160.8 (-4.74%)	159.1 (-5.77%)	157.2 (-6.87%)	155.7 (-7.77%)	152.2 (-9.87%)
RI	158.4 (-6.20%)	157.5 (-6.70%)	157.3 (-6.80%)	159.1 (-5.78%)	152.0 (-9.96%)
S-RI	158.8 (-5.97%)	157.4 (-6.74%)	156.9 (-7.09%)	156.5 (-7.29%)	152.7 (-9.55%)
(c) SK					
BL	200.7				
	Number of classes				BL+{1k–20k}
	1k	5k	10k	20k	
HAL	**182.4 (-9.10%)**	**182.7 (-8.96%)**	**183.5 (-8.56%)**	186.8 (-6.90%)	**174.0 (-13.27%)**
S-HAL	186.5 (-7.08%)	184.7 (-7.94%)	183.8 (-8.42%)	**183.3 (-8.66%)**	178.4 (-11.10%)
COALS	189.3 (-5.66%)	190.1 (-5.27%)	188.1 (-6.26%)	186.6 (-7.03%)	182.1 (-9.25%)
S-COALS	191.8 (-4.45%)	188.8 (-5.93%)	185.9 (-7.34%)	185.8 (-7.42%)	180.9 (-9.83%)
RI	188.6 (-6.04%)	187.2 (-6.73%)	187.0 (-6.82%)	189.7 (-5.46%)	181.0 (-9.79%)
S-RI	189.0 (-5.82%)	187.3 (-6.69%)	186.3 (-7.16%)	185.7 (-7.47%)	181.8 (-9.39%)
(d) PL					
BL	228.0				
	Number of classes				BL+{1k–20k}
	1k	5k	10k	20k	
HAL	**206.6 (-9.39%)**	**206.3 (-9.54%)**	**206.9 (-9.25%)**	210.5 (-7.68%)	**195.7 (-14.18%)**
S-HAL	213.3 (-6.46%)	210.3 (-7.78%)	209.0 (-8.33%)	**208.6 (-8.51%)**	203.3 (-10.86%)
COALS	214.5 (-5.92%)	214.2 (-6.07%)	211.7 (-7.14%)	210.8 (-7.54%)	204.0 (-10.53%)
S-COALS	217.4 (-4.65%)	213.3 (-6.48%)	210.3 (-7.77%)	209.6 (-8.06%)	204.3 (-10.40%)
RI	214.0 (-6.14%)	212.2 (-6.95%)	212.1 (-6.99%)	214.5 (-5.95%)	204.5 (-10.30%)
S-RI	215.8 (-5.35%)	213.7 (-6.29%)	212.6 (-6.76%)	211.4 (-7.28%)	207.5 (-9.01%)
(e) HU					
BL	287.4				
	Number of classes				BL+{1k–20k}
	1k	5k	10k	20k	
HAL	**263.8 (-8.24%)**	**261.6 (-8.98%)**	**262.5 (-8.67%)**	**268.3 (-6.65%)**	**251.4 (-12.54%)**
S-HAL	277.3 (-3.53%)	275.4 (-4.20%)	274.8 (-4.38%)	274.6 (-4.45%)	272.6 (-5.18%)
COALS	273.5 (-4.85%)	271.4 (-5.58%)	271.0 (-5.73%)	274.7 (-4.43%)	262.0 (-8.85%)
S-COALS	280.3 (-2.48%)	277.6 (-3.43%)	276.8 (-3.72%)	275.1 (-4.30%)	273.0 (-5.03%)
RI	273.6 (-4.80%)	270.8 (-5.78%)	270.2 (-6.01%)	274.3 (-4.56%)	263.4 (-8.38%)
S-RI	278.6 (-3.08%)	277.0 (-3.61%)	276.3 (-3.86%)	275.9 (-4.02%)	274.1 (-4.64%)
(f) EN					
BL	65.2				
	Number of classes				BL+{1k–20k}
	1k	5k	10k	20k	

HAL	**61.9 (-5.16%)**	**62.4 (-4.35%)**	**62.6 (-3.99%)**	63.42 (-2.80%)	**60.6 (-7.11%)**
S-HAL	62.8 (-3.74%)	63.1 (-3.29%)	63.3 (-3.03%)	63.5 (-2.62%)	62.4 (-4.37%)
COALS	62.8 (-3.68%)	63.3 (-3.04%)	63.5 (-2.73%)	64.0 (-1.93%)	61.7 (-5.45%)
S-COALS	63.4 (-2.87%)	63.7 (-2.38%)	63.8 (-2.27%)	64.0 (-1.92%)	62.8 (-3.72%)
RI	62.7 (-3.83%)	63.0 (-3.52%)	63.1 (-3.28%)	**63.4 (-2.82%)**	61.7 (-5.44%)
S-RI	63.1 (-3.31%)	63.3 (-3.01%)	63.4 (-2.85%)	63.6 (-2.52%)	62.6 (-3.99%)

Global semantic language models are shown in Table 5. The word unigram probability given by the LDA and S-LDA models with the number of topics ranging between 20 and 1,000 is interpolated with the baseline model. We can see similar improvements compared to baseline for all six languages including English. In the case of LDA the best results are achieved with 300 topics on average. LDA can be trained well on English corpora even for higher a number of topics (because of lower significance of data sparsity). We can see that S-LDA models always performed better than unstemmed versions. Moreover, thanks to stemming, it is possible to infer the more distinct latent topics (as was expected mainly for inflectional languages). S-LDA performs almost 3% better than the LDA model in the modelling of inflectional languages. For English, both models are similar. By comparison with local semantics, the global semantics improves the language modelling slightly more (up to 16%).

Table 5: Table displays the perplexities of the LDA and S-LDA language model (denoted as BL+LDA and BL+S-LDA, respectively) when combined with baseline according to different number of latent topics. Row denoted as BL represents perplexities of baseline.

	CZ	SL	SK	PL	HU	EN
BL	217.0	168.8	200.7	228.0	287.4	65.2
topics	BL+LDA					
20	198.5 (-8.53%)	155.1 (-8.12%)	184.1 (-8.27%)	209.4 (-8.17%)	264.0 (-8.14%)	60.5 (-7.30%)
50	194.3 (-10.47%)	151.8 (-10.07%)	180.4 (-10.13%)	205.2 (-10.00%)	259.3 (-9.79%)	59.5 (-8.76%)
100	190.3 (-12.32%)	149.0 (-11.74%)	176.6 (-11.99%)	201.4 (-11.67%)	255.3 (-11.19%)	58.7 (-9.97%)
200	188.3 (-13.24%)	**146.6 (-13.16%)**	174.6 (-12.99%)	**199.0 (-12.71%)**	255.3 (-11.19%)	57.8 (-11.42%)
300	**187.9 (-13.43%)**	146.8 (-13.02%)	**174.6 (-13.00%)**	199.4 (-12.57%)	255.6 (-11.08%)	57.3 (-12.20%)
400	188.1 (-13.33%)	146.8 (-13.02%)	174.6 (-12.99%)	199.8 (-12.38%)	**254.5 (-11.47%)**	56.9 (-12.81%)
500	188.4 (-13.19%)	147.1 (-12.88%)	174.8 (-12.88%)	199.8 (-12.38%)	254.6 (-11.43%)	56.5 (-13.35%)
600	188.3 (-13.25%)	147.1 (-12.85%)	175.2 (-12.68%)	199.6 (-12.47%)	255.1 (-11.23%)	56.4 (-13.61%)
700	188.8 (-13.00%)	147.2 (-12.81%)	175.3 (-12.66%)	200.0 (-12.28%)	255.2 (-11.20%)	**56.3 (-13.66%)**
800	188.5 (-13.14%)	147.4 (-12.67%)	175.3 (-12.64%)	199.7 (-12.44%)	255.1 (-11.26%)	56.4 (-13.61%)
900	188.8 (-13.02%)	147.5 (-12.66%)	175.6 (-12.52%)	200.5 (-12.08%)	255.2 (-11.23%)	56.4 (-13.52%)
1000	189.2 (-12.82%)	147.8 (-12.47%)	175.8 (-12.40%)	199.9 (-12.31%)	255.2 (-11.23%)	56.4 (-13.50%)
topics	BL+S-LDA					
20	198.7 (-8.43%)	154.7 (-8.35%)	184.3 (-8.15%)	209.3 (-8.22%)	263.3 (-8.41%)	60.5 (-7.26%)
50	194.4 (-10.45%)	151.8 (-10.08%)	180.4 (-10.13%)	205.3 (-9.97%)	258.0 (-10.24%)	59.7 (-8.50%)
100	190.2 (-12.35%)	148.8 (-11.85%)	176.8 (-11.91%)	201.6 (-11.59%)	252.5 (-12.17%)	58.9 (-9.74%)
200	185.8 (-14.39%)	146.1 (-13.48%)	172.6 (-13.98%)	197.4 (-13.45%)	247.2 (-13.99%)	58.0 (-11.03%)
300	183.1 (-15.66%)	144.1 (-14.65%)	170.3 (-15.11%)	194.6 (-14.67%)	**245.5 (-14.58%)**	57.5 (-11.85%)
400	**182.0 (-16.15%)**	142.7 (-15.47%)	**168.7 (-15.94%)**	**193.1 (-15.30%)**	246.1 (-14.39%)	57.2 (-12.41%)
500	182.5 (-15.93%)	**141.9 (-15.95%)**	168.9 (-15.82%)	193.2 (-15.28%)	247.1 (-14.04%)	56.8 (-12.95%)
600	182.8 (-15.79%)	142.4 (-15.68%)	169.3 (-15.62%)	193.7 (-15.04%)	246.8 (-14.12%)	56.6 (-13.23%)
700	183.4 (-15.49%)	142.7 (-15.48%)	169.8 (-15.41%)	194.2 (-14.84%)	247.5 (-13.89%)	56.4 (-13.57%)
800	183.6 (-15.42%)	142.9 (-15.34%)	170.1 (-15.22%)	194.6 (-14.64%)	247.6 (-13.87%)	56.3 (-13.76%)
900	184.0 (-15.21%)	143.3 (-15.14%)	170.2 (-15.20%)	194.7 (-14.59%)	247.0 (-14.05%)	56.2 (-13.91%)
1000	184.1 (-15.19%)	143.4 (-15.04%)	170.4 (-15.10%)	195.3 (-14.33%)	247.9 (-13.76%)	**56.2 (-13.93%)**

The most important experiments are shown in Table 6, where a combination of different sources of information is depicted. As the best-performing model for local semantics, the HAL model was chosen (a combination of HAL-based language models of 1,000, 5,000, 10,000, and 20,000 clusters). Global semantics in language models was best modelled by S-LDA with 400 topics. The stem-based model (HPS) is also added together with the baseline (BL) and the final model is thus a combination of all seven language models (the last two columns in the table, where we finally compare linear interpolation with a bucketed

linear interpolation).

From the table we can clearly state that all three sources of information (morphology, local semantics, and global semantics) significantly enrich each other (which is especially evident for inflectional languages). Their bucketed linear combination leads to improvements of up to almost 26% for inflectional languages and up to 15% for English compared to the stand-alone baseline. Our language models perform similarly for all inflectional languages (improvements by each part are quite similar). We can also see that the bucketed linear interpolation (BLI-all) produces slightly better results than simple linear interpolation (LI-all).

Table 6: Perplexity results by combining different sources of information. The operator + denotes the linear interpolation of appropriate models. The S-LDA model uses 400 topics, and HAL denotes all four class-based language models based on HAL (1k, 5k, 10k, and 20k clusters). The last two columns are the most important, where all models (BL+HPS+HAL+S-LDA) are combined by linear interpolation (LI-all) and bucketed linear interpolation (BLI-all).

	BL	BL + HPS	BL+HAL	BL+HAL+HPS	BL+S-LDA	LI-all	BLI-all
CZ	217.0	201.6 (-7.11%)	191.0 (-12.00%)	183.3 (-15.53%)	182.0 (-16.15%)	167.1 (-23.03%)	**165.4 (-23.79%)**
SL	168.8	157.4 (-6.77%)	146.7 (-13.09%)	141.6 (-16.11%)	142.7 (-15.47%)	130.7 (-22.56%)	**128.1 (-24.11%)**
SK	200.7	186.3 (-7.17%)	174.0 (-13.27%)	167.3 (-16.63%)	168.7 (-15.94%)	153.5 (-23.51%)	**151.0 (-24.75%)**
PL	228.0	209.9 (-7.94%)	195.7 (-14.18%)	187.4 (-17.81%)	193.1 (-15.30%)	173.7 (-23.82%)	**171.1 (-24.97%)**
HU	287.4	252.5 (-12.17%)	251.4 (-12.54%)	233.6 (-18.74%)	246.1 (-14.39%)	216.2 (-24.79%)	**213.4 (-25.77%)**
EN	65.2	64.3 (-1.42%)	60.6 (-7.11%)	60.3 (-7.62%)	57.2 (-12.41%)	55.7 (-14.60%)	**55.4 (-15.10%)**

In order to visualize what weights in the linear interpolation (LI-all model) are allocated for each sub-model by the EM algorithm, we render Figure 3, where our final model created from seven sub-models is shown. We can see that the baseline always has the highest weight, and then, in order of decreasing weight, S-LDA, HPS, HAL{20k}, HAL{10k}, HAL{5k} and HAL{1k} follow, where the numbers in brackets mean the numbers of clusters. A direct correlation between weights and perplexity improvements can be seen. The bigger the weights allocated by the EM algorithm, the greater the improvement in perplexity achieved.

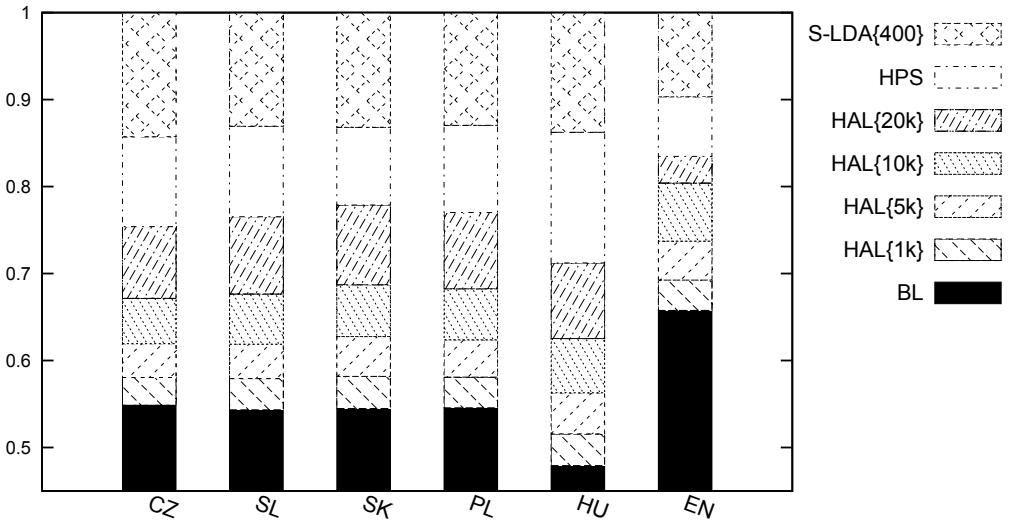


Figure 3: Interpolation weights of each sub-model in the final linear interpolation.

5.3. Machine translation results

This section describes the performance of the proposed language models in terms of a machine translation task. Success in this task should verify the ability of the models to improve the performance of a real-world application. The system used in this test is based upon the statistical machine translation toolkit called

Moses[3], briefly described in [Koehn et al., 2007].

Europarl parallel corpora were used for training and evaluation of machine translation (see Subsection 5.1). Table 7 shows the settings for translation model training.

Table 7: Parameters used for training the machine translation system.

Casing normalization	True casing
Minimum-maximum tokes per sentence	1-80
Language model order	4 (binary)
Language model smoothing	Modified Kneser-Ney
Language model toolkit	IRSTLM
Alignment heuristic	grow-diag-final-and
Reordering model	msd-bidirectional-fe
Tuning	MERT

Our translation experiment consists in measuring the difference in translation performance when the standard four-gram language model is used and when our improved language model is employed. Language models are used during training as well as during decoding in Moses. The best approach would be to replace the standard model with our model. However, our model does not support left-to-right decoding because global semantic models require knowledge of word contexts. This prevents the use of our model for training as well as directly for decoding. Instead, we generate 5,000 best hypotheses and re-score them with our model. Such a procedure is not as effective as direct incorporation of the model into Moses; however, it is the only possible approach.

The change of the language model in Moses is not possible without re-computing the model weights. Moses uses weights for translation, reordering, word penalty, and language models. The weights are set during the optimization phase by the MERT (Minimum Error Rate Training) algorithm [Och, 2003]. The algorithm optimizes the weights for best translation scores on the held-out data. However, such weights are valid only for the original model. A different model returns different probability estimates and thus they play different roles during interpolation of the final translation probability.

We estimate the correct weight for our improved language model with the same algorithm, MERT. We use the n-best list generated during the optimization phase. We replace the probability estimates of the original language model with the estimates from our model. Then, we run the optimization procedure, which returns news weights including the weight for our improved language model. These new weights are used for translating test sentences.

The results of our translation experiment are shown in Table 8. We measure the translation scores with the BLEU metric [Papineni et al., 2002]. This metric is based on the ratio of n-gram overlaps with reference translations. We compare original translations from Moses with translations obtained by re-scoring them with our language model. We generate and re-score 5000 hypotheses for each translated sentence. The most probable hypothesis is taken as the translation result. Beside showing scores for the whole language model composed of all sub-models (local, global semantics, and stemming), we also study the performance of language models composed of various combinations of sub-models. Numbers in brackets denote the absolute improvements in BLEU score.

[3]Available at http://www.statmt.org/moses/.

Table 8: BLEU scores achieved by combination of different language models. Arrows show the direction of translation. BL denotes the baseline language model (the standard output from the Moses). HPS is a stem-based language model. HAL denotes the use of all four HAL-based language models (1k, 5k, 10k, and 20k clusters). S-LDA denotes language model based on stemmed version of LDA using 400 topics. Linear interpolation of all seven sub-models is denotes as LI-all.

	BL	BL + HPS	BL+HAL	BL+HAL+HPS	BL+S-LDA	LI-all
EN → CZ	26.02	26.24 (+0.22)	26.64 (+0.62)	26.75 (+0.73)	26.34 (+0.32)	**26.93 (+0.91)**
EN → SL	28.58	28.77 (+0.19)	29.07 (+0.49)	29.17 (+0.59)	28.75 (+0.17)	**29.32 (+0.74)**
EN → SK	27.04	27.20 (+0.16)	27.60 (+0.56)	27.69 (+0.65)	27.26 (+0.22)	**27.79 (+0.75)**
EN → PL	23.70	23.86 (+0.16)	24.13 (+0.43)	24.27 (+0.57)	23.95 (+0.25)	**24.38 (+0.68)**
EN → HU	18.51	19.00 (+0.49)	19.19 (+0.68)	19.39 (+0.88)	19.02 (+0.51)	**19.47 (+0.96)**
CZ → EN	37.29	37.42 (+0.13)	37.70 (+0.41)	37.73 (+0.44)	37.75 (+0.46)	**37.97 (+0.68)**

We can see that each of the three sources of information (morphology, local semantics, and global semantics) gives at least some improvement. Their combination again enriches each of them even in machine translation tests. The highest improvements are always achieved by HAL (combination of 1k, 5k, 10k, and 20k clusters), which is the best representative of local semantics models. HPS and S-LDA provide similar improvements. Stemming is again useless for English. The fact that S-LDA performs worse than HAL is probably caused by working on the sentence level (not document level), on which Moses is focused. If the S-LDA model had wider context (i.e. the whole document) it would probably perform better as was shown in Subsection 5.2. The combination of all sources of information (the last column in the table) studied in this paper leads to signification improvements in BLEU score, which is a solid proof that the proposed language models are usable and effective in a real-world application.

6. Discussion

In this section we summarize our results and discuss the behaviour of our methods. In previous sections we presented various experiments on our language models. We experimented with a combination of three different sources of information (morphology, local semantics, and global semantics). Firstly, we tested language models from the theory of information point of view, where the perplexity was used as an evaluation measure. The results of these tests were followed by evaluation in a real-world application, where machine translation with the Moses system proved the quality of our methods.

First, let us look at the perplexity results. Stem-based language models (HPS) proved to be efficient for inflectional languages. The average relative improvement in perplexity for Slavic languages (Czech, Slovenian, Slovak, and Polish) was about 7%; for Hungarian it was as much as 12%. As we expected, stemming was found to be useless for English, with almost no improvement in perplexity.

Semantic spaces (HAL, COALS, and RI) and their stemmed versions (S-HAL, S-COALS, and S-RI) were explored as a next step, where we created class-based language models according to different numbers of clusters (i.e. 1,000, 5,000, 10,000, and 20,000). The HAL model performed best in the majority of experiments. It was found to be better not to use stemming as the results of stemmed versions of semantic spaces were almost always worse than those of unstemmed models. Stemming was often better only in cases where sparse clusters (20,000) were used. We suppose this is caused by the nature of local semantic models, which thanks to the short context, also incorporate morpho-syntactic information of words that is very useful especially in language models. In contrast, stemming inhibits this kind of information. The combination of class-based language models of different numbers of clusters was shown to improve language modelling by approximately 13% on average for inflectional languages and by 7% for English.

In our previous research [Brychcín and Konopík, 2014], HAL was also best for 1,000, 5,000, or 10,000 clusters, but the COALS model performed better than HAL in the case of 20,000 or more clusters. Our current results confirm that for inflectional languages the difference between HAL and COALS is lower with a growing number of clusters; moreover in the case of 20,000 clusters, the performance of COALS was slightly better. In the cases of Hungarian and English, HAL performed best all the time.

The third source of information was global semantics modelled by LDA and S-LDA (the stemmed version of LDA). LDA has already been proven by many researches to be useful in language modelling (see, e.g.,

[Tam and Schultz, 2005, 2006; Watanabe et al., 2011]) and our results agree with that. LDA improved the perplexity of models of all languages by about 13% on average. Moreover, our stemmed extension of LDA was able to achieve even better results for inflectional languages (improvements of up to 16%). Here, the situation is opposite to that of local semantics. LDA is based on bag-of-word hypotheses, where the word order is inhibited (the morpho-syntactic information is useless for topic inference). Stemming thus helps to deal with the data sparsity problem better.

The most important finding, which is also the aim of this paper, is that the investigated sources of information mutually help each other in terms of improving language modelling. Their combination was able to dramatically reduce the perplexities of language models. By grouping HPS, HAL, and S-LDA with a baseline we achieved approximately 25% improvement on average for all inflectional languages and 15% for English, which is a very satisfactory result.

The same models were also investigated in a machine translation task, where we measured BLEU scores. Only by changing the language model was significant growth of the BLEU score achieved (the average improvement among all languages was 0.8 BLEU points). If we compare our results with different works on the same corpora (e.g. works of [Virpioja et al., 2007; Sanchis-trilles et al., 2010]), we can claim that our models are very effective.

The results from machine translation correlate with the perplexity results but there are some deviations in improvements of inflectional languages even though the improvements in perplexities were almost identical. The performance of the machine translation system depends on several modules (not only on the language model) as well as on the optimization of parameters on held-out data (using MERT). The difficulty of the language must also be taken into account, as we observe proportionally different perplexities and different BLEU scores among all languages despite using the corpora within the same domain.

The fact that we significantly improved the modelling of several languages with different properties (different families of languages) testifies to the quality and multilingualism of our methods. Moreover, these methods are attractive due to their unsupervised nature and so they can be easily applied to other languages or tasks.

7. Summary

7.1. Future work

As shown in Section 2, there is a huge number of methods for latent semantics discovery that are worth investigating. These methods use various architectures, e.g. matrix factorization methods, graphical models, or neural networks. It is beyond the scope of any paper to compare them all. This is the main direction for the future work as we believe that other combinations may produce even better language models.

Another alternative for future work consists in studying machine translation more deeply. The experiments with the Moses machine translation framework revealed that it is an extraordinary framework with great extension capabilities. We expect that through the direct link between Moses and the methods investigated in this paper (local and global semantics or stemming), even higher improvements could be achieved. Moses supports several architectures for applying various sources of information. For example clusters given by semantic spaces, stems, or topics can be directly used in factored translation.

In addition we also want to test our methods in different NLP tasks (e.g. speech recognition, optical character recognition, spelling correction, etc.).

7.2. Conclusion

In this article we experimented with three kinds of various information sources whose application into language modelling yields significant improvements in model prediction ability. The beauty of our method is that all the information comes directly from the data. Nothing is added externally. The information we use is sometimes called *latent* or *hidden* because an algorithm must be employed to discover it. We use tree approaches for the discovery of hidden information. Topic models discover long semantic relations between words and the documents where they are used. Semantic spaces (HAL, COALS, RI) work with

short semantic relations between words and their direct contexts. The stemming studies the information hidden directly in different forms of words.

All these sources were previously used in a research (semantic spaces in our preceding work). However, nobody had studied whether their combination carries some extra information. We conclusively proved that the combination provides much higher perplexity reduction than individual models do.

Our stemming algorithm (HPS) proved to be very useful in language modelling of inflectional languages. The use of stems as classes for class-based language models or extending LDA with stem information was shown to be very effective, as we achieved significant improvements in language modelling. Taking into account the results and our findings during testing, we can recommend HAL for modelling local semantics and S-LDA for modelling global semantics.

We believe that our article carries a potential beyond language models. All the methods investigated in this paper are based upon unsupervised training. Thus, they can be easily applied to different NLP tasks.

Acknowledgement

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

Arsoy, E., Dutaac, H., Arslan, L. M., 2006. A unified language model for large vocabulary continuous speech recognition of turkish. Signal Processing 86 (10), 2844 – 2862.

Bahl, L. R., Jelinek, F., Mercer, R. L., 1983. A maximum likelihood approach to continuous speech recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-5 (2), 179 –190.

Bellegarda, J. R., Aug. 2000. Exploiting latent semantic information in statistical language modeling. Proceedings of the IEEE 88 (8), 1279 –1296.

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., Mar. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.
URL http://dl.acm.org/citation.cfm?id=944919.944966

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

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.

Brychcín, T., Konopík, M., 2011. Morphological based language models for inflectional languages. In: Proceedings of IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems.

Brychcín, T., Konopík, M., 2014. Semantic spaces for improving language modeling. Computer Speech & Language 28 (1), 192 – 209.

Brychcín, T., Konopík, M., 2015. Hps: High precision stemmer. Information Processing & Management 51 (1), 68 – 91.
URL http://www.sciencedirect.com/science/article/pii/S0306457314000843

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

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

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.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B 39 (1), 1–38.

Deschacht, K., Belder, J. D., Moens, M.-F., 2012. The latent words language model. Computer Speech & Language 26 (5), 384 – 409.

Dinu, G., Lapata, M., 2010. Measuring distributional similarity in context. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. EMNLP '10. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1162–1172.
URL http://dl.acm.org/citation.cfm?id=1870658.1870771

Erk, K., Padó, S., 2010. Exemplar-based models for word meaning in context. In: Proceedings of the ACL 2010 Conference Short Papers. ACLShort '10. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 92–97.
URL http://dl.acm.org/citation.cfm?id=1858842.1858859

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

Gildea, D., Hofmann, T., 1999. Topic-based language models using em. In: Proceedings of Eurospeech. pp. 2167–2170.

Griffiths, T. L., Steyvers, M., Apr. 2004. Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101 (Suppl 1), 5228–5235.

Habernal, I., Brychcín, T., 2013. Semantic spaces for sentiment analysis. In: Proceedings of the 16th international conference on Text, Speech and Dialogue (TSD13).

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

Huang, E. H., Socher, R., Manning, C. D., Ng, A. Y., 2012. Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1. ACL '12. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 873–882.
URL http://dl.acm.org/citation.cfm?id=2390524.2390645

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

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.

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

Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., Stolcke, A., Oct. 2006. Morphology-based language modeling for conversational Arabic speech recognition. Computer Speech & Language 20 (4), 589–608.

Koehn, P., 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In: Machine Translation Summit X. Phuket, Thailand, pp. 79–86.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., 2007. Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. ACL '07. Association for Computational Linguistics, pp. 177–180.

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

Luong, T., Socher, R., Manning, C., August 2013. Better word representations with recursive neural networks for morphology. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, Sofia, Bulgaria, pp. 104–113.
URL http://www.aclweb.org/anthology/W13-3512

Martin, S., Liermann, J., Ney, H., 1998. Algorithms for bigram and trigram word clustering. Speech Communication 24 (1), 19 – 37.

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

McNamara, D. S., 2011. Computational methods to extract meaning from text and advance theories of human cognition. Topics in Cognitive Science 3 (1), 3–17.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.
URL http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781

Mikolov, T., Karafit, M., Burget, L., ernock, J., Khudanpur, S., 2010. Recurrent neural network based language model. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010). Vol. 2010. International Speech Communication Association, pp. 1045–1048.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=9362

Mikolov, T., Zweig, G., 2012. Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012. pp. 234–239.
URL http://dx.doi.org/10.1109/SLT.2012.6424228

Mitchell, J., Lapata, M., 2009. Language models based on semantic composition. In: In Proceedings of EMNLP. pp. 430–439.

Och, F. J., 2003. Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. ACL '03. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 160–167.

Oikonomidis, D., Digalakis, V., 2003. Stem-based maximum entropy language models for inflectional languages. In: INTERSPEECH. ISCA.

Oparin, I., 2008. Language models for automatic speech recognition of inflectional languages. Ph.D. thesis, University of West Bohemia, Pilsen.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02. Association for Computational Linguistics, pp. 311–318.

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

Reisinger, J., Mooney, R., 2010a. A mixture model with sharing for lexical semantics. In: Proceedings of the 2010 Conference on

Empirical Methods in Natural Language Processing. EMNLP '10. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1173–1182.
URL http://dl.acm.org/citation.cfm?id=1870658.1870772

Reisinger, J., Mooney, R. J., 2010b. Multi-prototype vector-space models of word meaning. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. HLT '10. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 109–117.
URL http://dl.acm.org/citation.cfm?id=1857999.1858012

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 (2), 303–345.

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.

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

Sanchis-trilles, G., Cettolo, M., Kessler, F. B., 2010. Online language model adaptation via n-gram mixtures for statistical machine translation. In: Proceedings of the 14th Annual Conference of the European Association for Machine Translation.

Schwenk, H., Jul. 2007. Continuous space language models. Comput. Speech Lang. 21 (3), 492–518.
URL http://dx.doi.org/10.1016/j.csl.2006.09.003

Tam, Y., Schultz, T., 2005. Dynamic language model adaptation using variational bayes inference. In: Proceedings of Interspeech. pp. 5–8.

Tam, Y., Schultz, T., 2006. Unsupervised language model adaptation using latent semantic marginals. In: Proceedings of Interspeech.

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

Virpioja, S., Vyrynen, J. J., Creutz, M., Sadeniemi, M., 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In: PROC. OF MT SUMMIT XI. pp. 491–498.

Wang, S., Schuurmans, D., Peng, F., Zhao, Y., 2003. Semantic n-gram language modeling with the latent maximum entropy principle. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP03).

Watanabe, S., Iwata, T., Hori, T., Sako, A., Ariki, Y., April 2011. Topic tracking language model for speech recognition. Computer Speech & Language 25, 440–461.

Whittaker, E., Woodland, P., 2003. Language modelling for russian and english using words and classes. Computer Speech & Language 17 (1), 87 – 104.

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