Vector Space Models for Phrase-based Machine Translation

Tamer Alkhouli[1], **Andreas Guta**[1], **and Hermann Ney**[1,2]
[1]Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
[2]Spoken Language Processing Group
Univ. Paris-Sud, France and LIMSI/CNRS, Orsay, France
{surname}@cs.rwth-aachen.de

Abstract

This paper investigates the application of vector space models (VSMs) to the standard phrase-based machine translation pipeline. VSMs are models based on continuous word representations embedded in a vector space. We exploit word vectors to augment the phrase table with new inferred phrase pairs. This helps reduce out-of-vocabulary (OOV) words. In addition, we present a simple way to learn bilingually-constrained phrase vectors. The phrase vectors are then used to provide additional scoring of phrase pairs, which fits into the standard log-linear framework of phrase-based statistical machine translation. Both methods result in significant improvements over a competitive in-domain baseline applied to the Arabic-to-English task of IWSLT 2013.

1 Introduction

Categorical word representation has been widely used in many natural language processing (NLP) applications including statistical machine translation (SMT), where words are treated as discrete random variables. Continuous word representations, on the other hand, have been applied successfully in many NLP areas (Manning et al., 2008; Collobert and Weston, 2008). However, their application to machine translation is still an open research question. Several works tried to address the question recently (Mikolov et al., 2013b; Zhang et al., 2014; Zou et al., 2013), and this work is but another step in that direction.

While categorical representations do not encode any information about word identities, continuous representations embed words in a vector space, resulting in geometric arrangements that reflect information about the represented words. Such embeddings open the potential for applying information retrieval approaches where it becomes possible to define and compute similarity between different words. We focus on continuous representations whose training is influenced by the surrounding context of the token being represented. One motivation for such representations is to capture word semantics (Turney et al., 2010). This is based on the *distributional hypothesis* (Harris, 1954) which says that words that occur in similar contexts tend to have similar meanings.

We make use of continuous vectors learned using simple neural networks. Neural networks have been gaining increasing attention recently, where they have been able to enhance strong SMT baselines (Devlin et al., 2014; Sundermeyer et al., 2014). While neural language and translation modeling make intermediate use of continuous representations, there have been also attempts at explicit learning of continuous representations to improve translation (Zhang et al., 2014; Gao et al., 2013).

This work explores the potential of word semantics based on continuous vector representations to enhance the performance of phrase-based machine translation. We present a greedy algorithm that employs the phrase table to identify phrases in a training corpus. The phrase table serves to bilingually restrict the phrases spotted in the monolingual corpus. The algorithm is applied separately to the source and target sides of the training data, resulting in source and target corpora of phrases (instead of words). The phrase corpus is used to learn phrase vectors using the same methods that produce word vectors. The vectors are then used to provide semantic scoring of phrase pairs. We also learn *word* vectors and employ them to augment the phrase table with paraphrased entries. This leads to a reduction in

Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 1–10,
October 25, 2014, Doha, Qatar.

the OOV rate which translates to improved BLEU and and TER scores. We apply the two methods on the IWSLT 2013 Arabic-to-English task and show significant improvements over a strong in-domain baseline.

The rest of the paper is structured as follows. Section 2 presents a background on word and phrase vectors. The construction of the phrase corpus is discussed in Section 3, while Section 4 demonstrates how to use word and phrase vectors in the standard phrase-based SMT pipeline. Experiments are presented in Section 5, followed by an overview of the related word in Section 6, and finally Section 7 concludes the work.

2 Vector Space Models

One way to obtain context-based word vectors is through a neural network (Bengio et al., 2003; Schwenk, 2007). With a vocabulary size V, one-hot encoding of V-dimensional vectors is used to represent input words, effectively associating each word with a D-dimensional vector in the $V \times D$ input weight matrix, where D is the size of the hidden layer. Similarly, one-hot encoding on the output layer associates words with vectors in the output weight matrix.

Alternatively, a count-based V-dimensional word co-occurrence vector can serve as a word representation (Lund and Burgess, 1996; Landauer and Dumais, 1997). Such representations are sparse and high-dimensional, which might require an additional dimensionality reduction step (e.g. using SVD). In contrast, learning word representations via neural models results directly in relatively low-dimensional, dense vectors. In this work, we follow the neural network approach to extract the feature vectors. Whether word vectors are extracted by means of a neural network or co-occurrence counts, the context surrounding a word influences its final representation by design. Such context-based representations can be used to determine semantic similarities.

The construction of *phrase* representations, on the other hand, can be done in different ways. The compositional approach constructs the vector representation of a phrase by resorting to its constituent words (or sub-phrases) (Gao et al., 2013; Chen et al., 2010). Kalchbrenner and Blunsom (2013) obtain continuous sentence representations by applying a sequence of convolutions, starting with word representations.

Another approach for phrase representation considers phrases as atomic units that can not be divided further. The representations are learned directly in this case (Mikolov et al., 2013b; Hu et al., 2014).

In this work, we follow the second approach to obtain phrase vectors. To this end, we apply the same methods that yield word vectors, with the difference that phrases are used instead of words. In the case of neural word representations, a neural network that is presented with words at the input layer is presented with phrases instead. The resulting vocabulary size in this case would be the number of distinct phrases observed during training. Although learning phrase embeddings directly is amenable to data sparsity issues, it provides us with a simple means to build phrase vectors making use of tools already developed for word vectors, focussing the effort on preprocessing the data as will be discussed in the next section.

3 Phrase Corpus

When training word vectors using neural networks, the network is presented with a corpus. To build phrase vectors, we first identify phrases in the corpus and generate a *phrase corpus*. The phrase corpus is similar to the original corpus except that its words are joined to make up phrases. The new corpus is then used to train the neural network. The columns of the resulting input weight matrix of the network are the phrase vectors corresponding to the phrases encountered during training.

Mikolov et al. (2013b) identify phrases using a monolingual point-wise mutual information criterion with discounting. Since our end goal is to generate phrase vectors that are helpful for translation, we follow a different approach: we constrain the phrases by the conventional phrase table of phrase-based machine translation. This is done by limiting the phrases identified in the corpus to high quality phrases occurring in the phrase table. The quality is determined using bilingual scores of phrase pairs. While the phrase vectors of a language are eventually obtained by training the neural network on the monolingual phrase corpus of that language, the reliance on bilingual scores to

Algorithm 1 Phrase Corpus Construction

1: $p \leftarrow 1$
2: **for** $p \leq numPasses$ **do**
3: $\quad i \leftarrow 2$
4: $\quad$ **for** $i \leq corpus.size - 1$ **do**
5: $\quad\quad \tilde{w} \leftarrow join(t_i, t_{i+1})$ ▷ create a phrase using the current and next tokens
6: $\quad\quad \tilde{v} \leftarrow join(t_{i-1}, t_i)$ ▷ create a phrase using the previous and current tokens
7: $\quad\quad joinForward \leftarrow score(\tilde{w})$
8: $\quad\quad joinBackward \leftarrow score(\tilde{v})$
9: $\quad\quad$ **if** $joinForward \geq joinBackward$ **and** $joinForward \geq \theta$ **then**
10: $\quad\quad\quad t_i \leftarrow \tilde{w}$
11: $\quad\quad\quad$ remove t_{i+1}
12: $\quad\quad\quad i \leftarrow i+2$ ▷ newly created phrase not available for further merge during current pass
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ **if** $joinBackward > joinForward$ **and** $joinBackward \geq \theta$ **then**
15: $\quad\quad\quad\quad t_{i-1} \leftarrow \tilde{v}$
16: $\quad\quad\quad\quad$ remove t_i
17: $\quad\quad\quad\quad i \leftarrow i+2$ ▷ newly created phrase not available for further merge during current pass
18: $\quad\quad\quad$ **else**
19: $\quad\quad\quad\quad i \leftarrow i+1$
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: $\quad p \leftarrow p+1$
24: **end for**

construct the monolingual phrase corpus encodes bilingual information in the corpus, namely, the corpus will include phrases that having a matching phrase in the other language, which is in line with the purpose for which the phrases are constructed, that is, their use in the phrase-based machine translation pipeline which is explained in the next section. In addition, the aforementioned scoring serves to exclude noisy phrase-pair entries during the construction of the phrase corpus. Next, we explain the details of the construction algorithm.

3.1 Phrase Spotting

We propose Algorithm 1 as a greedy approach for phrase corpus construction. It is a multi-pass algorithm where each pass can extend tokens obtained during the previous pass by a single token at most. Before the first pass, all tokens are words. During the passes the tokens might remain as words or can be extended to become phrases. Given a token t_i at position i, a scoring function is used to score the phrase (t_i, t_{i+1}) and the phrase (t_{i-1}, t_i). The phrase having a higher score is adopted as long as its score exceeds a predefined threshold θ. The scoring function used in lines 7 and 8 is based on the phrase table. If the phrase does not belong to the phrase table it is given a score $\theta' < \theta$. If the phrase exists, a bilingual score is computed using the phrase table fields as follows:

$$score(\tilde{f}) = \max_{\tilde{e}} \left\{ \sum_{i=1}^{L} w_i g_i(\tilde{f}, \tilde{e}) \right\} \qquad (1)$$

where $g_i(\tilde{f}, \tilde{e})$ is the ith feature of the bilingual phrase pair $(\tilde{f}, \tilde{e})$. The maximization is carried out over all phrases $\tilde{e}$ of the other language. The score is the weighted sum of the phrase pair features. Throughout our experiments, we use 2 phrasal and 2 lexical features for scoring, with manual tuning of the weights w_i.

The resulting corpus is then used to train phrase vectors following the same procedure of training word vectors.

4 End-to-end Translation

In this section we will show how to employ phrase vectors in the phrase-based statistical machine translation pipeline.

4.1 Phrase-based Machine Translation

The phrase-based decoder consists of a search using a log-linear framework (Och and Ney, 2002) as follows:

$$\hat{e}_1^{\hat{I}} = arg \max_{I,e_1^I} \left\{ \max_{K,s_1^K} \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \quad (2)$$

where $e_1^I = e_1 ... e_I$ is the target sentence, $f_1^J = f_1 ... f_J$ is the source sentence, $s_1^K = s_1 ... s_K$ is the hidden alignment or derivation. The models $h_m(e_1^I, s_1^K, f_1^J)$ are weighted by the weights λ_m which are tuned using minimum error rate training (MERT) (Och, 2003). The rest of the section presents two ways to integrate vector representations into the system described above.

4.2 Semantic Phrase Feature

Words that occur in similar contexts tend to have similar meanings. This idea is known as the *distributional hypothesis* (Harris, 1954), and it motivates the use of word context to learn word representations that capture word semantics (Turney et al., 2010). Extending this notion to phrases, phrase vectors that are learned based on the surrounding context encode phrase semantics. Since we will use phrase vectors to compute a feature of a phrase pair in the following, we refer to the feature as a semantic phrase feature.

Given a phrase pair $(\tilde{f}, \tilde{e})$, we can use the phrase vectors of the source and target phrases to compute a semantic phrase feature as follows:

$$h_{M+1}(\tilde{f}, \tilde{e}) = sim(W x_{\tilde{f}}, z_{\tilde{e}}) \quad (3)$$

where *sim* is a similarity function, $x_{\tilde{f}}$ and $z_{\tilde{e}}$ are the S-dimensional source and T-dimensional target vectors respectively corresponding to the source phrase $\tilde{f}$ and target phrase $\tilde{e}$. W is an $S \times T$ linear projection matrix that maps the source space to the target space (Mikolov et al., 2013a). The matrix is estimated by optimizing the following criterion with stochastic gradient descent:

$$\min_W \sum_{i=1}^{N} ||W x_i - z_i||^2 \quad (4)$$

where the training data consists of the pairs $\{(x_1, z_1), ..., (x_N, z_N)\}$ corresponding to the source and target vectors.

Since the source and target phrase vectors are learned separately, we do not have an immediate mapping between them. As such mapping is needed for the training of the projection matrix, we resort to the phrase table to obtain it. A source and a target phrase vectors are paired if there is a corresponding phrase pair entry in the phrase table whose score exceeds a certain threshold. Scoring is computed using Eq. 1. Similarly, word vectors are paired using IBM 1 $p(e|f)$ and $p(f|e)$ lexica. Noisy entries are assumed to have a probability less than a certain threshold and are not used to pair word vectors.

4.3 Paraphrasing

While the standard phrase table is extracted using parallel training data, we propose to extend it and infer new entries relying on continuous representations. With a similarity measure (e.g. cosine similarity) that computes the similarity between two phrases, a new phrase pair can be generated by replacing either or both of its constituent phrases by similar phrases. The new phrase is referred to as a paraphrase of the phrase it replaces. This enables a richer use of the bilingual data, as a source paraphrase can be borrowed from a sentence that is not aligned to a sentence containing the target side of the phrase pair. It also enables the use of monolingual data, as the source and target paraphrases do not have to occur in the parallel data. The cross-interaction between sentences in the parallel data and the inclusion of the monolingual data to extend the phrase table are potentially capable of reducing the out-of-vocabulary (OOV) rate.

In order to generate a new phrase rule, we ensure that noisy rules do not contribute to the generation process, depending on the score of the phrase pair (cf. Eq. 1). High scoring entries are paraphrased as follows. To paraphrase the source side, we perform a k-nearest neighbor search over the source phrase vectors. The top-k similar entries are considered paraphrases of the given phrase. The same can be done for the target side. We assign the newly generated phrase pair the same feature values of the pair used to induce it. However, two extra phrase features are added: one measuring the similarity between the source phrase and its paraphrase, and another for the target phrase and its paraphrase. The new feature values for the original non-paraphrased entries are set to the

highest similarity value.

We focus on a certain setting that avoids interference with original phrase rules, by extending the phrase table to cover OOVs only. That is, source-side paraphrasing is performed only if the source paraphrase does not already occur in the phrase table. This ensures that original entries are not interfered with and only OOVs are affected during translation. Reducing OOVs by extending the phrase table has the advantage of exploiting the full decoding capabilities (e.g. LM scoring), as opposed to post-decoding translation of OOVs, which would not exhibit any decoding benefits.

The k-nearest neighbor (k-NN) approach is computationally prohibitive for large phrase tables and large number of vectors. This can be alleviated by resorting to approximate k-NN search (e.g. locality sensitive hashing). Note that this search is performed during training time to generate additional phrase table entries, and does not affect decoding time, except through the increase of the phrase table size. In our experiments, the training time using exact k-NN search was acceptable, therefore no search approximations were made.

5 Experiments

In the following we first provide an analysis of the word vectors that are later used for translation experiments. We use word vectors (as opposed to phrase vectors) for phrase table paraphrasing to reduce the OOV rate. Next, we present end-to-end translation results using the proposed semantic feature and our OOV reduction method.

The experiments are based on vectors trained using the word2vec[1] toolkit, setting vector dimensionality to 800 for Arabic and 200 for English vectors. We used the skip-gram model with a maximum skip length of 10. The phrase corpus was constructed using 5 passes, with scores computed according to Eq. 1 using 2 phrasal and 2 lexical features. The phrasal and lexical weights were set to 1 and 0.5 respectively, with all features being negative log-probabilities, and the scoring threshold θ was set to 10. All translation experiments are performed with the *Jane* toolkit (Vilar et al., 2010; Wuebker et al., 2012).

[1] https://code.google.com/p/word2vec/

5.1 Baseline System

Our phrase-based baseline system consists of two phrasal and two lexical translation models, trained using a word-aligned bilingual training corpus. Word alignment is automatically generated by GIZA++ (Och and Ney, 2003) given a sentence-aligned bilingual corpus. We also include binary count features and bidirectional hierarchical reordering models (Galley and Manning, 2008), with three orientation classes per direction resulting in six reordering models. The baseline also includes word penalty, phrase penalty and a simple distance-based distortion model.

The language model (LM) is a 4-gram mixture LM trained on several data sets using modified Kneser-Ney discounting with interpolation, and combined with weights tuned to achieve the lowest perplexity on a development set using the SRILM toolkit (Stolcke, 2002). Data selection is performed using cross-entropy filtering (Moore and Lewis, 2010).

5.2 Word Vectors

Here we analyze the quality of word vectors used in the OOV reduction experiments. The vectors are trained using an unaltered word corpus. We build a lexicon using source and target word vectors together with the projection matrix using the similarity score $sim(Wx_f, z_e))$, where the projection matrix W is used to project the source word vector x_f, corresponding to the source word f, to the target vector space. The similarity between the projection result Wx_f and the target word vector z_e is computed. In the following we will refer to these scores computed using vector representation as VSM-based scores.

The resulting lexicon is compared to the IBM 1 lexicon[2]. Given a source word, we select the the best target word according to the VSM-based score. This is compared to the best translation based on the IBM 1 probability. If both translations coincide, we refer to this as a 1-best match. We also check whether the best translation according to IBM 1 matches any of the top-5 translations based on the VSM model. A match in this case is referred to as a 5-best match.

[2] We assume for the purpose of this experiment that the IBM 1 lexicon provides perfect translations, which is not necessarily the case in practice.

corpus	Lang.	# tokens	# segments
WIT	Ar	3,185,357	147,256
UN	Ar	228,302,244	7,884,752
arGiga3	Ar	782,638,101	27,190,387
WIT	En	2,951,851	147,256
UN	En	226,280,918	7,884,752
news	En	1,129,871,814	45,240,651

Table 1: Arabic and English corpora statistics.

The vectors are trained on a mixture of in-domain data (WIT) which correspond to TED talks, and out-of-domain data (UN). These sets are provided as part of the IWSLT 2013 evaluation campaign. We include the LDC2007T40 Arabic Gigaword v3 (arGiga3) and English news crawl articles (2007 through 2012) to experiment with the effect of increasing the size of the training corpus on the quality of the word vectors. Table 1 shows the corpora statistics obtained after preprocessing.

The fractions of the 1- and 5-best matches are shown in table 2. The table is split into two halves. The upper part investigates the effect of increasing the amount of Arabic data while keeping the English data fixed (2nd row), the effect of increasing the amount of the English data while keeping the Arabic data fixed (3rd row), and the effect of using more data on both sides (4th row). The projection is done on the representation of the Arabic word f, and the similarity is computed between the projection and the representation of the English word e. In the lower half of the table, the same effects are explored, except that the projection is performed on the English side instead. The results indicate that the accuracy increases when increasing the amount of data only on the side being projected. More data on the corresponding side (i.e. the side being projected to) decreases the accuracy. The same behavior is observed whether the projected side is Arabic (upper half) or English (lower half). All in all, the accuracy values are low. The accuracy increases about three times when looking at the 5-best instead of the 1-best accuracy. While the accuracies 32.2% and 33.1% are low, they reflect that the word representations are encoding some information about the words, although this information might not be good enough to build a word-to-word lexicon. However, using this information for OOV reduction might still yield improvements as we will see in the translation results.

	Arabic	English
word corpus size	231M	229M
phrase corpus size	126M	115M
word corpus vocab. size	467K	421K
phrase corpus vocab. size	5.8M	5.3M
# phrase vectors	934K	913K

Table 3: Phrase vectors statistics.

5.3 Phrase Vectors

Translation experiments pertaining to the proposed semantic feature are presented here. The feature is based on phrase vectors which are built with the word2vec toolkit in a similar way word vectors are trained, except that the training corpus is the phrase corpus containing phrases constructed as described in section 3. Once trained, a new feature is added to the phrase table. The feature is computed for each phrase pair using phrase vectors as described in Eq. 3.

Table 3 shows statistics about the phrase corpus and the original word corpus it is based on. Algorithm 1 is used to build the phrase corpus using 5 passes. The number of phrase vectors trained using the phrase corpus are also shown. Note that the tool used does not produce vectors for all 5.8M Arabic and 5.3M English phrases in the vocabulary. Rather, noisy phrases are excluded from training, eventually leading to 934K Arabic and 913K English phrase embeddings.

We perform two experiments on the IWSLT 2013 Arabic-to-English evaluation data set. In the first experiment, we examine how the semantic feature affects a small phrase table (2.3M phrase pairs) trained on the in-domain data (WIT). The second experiment deals with a larger phrase table (34M phrase pairs), constructed by a linear interpolation between in- and out-of-domain phrase tables including UN data, resulting in a competitive baseline. The two baselines have hierarchical reordering models (HRMs) and a tuned mixture LM, in addition to the standard models, as described in section 5.1. The results are shown in table 4.

In the small experiment, the semantic phrase feature improves TER by 0.7%, and BLEU by 0.4% on the test set eval13. The translation seems to benefit from the contextual information encoded in the phrase vectors during training. This is in contrast to the training of the standard phrase

Arabic Data	English Data	1-best Match %	5-best Matches %
WIT+UN	WIT+UN	8.0	26.1
WIT+UN+arGiga3	WIT+UN	**10.9**	**32.2**
WIT+UN	WIT+UN+news	4.9	17.9
WIT+UN+arGiga3	WIT+UN+news	7.5	25.7
WIT+UN	WIT+UN	8.4	27.2
WIT+UN	WIT+UN+news	**10.9**	**33.1**
WIT+UN+arGiga3	WIT+UN	5.7	18.9
WIT+UN+arGiga3	WIT+UN+news	8.3	25.2

Table 2: The effect of increasing the amount of data on the quality of word vectors. VSM-based scores are compared to IBM model 1 $p(e|f)$ (upper half) and $p(f|e)$ (lower half), effectively regarding the IBM 1 models as the true probability distributions. In the upper part, the projection is done on the representation of the Arabic word f, and the similarity is computed between the projection and the representation of the English word e. In the lower half of the table, the role of f and e is interchanged, where the English side in this case will be projected.

system	**dev2010**		**eval2013**	
	BLEU	TER	BLEU	TER
WIT	29.1	50.5	28.9	52.5
+ feature	29.1	‡**50.1**	‡29.3	‡**51.8**
+ paraph.	29.2	‡50.2	‡**29.5**	‡**51.8**
+ both	29.2	50.2	‡29.4	‡**51.8**
WIT+UN	29.7	49.3	30.5	50.5
+ feature	29.8	49.2	30.2	50.7

Table 4: Semantic feature and paraphrasing results. The symbol ‡ indicates statistical significance with $p < 0.01$.

features, which disregards context. As for the hierarchical reordering models which are part of the baseline, they do not capture lexical information about the context. They are only limited to the ordering information. The skip-gram-based phrase vectors used for the semantic feature, on the other hand, discard ordering information, but uses contextual lexical information for phrase representation. In this sense, HRMs and the semantic feature can be said to complement each other. Using the semantic feature for the large phrase table did not yield improvements. The difference compared to the baseline in this case is not statistically significant.

All reported results are averages of 3 MERT optimizer runs. Statistical significance is computed using the Approximate Randomization (AR) test. We used the multeval toolkit (Clark et al., 2011) for evaluation.

5.4 Paraphrasing and OOV Reduction

The next set of experiments investigates the reduction of the OOV rate through paraphrasing, and its impact on translation. Paraphrasing is performed employing the cosine similarity, and the k-NN search is done on the source side, with $k = 3$. The nearest neighbors are required to satisfy a radius threshold $r > 0.3$, i.e., neighbors with a similarity value less or equal to r are rejected. Training the projection matrices is performed using a small amount of training data amounting to less than $30k$ translation pairs.

To examine the effect of OOV reduction, we perform paraphrasing on a resource-limited system, where a small amount of parallel data exists, but a larger amount of monolingual data is available. Such a system is simulated by training word vectors on the WIT+UN data monolingually , while extracting the phrase table using the much smaller in-domain WIT data set only. Table 5 shows the change in the number of OOV words after introducing the paraphrased rules to the WIT-based phrase table. 19% and 30% of the original OOVs are eliminated in the dev and eval13 sets, respectively. This reduction translates to an improvement of 0.6% BLEU and 0.7% TER as indicated in table 4.

Since BLEU or TER are based on word identities and do not detect semantic similarities, we make a comparison between the reference translations and translations of the system that employed

phrase table	# OOV dev	eval13
WIT	185	254
WIT+paraph.	150	183
Vocab. size	3,714	4,734

Table 5: OOV change due to paraphrasing. Vocabulary refers to the number of unique tokens in the Arabic dev and test sets.

OOV	VSM-based Translation	Reference
تكشفت	found	unfolded
حريصة	interested	keen
سجني	jail	imprisoned
بلاغ	claim	report
ملتبسة	confusing	confounding
حثت	encourage	rallied for
قرويا	villagers	redneck

Table 6: Examples of OOV words that were translated due to paraphrasing. The examples are extracted from the translation hypotheses of the small experiment.

OOV reduction. Examples are shown in Table 6. Although the reference words are not matched exactly, the VSM translations are semantically close to them, suggesting that OOV reduction in these cases was somewhat successful, although not rewarded by either of the scoring measures used.

6 Related Work

Bilingually-constrained phrase embeddings were developed in (Zhang et al., 2014). Initial embeddings were trained in an unsupervised manner, followed by fine-tuning using bilingual knowledge to minimize the semantic distance between translation equivalents, and maximizing the distance between non-translation pairs. The embeddings are learned using recursive neural networks by decomposing phrases to their constituents. While our work includes bilingual constraints to learn phrase vectors, the constraints are implicit in the phrase corpus. Our approach is simple, focusing on the preprocessing step of preparing the phrase corpus, and therefore it can be used with different existing frameworks that were developed for word vectors.

Zou et al. (2013) learn bilingual word embeddings by designing an objective function that combines unsupervised training with bilingual constraints based on word alignments. Similar to our work, they compute an additional feature for phrase pairs using cosine similarity. Word vectors are averaged to obtain phrase representations. In contrast, our approach learns phrase representations directly.

Recurrent neural networks were used with minimum translation units (Hu et al., 2014), which are phrase pairs undergoing certain constraints. At the input layer, each of the source and target phrases are modeled as a bag of words, while the output phrase is predicted word-by-word assuming conditional independence. The approach seeks to alleviate data sparsity problems that would arise if phrases were to be uniquely distinguished. Our approach does not break phrases down to words, but learns phrase embeddings directly.

Chen et al. (2010) represent a rule in the hierarchical phrase table using a bag-of-words approach. Instead, we learn phrase vectors directly without resorting to their constituent words. Moreover, they apply a count-based approach and employ IBM model 1 probabilities to project the target space to the source space. In contrast, our mapping is similar to that of Mikolov et al. (2013a) and is learned directly from a small set of bilingual data.

Mikolov et al. (2013a) proposed an efficient method to learn word vectors through feed-forward neural networks by eliminating the hidden layer. They do not report end-to-end sentence translation results as we do in this work.

Mikolov et al. (2013b) learn direct representations of phrases after joining a training corpus using a simple monolingual point-wise mutual information criterion with discounting. Our work exploits the rich bilingual knowledge provided by the phrase table to join the corpus instead.

Gao et al. (2013) learn shared space mappings using a feed-forward neural network and represent a phrase vector as a bag-of-words vector. The vectors are learned aiming to optimize an expected BLEU criterion. Our work is different in that we learn two separate source and target mappings.

We also do not follow their bag-of-words phrase model approach.

Marton et al. (2009) proposed to eliminate OOVs by looking for similar words using distributional vectors, but they prune the search space limiting it to candidates observed in the same context as that of the OOV. We do not employ such a heuristic. Instead, we perform a k-nearest neighbor search spanning the full phrase table to paraphrase its rules and generate new entries.

Estimating phrase table scores using monolingual data was investigated in (Klementiev et al., 2012), by building co-occurrence context vectors and using a small dictionary to induce new scores for existing phrase rules. Our work explores the use of distributional vectors extracted from neural networks, moreover, we induce new phrase rules to extend the phrase table. New phrase rules were also generated in (Irvine and Callison-Burch, 2014), where new phrases were produced as a composition of unigram translations.

7 Conclusion

In this work we adapted vector space models to provide the state-of-the-art phrase-based statistical machine translation system with semantic information. We leveraged the bilingual knowledge of the phrase table to construct source and target phrase corpora to learn phrase vectors, which were used to provide semantic scoring of phrase pairs. Word vectors allowed to extend the phrase table and eliminate OOVs. Both methods proved beneficial for low-resource tasks.

Future work would investigate decoder integration of semantic scoring that extends beyond phrase boundaries to provide semantically coherent translations.

Acknowledgments

This material is partially based upon work supported by the DARPA BOLT project under Contract No. HR0011- 12-C-0015. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 287658.

References

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Boxing Chen, George Foster, and Roland Kuhn. 2010. Bilingual sense similarity for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 834–843.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *49th Annual Meeting of the Association for Computational Linguistics:shortpapers*, pages 176–181, Portland, Oregon, June.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, June.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2013. Learning semantic representations for the phrase translation model. *arXiv preprint arXiv:1312.0482*.

Zellig S Harris. 1954. Distributional structure. *Word*.

Yuening Hu, Michael Auli, Qin Gao, and Jianfeng Gao. 2014. Minimum translation modeling with recurrent neural networks. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 20–29, Gothenburg, Sweden, April. Association for Computational Linguistics.

Ann Irvine and Chris Callison-Burch. 2014. Hallucinating phrase translations for low resource mt. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.

Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. 2012. Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 130–140. Association for Computational Linguistics.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

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

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 381–390. Association for Computational Linguistics.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

R.C. Moore and W. Lewis. 2010. Intelligent Selection of Language Model Training Data. In *ACL (Short Papers)*, pages 220–224, Uppsala, Sweden, July.

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech & Language*, 21(3):492–518.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of the Int. Conf. on Speech and Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO, September.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation Modeling with Bidirectional Recurrent Neural Networks. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, October.

Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 262–270, Uppsala, Sweden, July.

Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012. Jane 2: Open source phrase-based and hierarchical statistical machine translation. In *International Conference on Computational Linguistics*, pages 483–491, Mumbai, India, December.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.