Learning Semantic Representations for the Phrase Translation Model

Jianfeng Gao
Microsoft Research, Redmond
Washington 98052, USA
jfgao@microsoft.com

Xiaodong He
Microsoft Research, Redmond
Washington 98052, USA
xiaohe@microsoft.com

Wen-tau Yih
Microsoft Research, Redmond
Washington 98052, USA
scottyih@microsoft.com

Li Deng
Microsoft Bing, Bellevue
Washington 98004, USA
deng@microsoft.com

Abstract

This paper presents a novel semantic-based phrase translation model. A pair of source and target phrases are projected into continuous-valued vector representations in a low-dimensional latent semantic space, where their translation score is computed by the distance between the pair in this new space. The projection is performed by a multi-layer neural network whose weights are learned on parallel training data. The learning is aimed to directly optimize the quality of end-to-end machine translation results. Experimental evaluation has been performed on two Europarl translation tasks, English-French and German-English. The results show that the new semantic-based phrase translation model significantly improves the performance of a state-of-the-art phrase-based statistical machine translation system, leading to a gain of 0.7-1.0 BLEU points.

1 Introduction

The phrase translation model, also known as the *phrase table*, is one of the core components of phrase-based statistical machine translation (SMT) systems. The most common method of constructing the phrase table takes a two-phase approach (e.g., Koehn et al. 2003). First, the bilingual phrase pairs are extracted heuristically from an automatically word-aligned training data. The second phase, which is the focus of this paper, is parameter estimation where each phrase pair is assigned with some scores that are estimated based on counting these phrases or their words using the same word-aligned training data.

Phrase-based SMT systems have achieved state-of-the-art performance largely due to the fact that long phrases, rather than single words, are used as translation units so that useful context information can be captured in selecting translations. However, longer phrases occur less often in training data, leading to a severe data sparseness problem in parameter estimation. There has been a plethora of research reported in the literature on improving parameter estimation for the phrase translation model (e.g., DeNero et al. 2006; Wuebker et al. 2010; He and Deng 2012; Gao and He 2013).

This paper revisits the problem of scoring a phrase translation pair by developing a novel, Semantic-based Phrase Translation Model (SPTM). The translation score of a phrase pair in this model is computed as follows. First, we represent each phrase as a bag-of-words vector, called *word vector* henceforth. We then project the word vector, in either the source language or the target language, into a respective continuous feature vector in a common low-dimensional latent semantic space that is intended to be language independent. The projection is performed by a multi-layer neural network. The projected feature vector forms the *semantic representation* of a phrase. Finally, the translation score of a source-target phrase pair is computed by the distance between their feature vectors.

The main motivation behind the SPTM is to alleviate the data sparseness problem associated with the traditional counting-based methods by grouping phrases with a similar meaning across different languages. In this model, semantically related phrases, in both the source and the target languages, would have similar (close) feature vectors in the semantic space. Since the translation score is a smooth function of these feature vectors, a small change in semantics (e.g., the phrases that differ only in morphological forms) should only lead to a small change in the translation score.

The primary research task in developing the SPTM is learning the semantic representation of a phrase that is effective for SMT. Motivated by recent studies on continuous-space language models (e.g., Bengio et al. 2003; Mikovlov et al. 2011), we use a neural network to project a word vector to a feature vector. Ideally, the projection would discover those latent semantic features that are useful to differentiate *good* translations from *bad* ones, for a given source phrase. However, there is no training data with explicit annotation on the quality of phrase translations. The phrase translation pairs are *hidden* in the parallel source-target sentence pairs, which are used to train the traditional translation models. The quality of a phrase translation can only be judged implicitly through the translation quality of the sentences, as measured by BLEU, which contain the phrase pair. In order to overcome this challenge and let the BLEU metric guide the projection learning, we propose a new method to learn the parameters of a neural network. This new method automatically forces the feature vector of a source phrase to be closer to the feature vectors of its candidate translations that lead to a better BLEU score, if these translations are selected by an SMT decoder to produce final, sentence-level translations. The new learning method makes use of the L-BFGS algorithm and the expected BLEU as the objective function defined on N-best lists.

To the best of our knowledge, the SPTM proposed in this paper is the first continuous-space phrase translation model that is shown to lead to significant improvement over a standard phrase-based SMT system (to be detailed in Section 6). Like the traditional phrase translation model, the translation score of each bilingual phrase pair is modeled *explicitly* in our model. However, instead of estimating the phrase translation score on aligned parallel data, our model intends to capture the semantic similarity between a source phrase and its paired target phrase by projecting them into a common, latent semantic space that is language independent.

The rest of the paper is organized as follows. Section 2 reviews previous work that lays the foundation of this study. Section 3 reviews the log-linear model for phrase-based SMT and Sections 4 presents the SPTM. Section 5 describes the way the model parameters are estimated, followed by the experimental results in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

Originally designed for information retrieval (IR), Latent Semantic Analysis (LSA) (Deerwester et al. 1990) is arguably the earliest continuous semantic model. Using a document collection, LSA first forms a document-term matrix and then finds its low-rank approximation using singular value decomposition. Whether two words or two documents are semantically related can be determined by the distance of their projected vectors in the concept or semantic space. Unlike LSA, where the concept vectors do not have a clear probabilistic interpretation, generative topic models, such as Probabilistic LSA (Hofmann 1999) and Latent Dirichlet Allocation (LDA) (Blei et al. 2003), represent a document as a multinomial distribution over a small set of topics (called the document-topic distribution). Each of the topics is in turn a multinomial distribution over words (called the topic-word distribution). In contrast, recent work on continuous space language models, e.g., the feed-forward neural network language model (NNLM) (Bengio et al. 2003) and the recurrent neural network language model (RNNLM) (Mikolov et al. 2010; 2011; Auli et al. 2013), provide a different kind of latent semantic representation. While these models have shown to significantly outperform the traditional *n*-gram language model in speech and natural language tasks (Mikolov et al. 2011; Collobert et al. 2011), the by-product of these models, the low-dimensional real-valued vector of each word, also captures both syntactic and semantic regularities in languages (Mikolov et al. 2013; Zhila et al. 2013).

Because these latent semantic models are developed for mono-lingual settings, word embedding from these models is not directly applicable to translation. As a result, variants of such models for cross-lingual scenarios have been proposed. For instance, projection methods like Cross-lingual LSI (Dumais et al. 1997), oriented PCA (OPCA) (Platt et al. 2010) and canonical correlation analysis (Vinokourov et al. 2002) have been applied to cross-lingual IR, where documents in different languages are projected into the shared latent concept space. These models can be further improved by using a Siamese neural network approach, S2Net (Yih et al. 2011), and by using deep structured models, DSSM (Huang et al. 2013). In addition, generative counterparts like Polylingual topic models (Mimno et al. 2009) and Bi-Lingual Topic Model (BLTM) (Gao et al. 2011) have also been proposed. In principle, a phrase translation table can be derived using any of these cross-lingual models, although decoupling the derivation from the SMT training often results in suboptimal performance.

Despite the success of latent semantic representations in various applications, there is, however, much less work on continuous-space translation models. The only exception we are aware of is the work of continuous space *n*-gram translation models (Schwenk et al. 2007; Son et al. 2012), where the feed-forward neural network language model is extended to represent translation probabilities. However, these earlier studies focused on a particular type of models, the so-called *n*-gram translation models, where the translation probability of a phrase or a sentence is decomposed as a product of word (or bilingual translation unit) *n*-gram probabilities of the same form as that in a standard *n*-gram language model. Therefore, it is not clear how their approaches can be applied to the phrase translation model, which is much more widely used in modern SMT systems.

In contrast, our model learns jointly the representations of a phrase in the source language as well as its translation in the target language. Our approach is inspired by the research on learning continuous-space vector representations of words (Mikolov et al. 2013), phrases (Socher et al. 2012) and scenes (Socher et al. 2011), and is analogous to the latent semantic model proposed by Weston et al. (2011) that simultaneously learns semantic representations of an image and its word label. The SPTM proposed in this paper bears a strong resemblance to the latent semantic models for IR, where queries and documents are matched using their learned semantic representations in order to tackle the lexical mismatch problem: a concept is often expressed using different words in documents and queries. If we view a source-target phrase pair as a query-document pair, these models can be readily applied to modeling phrase translations.

There has been much recent research on improving the phrase table using more principled methods for phrase extraction (e.g., Lamber and Banchs 2005), parameter estimation (e.g., Wuebker et al. 2010; He and Deng 2012; Gao and He 2013), or both (e.g., Marcu and Wong 2002; Denero et al. 2006). A recent survey is due to Koehn (2010). Among them, Gao and He (2013) is most relevant to the work described in this paper. They estimate phrase translation probabilities using a discriminative training method under the N-best reranking framework of SMT. In particular, they demonstrate via a comparative study that using an N-best list based expected BLEU as the objective function is crucial to obtaining good results. Expected BLEU and its variants have also been explored in earlier studies (e.g., Rosti et al. 2011; He and Deng 2012). In this study we use the same objective function to learn the semantic representations of phrases, integrating the strengths associated with both of these earlier studies.

In the next three sections, we will describe the SPTM, starting with a brief review of the reranking framework for SMT.

3 The Log-Linear Model for SMT

Phrase-based SMT is based on a log-linear model which requires learning a mapping between inputs $F \in \mathcal{F}$ to outputs $E \in \mathcal{E}$. We are given

- Training samples (F_i, E_i) for $i = 1 \dots N$, where each source sentence F_i is paired with a reference translation in target language E_i;
- A procedure GEN to generate a list of N-best candidates $\mathrm{GEN}(F_i)$ for an input F_i, where GEN in this study is the baseline phrase-based SMT system, i.e., a reimplementation of the Moses system (Koehn et al. 2007) that does not use the SPTM, and each $E \in \mathrm{GEN}(F_i)$ is labeled by the sentence-level BLEU score (He and Deng 2012), denoted by $\mathrm{sBleu}(E_i, E)$, which meaures the quality of E with respect to its reference translation E_i;
- A vector of features $\mathbf{h} \in \mathbb{R}^M$ that maps each (F_i, E) to a vector of feature values; and
- A parameter vector $\boldsymbol{\lambda} \in \mathbb{R}^M$, which assigns a real-valued weight to each feature.

SMT involves hidden-variable models such that a hidden variable A is assumed to be constructed during the process of generating E. In the phrase-based SMT, A consists of a segmentation of the source and target sentences into phrases and an alignment between source and target phrases.

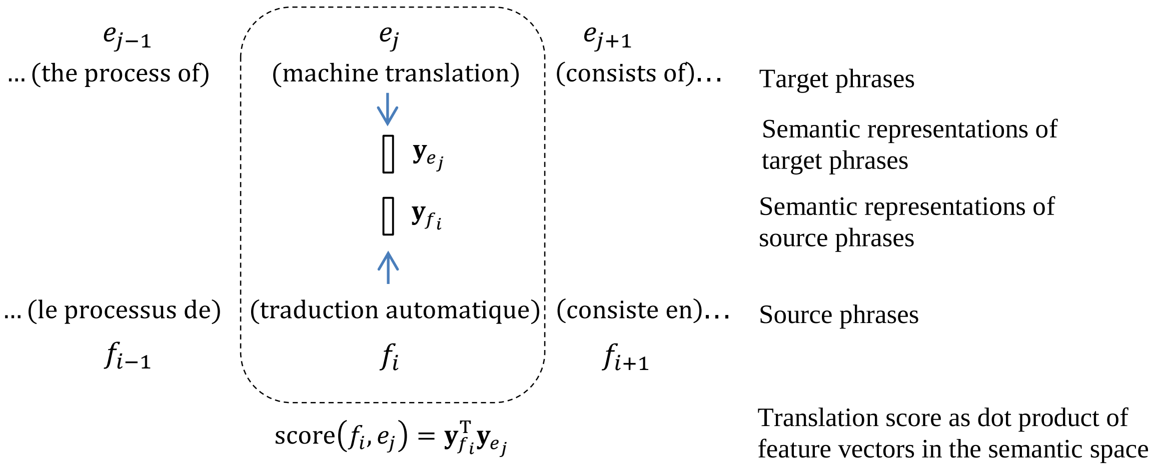
e_{j-1}
... (the process of)
e_j
(machine translation)
e_{j+1}
(consists of)...
Target phrases
$\mathbf{y}_{e_j}$
Semantic representations of target phrases
$\mathbf{y}_{f_i}$
Semantic representations of source phrases
... (le processus de)
(traduction automatique)
(consiste en)...
Source phrases
f_{i-1}
f_i
f_{i+1}
$\text{score}(f_i, e_j) = \mathbf{y}_{f_i}^{\mathrm{T}} \mathbf{y}_{e_j}$
Translation score as dot product of feature vectors in the semantic space

Figure 1. The architecture of the SPTM, where the mapping from a phrase to its semantic representation is shown in Figure 2.

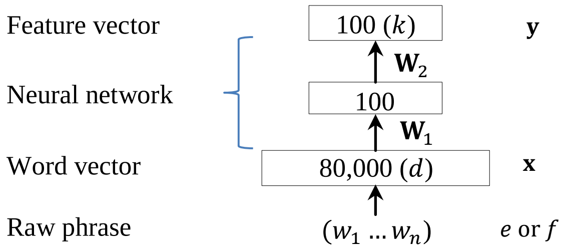

Figure 2. A neural network model for phrases giving rise to their semantic representations. The model with the same form is used for both source and target languages.

The components $\text{GEN}(.)$, $\mathbf{h}$ and $\boldsymbol{\lambda}$ define a log-linear model that maps F_i to an output as follows:

$$E^* = \underset{(E,A) \in \text{GEN}(F_i)}{\text{argmax}} \; \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{h}(F_i, E, A) \qquad (1)$$

which states that given $\boldsymbol{\lambda}$ and $\mathbf{h}$, argmax returns the highest scoring translation E^*, maximizing over correspondences A. In phrase translation models, computing the argmax exactly is intractable, so it is performed approximatedly by beam search (Och and Ney 2004). Following Liang et al. (2006), we assume that every translation candidate is always coupled with a corresponding A, called *Viterbi derivation*, generated by (1).

4 A Semantic-Based Phrase Translation Model (SPTM)

The architecture of the SPTM is shown in Figures 1 and 2, where for each pair of source and target phrases (f_i, e_j) in a source-target sentence pair, we first project them into feature vectors $\mathbf{y}_{f_i}$ and $\mathbf{y}_{e_j}$ in a latent semantic space via a neural network with one hidden layer (as shown in Figure 2), and then compute the translation score, $\text{score}(f_i, e_j)$, by the distance of their feature vectors in that space.

We start with a bag-of-words representation of a phrase $\mathbf{x} \in \mathbb{R}^d$, where $\mathbf{x}$ is a word vector and d is the size of the vocabulary consisting of words in both source and target languages. We then learn to project $\mathbf{x}$ to a low-dimensional semantic space $\mathbb{R}^k$:

$$\phi(\mathbf{x}): \mathbb{R}^d \to \mathbb{R}^k$$

The projection is performed using a fully connected neural network with one hidden layer and tanh activation functions. Let $\mathbf{W}_1$ be the projection matrix from the input layer to the hidden layer and $\mathbf{W}_2$ the projection matrix from the hidden layer to the output layer, we have

$$\mathbf{y} \equiv \phi(\mathbf{x}) = \tanh\left(\mathbf{W}_2{}^{\mathrm{T}}\left(\tanh\left(\mathbf{W}_1{}^{\mathrm{T}}\mathbf{x}\right)\right)\right) \qquad (2)$$

The translation score of a source phrase f and a target phrase e can be measured as the similarity (or distance) between their feature vectors. We choose the dot product as the similarity function[1]:

$$\text{score}(f,e) \equiv \text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e) = \mathbf{y}_f^{\mathrm{T}}\mathbf{y}_e \qquad (3)$$

According to (2), we see that the value of the scoring function is determined by the projection matrices $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2\}$.

The SPTM of (2) and (3) can be incoporated into the log-linear model for SMT (1) by introducing a new feature h_{M+1} and a new feature weight λ_{M+1}. The new feature is defined as

$$h_{M+1}(F_i,E,A) = \sum_{(f,e)\in A} \text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e) \qquad (4)$$

Thus, the phrase-based SMT system, into which the SPTM is incorporated, is parameterized by $(\boldsymbol{\lambda},\boldsymbol{\theta})$, where $\boldsymbol{\lambda}$ is a vector of a handful of parameters used in the log-linear model of (1), with one weight for each feature; and $\boldsymbol{\theta}$ is the projection matrices used in the SPTM defined by (2) and (3). In our experiments we take three steps to learn $(\boldsymbol{\lambda},\boldsymbol{\theta})$:

1. Given a baseline phrase-based SMT system and a pre-set $\boldsymbol{\lambda}$ where $\lambda_{M+1} = 0$, we generate for each source sentence in training data an N-best list of translation hypotheses.
2. We fix $\boldsymbol{\lambda}$ and set $\lambda_{M+1} = 1$, and optimize $\boldsymbol{\theta}$ w.r.t. a loss function on training data.
3. We fix $\boldsymbol{\theta}$, and optimize $\boldsymbol{\lambda}$ using MERT (Och 2003) to maximize the BLEU score on development data.

In the next section, we will describe Step 2 in detail because it is directly related to the SPTM training.

5 Training SPTM

This section describes the kind of loss function we employ with the SPTM and the algorithm to train the neural network weights using the loss function as the optimization objective.

We define the loss function $\mathcal{L}(\boldsymbol{\theta})$ as the negative of the N-best list based expected BLEU, denoted by $\text{xBleu}(\boldsymbol{\theta})$. In the reranking framework of SMT outlined in Section 3, $\text{xBleu}(\boldsymbol{\theta})$ over one training sample (F_i, E_i) is defined as

$$\text{xBleu}(\boldsymbol{\theta}) = \sum_{E\in\text{GEN}(F_i)} P(E|F_i)\text{sBleu}(E_i,E) \qquad (5)$$

where $\text{sBleu}(E_i,E)$ is the sentence-level BLEU score, and $P(E|F_i)$ is a normalized translation probability from F_i to E computed using *softmax* as

$$P(E|F_i) = \frac{\exp\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i,E,A)\right)}{\sum_{E\in\text{GEN}(F_i)}\exp\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i,E,A)\right)} \qquad (6)$$

where $\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}$ is the log-linear model of (1), which also includes the feature derived from the SPTM as defined by (4).

Let $\mathcal{L}(\boldsymbol{\theta})$ be a loss function which is differentiable w.r.t. the parameters of the SPTM, $\boldsymbol{\theta}$. We can compute the gradient of the loss and learn $\boldsymbol{\theta}$ using gradient-based numerical optimization algorithms, such as L-BFGS (Nocedal and Wright 2006) or stochastic gradient descent (SGD) (Bottou 2004).

5.1 Computing the Gradient

Since the loss does not explicitly depend on $\boldsymbol{\theta}$, we use the chain rule for differentiation:

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} &= \sum_{(f,e)} \frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)}\frac{\partial\text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)}{\partial\boldsymbol{\theta}} \\ &= \sum_{(f,e)} -\delta_{(f,e)}\frac{\partial\text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)}{\partial\boldsymbol{\theta}} \qquad (7)\end{aligned}$$

which takes the form of summation over all phrase pairs occurring either in a training sample (stochastic mode) or in the entire training data (batch mode). $\delta_{(f,e)}$ in (7) is known as the *error* term of the phrase pair (f,e), and is defined as

$$\delta_{(f,e)} = -\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)} \qquad (8)$$

It describes how the overall loss changes with the translation score of the phrase pair (f,e). We will leave the derivation of $\delta_{(f,e)}$ to Section 5.1.2, and will first describe how the gradient of $\text{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)$ w.r.t. $\boldsymbol{\theta}$ is computed.

[1] In our experiments, we compared dot product and the cosine similarity function and found that the former works better for nonlinear multi-layer neural networks, and the latter works better for linear neural networks. For the sake of clarity, we choose dot product when we describe the SPTM and its training in Sections 4 and 5, respectively.

5.1.1 Computing $\partial \mathbf{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)/\partial \boldsymbol{\theta}$

Without loss of generality, we use the following notations to describe a neural network:

- $\mathbf{W}_l$ is the projection matrix for the *l*-th layer of the neural network;
- $\mathbf{x}$ is the input word vector of a phrase;
- $\mathbf{z}^l$ is the sum vector of the *l*-th layer; and
- $\mathbf{y}^l = \sigma(\mathbf{z}^l)$ is the output vector of the *l*-th layer, where σ is an activation function;

Thus, the SPTM defined by (2) and (3) can be represented as

$$\mathbf{z}^1 = \mathbf{W}_1{}^{\mathrm{T}}\mathbf{x}$$
$$\mathbf{y}^1 = \sigma(\mathbf{z}^1)$$
$$\mathbf{z}^2 = \mathbf{W}_2{}^{\mathrm{T}}\mathbf{y}^1$$
$$\mathbf{y}^2 = \sigma(\mathbf{z}^2)$$
$$\mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e) = \left(\mathbf{y}_f^2\right)^{\mathrm{T}}\mathbf{y}_e^2$$

The gradient of the matrix $\mathbf{W}_2$ which projects the hidden vector to the output vector is computed as:

$$\frac{\partial \mathrm{sim}_{\boldsymbol{\theta}}\left(\mathbf{x}_f, \mathbf{x}_e\right)}{\partial \mathbf{W}_2} = \frac{\partial \left(\mathbf{y}_f^2\right)^{\mathrm{T}}}{\partial \mathbf{W}_2}\mathbf{y}_e^2 + \left(\mathbf{y}_f^2\right)^{\mathrm{T}}\frac{\partial \mathbf{y}_e^2}{\partial \mathbf{W}_2}$$
$$= \mathbf{y}_f^1\left(\mathbf{y}_e^2 \circ \sigma'\left(\mathbf{z}_f^2\right)\right)^{\mathrm{T}} + \mathbf{y}_e^1\left(\mathbf{y}_f^2 \circ \sigma'\left(\mathbf{z}_e^2\right)\right)^{\mathrm{T}} \quad (9)$$

where $\circ$ is the element-wise multiplication (Hadamard product). Applying the back propagation principle, the gradient of the projection matrix mapping the input vector to the hidden vector $\mathbf{W}_1$ is computed as

$$\frac{\partial \mathrm{sim}_{\boldsymbol{\theta}}\left(\mathbf{x}_f, \mathbf{x}_e\right)}{\partial \mathbf{W}_1}$$
$$= \mathbf{x}_f\left(\mathbf{W}_2\left(\mathbf{y}_e^2 \circ \sigma'\left(\mathbf{z}_f^2\right)\right) \circ \sigma'\left(\mathbf{z}_f^1\right)\right)^{\mathrm{T}}$$
$$+\mathbf{x}_e\left(\mathbf{W}_2\left(\mathbf{y}_f^2 \circ \sigma'(\mathbf{z}_e^2)\right) \circ \sigma'(\mathbf{z}_e^1)\right)^{\mathrm{T}} \quad (10)$$

The derivation can be easily extended to a neural network with multiple hidden layers.

5.1.2 Computing $\boldsymbol{\delta}_{(f,e)}$

To simplify the notation, we rewrite our loss function of (5) and (6) over one training sample as

$$\mathcal{L}(\boldsymbol{\theta}) = -\mathrm{xBleu}(\boldsymbol{\theta}) = -\frac{\mathrm{G}(\boldsymbol{\theta})}{\mathrm{Z}(\boldsymbol{\theta})} \quad (11)$$

where

$$\mathrm{G}(\boldsymbol{\theta}) = \sum\nolimits_E \mathrm{sBleu}(E, E_i)\exp\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i, E, A)\right)$$
$$\mathrm{Z}(\boldsymbol{\theta}) = \sum\nolimits_E \exp\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i, E, A)\right)$$

Combining (8) and (11), we have

$$\delta_{(f,e)} = \frac{\partial \mathrm{xBleu}(\boldsymbol{\theta})}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)} \quad (12)$$
$$= \frac{1}{\mathrm{Z}(\boldsymbol{\theta})}\left(\frac{\partial \mathrm{G}(\boldsymbol{\theta})}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)} - \frac{\partial \mathrm{Z}(\boldsymbol{\theta})}{\partial \mathrm{sim}_{\boldsymbol{\theta}}\left(\mathbf{x}_f, \mathbf{x}_e\right)}\mathrm{xBleu}(\boldsymbol{\theta})\right)$$

Because $\boldsymbol{\theta}$ is only relevant to h_{M+1} which is defined in (4), we have

$$\frac{\partial \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i, E, A)}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)} = \lambda_{M+1}\frac{\partial h_{M+1}(F_i, E, A)}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)}$$
$$= \lambda_{M+1}N(f, e; A) \quad (13)$$

where $N(f, e; A)$ is the number of times the phrase pair (f, e) occur in A. Combining (12) and (13), we end up with the following equation

$$\delta_{(f,e)}$$
$$= \sum_{(E,A)\in GEN(F_i)} \mathrm{U}(\boldsymbol{\theta}, E)P(E|F_i)\lambda_{M+1}N(f, e; A)$$

where (14)

$\mathrm{U}(\boldsymbol{\theta}, E) = \mathrm{sBleu}(E_i, E) - \mathrm{xBleu}(\boldsymbol{\theta})$.

5.2 The Training Algorithm

In our experiments we train the parameters of the SPTM $\boldsymbol{\theta}$ using the L-BFGS optimizer described in Andrew and Gao (2007), together with the loss function described in (5). The gradient is computed as described in Sections 5.1. Even though the loss function is not convex, we found that the L-BFGS iterations over the complete training data (batch mode) minimizes the loss in practice in a desirable fashion; e.g., convergence of the algorithm was found to be smooth.

6 Experiments

We conducted our experiments on two Europarl translation tasks, English-to-French (EN-FR) and

Systems	EN-FR (TEST2)	DE-EN (TEST2)
Rank-1 system	31.92	**27.30**
Rank-2 system	31.79	25.97
Rank-3 system	31.75	25.54
Our baseline	**32.84**	26.04

Table 1: Baseline results in BLEU. The results of top ranked systems are reported in Koehn and Monz (2006).

#	Systems	EN-FR		DE-EN	
		TEST1	TEST2	TEST1	TEST2
1	**Baseline**	32.79	32.84	26.04	26.04
2	**SPTM**	**33.79**$^{\alpha}$	**33.81**$^{\alpha}$	**26.82**$^{\alpha}$	**26.72**$^{\alpha}$
3	**SPTM$_{\text{L}}$**	33.56$^{\alpha\beta}$	33.51$^{\alpha\beta}$	26.67$^{\alpha}$	26.50$^{\alpha\beta}$
4	**SPTM$_{\text{W}}$**	33.21$^{\alpha\beta}$	33.27$^{\alpha\beta}$	26.56$^{\alpha\beta}$	26.49$^{\alpha\beta}$
5	**SPTM$_{\text{L-W}}$**	33.25$^{\alpha\beta}$	33.35$^{\alpha\beta}$	26.46$^{\alpha\beta}$	26.33$^{\alpha\beta}$
6	**2 + 4**	33.79$^{\alpha}$	33.81$^{\alpha}$	26.81$^{\alpha}$	26.73$^{\alpha}$

Table 2: Main results (BLEU scores) of semantic-based phrase translation models. The superscripts α and β indicate statistically significant difference ($p < 0.05$) from **Baseline** and **SPTM**, respectively.

German-to-English (DE-EN). The data sets are published for the shared task in NAACL 2006 Workshop on Statistical Machine Translation (WMT06) (Koehn and Monz 2006).

For EN-FR, the training set contains 688K sentence pairs, with 21 words per sentence on average. The development set contains 2000 sentences. We used 2000 sentences from the WMT05 shared task as the first test set (TEST1), and the 2000 sentences from the WMT06 shared task as the second test set (TEST2). For DE-EN, the training set contains 751K sentence pairs, with 21 words per sentence on average. The official development set used for the shared task contains 2000 sentences. We used 2000 sentences from the WMT05 shared task as TEST1, and the 2000 sentences from the WMT06 shared task as TEST2.

Two baseline phrase-based SMT systems, each for one language pair, are developed as follows. These baseline systems are used in our experiments both for comparison purpose and for generating N-best lists for training the SPTM. First, we performed word alignment on the training set using a hidden Markov model with lexicalized distortion (He 2007), and then extracted the phrase table from the word aligned bilingual texts (Koehn et al. 2003). The maximum phrase length is set to four. Other models used in a baseline system include a lexicalized reordering model, word count and phrase count, and a trigram language model trained on the English training data provided by the WMT06 shared task. A fast beam-search phrase-based decoder (Moore and Quirk 2007) is used and the distortion limit is set to four. The decoder is modified so as to output the Viterbi derivation for each translation hypothesis.

The metric used for evaluation is case insensitive BLEU score (Papineni et al. 2002). We also performed a significance test using the paired *t*-test. Differences are considered statistically significant when the *p*-value is less than 0.05. Table 1 presents the baseline results. The performance of our phrase-based SMT systems compares favorably to the top-ranked systems, thus providing a fair baseline for our research.

6.1 Results

Table 2 shows the main results measured in BLEU evaluated on TEST1 and TEST2, where Row 1 is the baseline system. Rows 2 to 5 are the systems enhanced by integrating different versions of the SPTM.

SPTM in Row 2 is the model described in Sections 4. As illustrated in Figure 2, the number of the nodes in the input layer is the vocabulary size d. Both the hidden layer and the output layer have 100 nodes. That is, $\mathbf{W}^1$ is a $d \times 100$ matrix and $\mathbf{W}^2$ a 100×100 matrix. Table 2 shows that **SPTM** leads to a substantial improvement over the baseline system across all test sets, with a statistically significant margin from 0.7 to 1.0 BLEU points.

We have developed a set of variants of **SPTM**, as shown in Rows 3 to 5, to investigate two design choices we made in developing the SPTM: (1) whether to use a linear projection or a multi-layer nonlinear projection; and (2) whether to compute the phrase similarity using word-word similarities as suggested by e.g., the lexical weighting model (Koehn et al. 2003).

SPTM$_{\text{L}}$ (Row 3) uses a linear neural network to project a word vector of a phrase $\mathbf{x}$ to a feature vector $\mathbf{y}$

$$\mathbf{y} \equiv \phi(\mathbf{x}) = \mathbf{W}^{\mathrm{T}}\mathbf{x}$$

where $\mathbf{W}$ is a $d \times 100$ projection matrix. The translation score of a source phrase f and a target phrase

e is measured as the similarity of their feature vectors. We choose cosine similarity because it works better than dot product for linear projection.

$$\text{score}(f, e) \equiv \text{sim}_{\mathbf{W}}(\mathbf{x}_f, \mathbf{x}_e) = \frac{\mathbf{y}_f^{\mathrm{T}}\mathbf{y}_e}{\|\mathbf{y}_f\|\|\mathbf{y}_e\|}$$

SPTM$_{\mathbf{W}}$ (Row 4) computes the phrase similarity using word-word similarity scores. This follows the common smoothing strategy of addressing the data sparseness problem in modeling phrase translations, such as the lexical weighting model (Koehn et al. 2003) and the word factored n-gram translation model (Son et al. 2012). Let w denote a word, and f and e the source and target phrases, respectively. We define

$$\begin{aligned}\text{sim}(f, e) = \frac{1}{|f|}\sum\nolimits_{w \in f} \text{sim}_\tau(w, e) \\ + \frac{1}{|e|}\sum\nolimits_{w \in e} \text{sim}_\tau(w, f)\end{aligned}$$

where $\text{sim}_\tau(w, e)$ (or $\text{sim}_\tau(w, f)$) is the word-phrase similarity, and is defined as a smooth approximation of the maximum function

$$\begin{aligned}&\text{sim}_\tau(w, e) \\ &= \frac{\sum_{w' \in e} \text{sim}(w, w') \exp\big(\tau \text{sim}(w, w')\big)}{\sum_{w' \in e} \exp\big(\tau \text{sim}(w, w')\big)}\end{aligned}$$

where τ is the temperature parameter optimized on development data. sim_τ has the following properites:

1. $\text{sim}_\tau \rightarrow \max$ as $\tau \rightarrow \infty$
2. sim_0 is the average of its inputs
3. $\text{sim}_\tau \rightarrow \min$ as $\tau \rightarrow -\infty$

Similar to **SPTM**, **SPTM**$_{\mathbf{W}}$ also uses a nonlinear projection to map each word (not a word vector of a phrase as in **SPTM**) to a feature vector. **SPTM**$_{\mathbf{L\text{-}W}}$ (Row 5) computes the phrase similarity using word-word similarity scores, but the projection is performed using a linear model, similar to **SPTM**$_{\mathbf{L}}$.

Two observations can be made by comparing **SPTM** in Row 2 to its variants in Rows 3-5. First of all, it is more effective to model the phrase translation directly than decomposing it into word-word translations in the SPTMs (Row 2 vs. Row 4 and Row 3 vs. Row 5). Moreover, unlike the case of traditional phrase translation models, combining the phrase model and the word model does not lead to any visible improvement (Row 6 vs. Row 2), indicating that with semantic representations, a phrase model is no longer sparser than a word model. Second, we see that in phrase models (Rows 2 and 3) the nonlinear projection is able to capture more sophisticated semantic information and leads to better results than the linear projection.

#	Systems	EN-FR		DE-EN	
		TEST1	**TEST2**	**TEST1**	**TEST2**
1	**Baseline**	32.79	32.84	26.04	26.04
2	**SPTM**	**33.79**$^{\alpha}$	**33.81**$^{\alpha}$	**26.82**$^{\alpha}$	**26.72**$^{\alpha}$
3	**BLTM**$_{\mathbf{PR}}$	32.78$^{\beta}$	32.95	26.06$^{\beta}$	26.09$^{\beta}$
4	**DPM**	32.90$^{\beta}$	32.99$^{\alpha\beta}$	26.20$^{\alpha\beta}$	26.16$^{\beta}$

Table 3: Comparing **SPTM** to two latent semantic models. The superscripts α and β indicate statistically significant difference (p < 0.05) from **Baseline** and **SPTM**, respectively.

6.2 Comparing with Previous Latent Semantic Models

This section compares the best version of the SPTM i.e., **SPTM** in Row 2 of Table 2, with two state-of-the-art latent semantic models that are originally trained on clicked query-document pairs (i.e., click-through data extracted from search logs) for query-document matching (Gao et al. 2011). To adopt these models for SMT, we view source-target sentence pairs as clicked query-document pairs, and trained both models using the same methods as in Gao et al. (2011) on the parallel bilingual training data described earlier.

The results are shown in Table 3. **BTLM**$_{\mathbf{PR}}$ (Row 3) is an extension to PLSA, and is the best performer among different versions of the Bi-Lingual Topic Model (BLTM) described in Gao et al. (2011). BLTM with Posterior Regularization (**BLTM**$_{\mathbf{PR}}$) is trained on parallel training data using the EM algorithm with a constraint enforcing a source sentence and its paralleled target sentence to not only share the same prior topic distribution, but to also have similar fractions of words assigned to each topic (Ganchev et al. 2010). We incorporated the model into the log-linear model for SMT (1) as follows. First of all, the topic distribution (i.e., semantic representation) of a source sentence F_i, denoted by $P(z|F_i)$, is induced from the learned topic-word distributions using EM. Then, each translation candidate E in the N-best list $\text{GEN}(F_i)$ is scored as

$$P(E|F_i) = \prod_{w \in E} \sum_{z} P(w|z) P(z|F_i)$$

$P(F_i|E)$ can be similarly computed. Finally, the logarithms of the two probabilities are incorporated into the log-linear model of (1) as two additional features.

DPM (Row 4) is the Discriminative Projection Model described in Gao et al. (2011). **DPM** uses a matrix $\mathbf{W}$ to project a word vector of a sentence to a feature vector. $\mathbf{W}$ is trained on parallel training data using the S2Net algorithm (Yih et al. 2011) as follows. For each source sentence in training data, we treat it and its paralleled translation in target language as a positive pair, and we randomly selected 4 other target sentences from training data to form 4 negative pairs. $\mathbf{W}$ is trained in such a way that a positive source-target sentence pair has a higher similarity (i.e., cosine similarity) than that of the negative ones of the same source sentence. **DPM** can be incorporated into the log-linear model for SMT (1) by introducing a new feature h_{M+1}. Let $\mathbf{x}$ be the word vector of a source sentence F_i (or its translation candidate E), and $\mathbf{y}$ be the projected feature vector, i.e., $\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x}$. The new feature is defined as

$$h_{M+1}(F_i, E) \equiv \mathrm{sim}_{\mathbf{W}}(\mathbf{x}_{F_i}, \mathbf{x}_E) = \frac{\mathbf{y}_{F_i}^{\mathrm{T}}\mathbf{y}_E}{\|\mathbf{y}_{F_i}\|\|\mathbf{y}_E\|}$$

Similar to that BLTM is an extension to PLSA, DPM can be viewed as an extension of LSA where bilingual parallel data can be explored for translation model training. As we see from Table 3, both latent semantic models, although leading to some slight improvement over **Baseline**, are much less effective than **SPTM** which is based on a multi-layer neural network trained on the N-best lists using a loss function that tailors to the BLEU metric. However, we found in our experiments that these models can be useful for “pre-training” to provide a good initial model that not only speeds up the SPTM training but also leads to a better final model.

6.3 Discussion

Although SGD has been advocated for neural network training due to its simplicity and its robustness to local minimum (Bengio 2009), we found that in our task the L-BFGS based batch training performs well despite the non-convexity in our loss. Another merit of batch training is that the gradient over all training data can be computed efficiently. As shown in Section 5, computing $\partial \mathrm{sim}_{\theta}(\mathrm{x}_f, \mathrm{x}_e)/\partial\theta$ requires large-scale matrix multiplications, and is expensive for multi-layer neural networks. Eq. (7) suggests that $\partial \mathrm{sim}_{\theta}(\mathrm{x}_f, \mathrm{x}_e)/\partial\theta$ and $\delta_{(f,e)}$ can be computed separately, thus making the computation cost of the former term only depends on the number of phrase pairs in the phrase table, but not the size of training data. Therefore, the training method described in Section 5 can be used on larger amounts of training data with little difficulty.

7 Conclusions

The work presented in this paper makes two important contributions. First, we develop a novel phrase translation model for SMT, where the translation score of a pair of source-target phrases is represented as the distance between their feature vectors in a low-dimensional, continuous-valued semantic space. The semantic space is derived from the representations generated using a multi-layer neural network. Second, we present a new learning method to train the weights in the multi-layer neural network for the end-to-end BLEU metric directly. The training method is based on L-BFGS. We describe in detail how the gradient in closed form, as required for efficient optimization, is derived. The objective function, which takes the form of the expected BLEU computed from N-best lists, is very different from the usual objective functions used in most existing neural networks, e.g., cross entropy (Hinton et al. 2012) or mean square error (Deng et al. 2012). We hence have provided details in the derivation of the gradient, which can serve as an example to guide the derivation of neural network learning with other non-standard objective functions in the future.

Our evaluation on two Europal translation tasks show that incorporating the SPTM into the log-linear framework of SMT significantly improves the performance of a state-of-the-art phrase-based SMT system, leading to a gain between 0.7 to 1.0 BLEU points. Careful implementation of the L-BFGS optimization based on the BLEU-centric objective function, together with the associated closed-form gradient, is a key to the success.

A natural extension of this work is to expand the model and learning algorithm from shallow to deep neural networks. The deep models are expected to produce more powerful and flexible semantic representations, and thus greater performance gain than what is presented in this paper.

References

Andrew, G. and Gao, J. 2007. Scalable training of L1-regularized log-linear models. In *ICML*.

Auli, M., Galley, M., Quirk, C. and Zweig, G. 2013 Joint language and translation modeling with recurrent neural networks. In *EMNLP*.

Bengio, Y. 2009. Learning deep architectures for AI. *Fundamental Trends Machine Learning*, vol. 2, no. 1, pp. 1–127.

Bengio, Y., Duharme, R., Vincent, P., and Janvin, C. 2003. A neural probabilistic language model. *JMLR*, 3:1137-1155.

Blei, D. M., Ng, A. Y., and Jordan, M. J. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research,* 3: 993-1022.

L. Bottou. 2004. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning, Lecture Notes in Artificial Intelligence*, LNAI 3176, pages 146–168. Springer Verlag, Berlin.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, vol. 12.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6): 391-407

DeNero, J., Gillick, D., Zhang, J., and Klein, D. 2006. Why generative phrase models underperform surface heuristics. In *Workshop on Statistical Machine Translation*, pp. 31-38.

Deng, L., Yu, D., and Platt, J. 2012. Scalable stacking and learning for building deep architectures. In *ICASSP*.

Diamantaras, K. I., and Kung, S. Y. 1996. *Principle Component Neural Networks: Theory and Applications.* Wiley-Interscience.

Dumais S., Letsche T., Littman M. and Landauer T. 1997. Automatic cross-language retrieval using latent semantic indexing. In AAAI-97 Spring Symposium Series: Cross-Language Text and Speech Retrieval.

Ganchev, K., Graca, J., Gillenwater, J., and Taskar, B. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research,* 11 (2010): 2001-2049.

Gao, J., and He, X. 2013. Training MRF-based translation models using gradient ascent. In *NAACL-HLT*, pp. 450-459.

Gao, J., Toutanova, K., Yih., W-T. 2011. Clickthrough-based latent semantic models for web search. In *SIGIR*, pp. 675-684.

He, X. 2007. Using word-dependent transition models in HMM based word alignment for statistical machine translation. In *Proc. of the Second ACL Workshop on Statistical Machine Translation*.

He, X., and Deng, L. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *ACL*, pp. 292-301.

Hinton, G., and Salakhutdinov, R., 2010. Discovering Binary Codes for Documents by Learning Deep Generative Models. *Topics in Cognitive Science*, pp. 1-18.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR*, pp. 50-57.

Huang, P-S., He, X., Gao, J., Deng, L., Acero, A. and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Koehn, P. 2010. *Statistical machine translation*. Cambridge University Press.

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., and Herbst, E. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007,* demonstration session.

Koehn, P. and Monz, C. 2006. Manual and automatic evaluation of machine translation between European languages. In *Workshop on Statistical Machine Translation*, pp. 102-121.

Koehn, P., Och, F., and Marcu, D. 2003. Statistical phrase-based translation. In *HLT-NAACL*, pp. 127-133.

Lambert, P. and Banchs, R. E. 2005. Data inferred multi-word expressions for statistical machine translation. In *MT Summit X*, Phuket, Thailand.

Liang, P., Bouchard-Cote, A. Klein, D., and Taskar, B. 2006. An end-to-end discriminative approach to machine translation. In *COLING-ACL*.

Marcu, D., and Wong, W. 2002. A phrase-based, joint probability model for statistical machine translation. In *EMNLP*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space. In *CoRR*, abs/1301.3781.

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., and Khudanpur, S. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pp. 1045-1048.

Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., and Khudanpur, S. 2011. Extensions of recurrent neural network language model. In *ICASSP*, pp. 5528-5531.

Mikolov, T., Yih, W. and Zweig, G. 2013. Linguistic Regularities in Continuous Space Word Representations. In *NAACL-HLT*.

Mimno, D., Wallach, H., Naradowsky, J., Smith, D. and McCallum, A. 2009. Polylingual topic models. In *EMNLP*.

Moore, R., and Quirk, C. 2007. Faster beam-search decoding for phrasal statistical machine translation. In *MT Summit XI*.

Nocedal, J. and Wright, S. 2006. *Numerical Optimization*. Springer, 2nd edition.

Och, F. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pp. 160-167.

Och, F., and Ney, H. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 29(1): 19-51.

Papineni, K., Roukos, S., Ward, T., and Zhu W-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Platt, J., Toutanova, K., and Yih, W. 2010. Translingual Document Representations from Discriminative Projections. In *EMNLP*.

Rosti, A-V., Hang, B., Matsoukas, S., and Schwartz, R. S. 2011. Expected BLEU training for graphs: bbn system description for WMT system combination task. In *Workshop on Statistical Machine Translation*.

Schwenk, H., Costa-Jussa, M. R. and Fonollosa, J. A. R. 2007. Smooth bilingual n-gram translation. In *EMNLP-CoNLL*, pp. 430-438.

Socher, R., Huval, B., Manning, C., Ng, A., 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *EMNLP*.

Socher, R., Lin, C., Ng, A. Y., and Manning, C. D. 2011. Parsing natural scenes and natural language with recursive neural networks. In *ICML*.

Son, L. H., Allauzen, A., and Yvon, F. 2012. Continuous space translation models with neural networks. In *NAACL-HLT*, pp. 29-48.

Vinokourov, A., Shawe-Taylor, J. and Cristianini, N. 2002. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*.

Weston, J., Bengio, S., and Usunier, N. 2011. Large scale image annotation: learning to rank with joint word-image embeddings. In *IJCAI*.

Wuebker, J., Mauser, A., and Ney, H. 2010. Training phrase translation models with leaving-one-out. In *ACL*, pp. 475-484.

Yih, W., Toutanova, K., Platt, J., and Meek, C. 2011. Learning discriminative projections for text similarity measures. In *CoNLL*.

Zhila, A., Yih, W., Meek, C., Zweig, G. and Mikolov, T. 2013. Combining Heterogeneous Models for Measuring Relational Similarity. In *NAACL-HLT*.