Learning Semantic Representations for Nonterminals in Hierarchical Phrase-Based Translation

Xing Wang, Deyi Xiong[*] and **Min Zhang**
Soochow University, Suzhou, China 215006
xingwsuda@gmail.com, {dyxiong, minzhang}@suda.edu.cn

Abstract

In hierarchical phrase-based translation, coarse-grained nonterminal Xs may generate inappropriate translations due to the lack of sufficient information for phrasal substitution. In this paper we propose a framework to refine nonterminals in hierarchical translation rules with real-valued semantic representations. The semantic representations are learned via a weighted mean value and a minimum distance method using phrase vector representations obtained from large scale monolingual corpus. Based on the learned semantic vectors, we build a semantic nonterminal refinement model to measure semantic similarities between phrasal substitutions and nonterminal Xs in translation rules. Experiment results on Chinese-English translation show that the proposed model significantly improves translation quality on NIST test sets.

1 Introduction

Hierarchical phrase-based translation (Chiang, 2007) explores formal synchronous context free grammar (SCFG) rules for translation. Two types of nonterminal symbols are used in translation rules: nonterminal X in ordinary SCFG rules and nonterminal S in glue rules that are specially introduced to concatenate nonterminal Xs in a monotonic manner. The same generic symbol X for all ordinary nonterminals makes it difficult to distinguish and select proper translation rules.

In order to address this issue, researchers either use syntactic labels to annotate nonterminal Xs (Zollmann and Venugopal, 2006; Zollmann and Vogel, 2011; Li et al., 2012; Hanneman and Lavie, 2013), or employ syntactic information from parse trees to refine nonterminals with real-valued vectors (Venugopal et al., 2009; Huang et al., 2013). In addition to syntactic knowledge, semantic structures are also leveraged to refine nonterminals (Gao and Vogel, 2011). All these efforts focus on incorporating linguistic knowledge into hierarchical translation rules.

Unfortunately, syntactic or semantic parsers for many languages are not accessible due to the lack of labeled training data. In contrast, a large amount of unlabeled data are easily available. Therefore, can we mine syntactic or semantic properties for nonterminals from unlabeled data? Or can we exploit these data to refine nonterminals for SMT?

Learning semantic representations for terminals (words, multi-word phrases or sentences) from unlabeled data has achieved substantial progress in recent years (Mitchell and Lapata, 2008; Turian et al., 2010; Socher et al., 2010; Mikolov et al., 2013c; Blunsom et al., 2014). These representations have been used successfully in various NLP tasks. However, there is no attempt to learn semantic representations for nonterminals from unlabeled data. In this paper we propose a framework to learn semantic representations for nonterminal Xs in translation rules. Our framework is established on the basis of real-valued vector representations learned for multi-word phrases, which are substituted with nonterminal Xs during hierarchical rule extraction. We propose a weighted mean value and a minimum distance method to obtain nonterminal representations from representations of their phrasal substitutions. We further build a semantic nonterminal refinement model with semantic representations of nonterminals to compute similarities between phrasal substitutions and nonterminals. In doing so, we want to enhance phrasal substitution and translation rule selection during decoding.

The big challenge here is that thousands of tar-

[*] Corresponding author

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1391–1400,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

get phrasal substitutions will be generated for one single nonterminal during decoding. Computing vector representations for all these phrases will be very time-consuming. We therefore introduce two different methods to handle it. In the first method, we project representations of source phrases onto their target counterparts linearly/nonlinearly via a neural network. These projected vectors are used as approximations to real target representations to compute semantic similarities. In the second method, we decode sentences in two passes. The first pass collects target phrase candidates from n-best translations of sentences generated by the baseline. The second pass calculates vector representations of these collected target phrases and then computes similarities between them and target-side nonterminals.

Our contributions are two-fold. First, we learn semantic representations for nonterminals from their phrasal substitutions with two different methods. This is the first time, to the best of our knowledge, to induce semantic representations for nonterminals from unlabeled data in the context of SMT. Second, we successfully address the issue of time-consuming target-side phrase-nonterminal similarity computation mentioned above. We incorporate both source-/target-side semantic nonterminal refinement model and their combination based on learned nonterminal representations into translation system. Experiment results show that our method can achieve an improvement of 1.16 BLEU points over the baseline system on NIST MT evaluation test sets.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 presents our approach of learning semantic vectors for nonterminals, followed by Section 4 describing the details of our semantic nonterminal refinement model. Section 5 introduces the integration of the proposed model into SMT. Experiment results are reported in Section 6. Finally, we conclude our work in Section 7.

2 Related Work

A variety of approaches have been explored for nonterminal refinement in hierarchical phrase-based translation. These approaches can be categorized into two groups: 1) augmenting the nonterminal symbol X with informative labels, and 2) attaching distributional linguistic knowledge to each nonterminal in hierarchical rules. The former only allows substitution operations with matched labels. The latter normally builds an additional model as a new feature of the log-linear model to incorporate attached knowledge.

Among approaches which directly refine the single label to more fine-grained labels, syntactic and semantic knowledge are explored in various ways. The syntactically augmented translation model (SAMT) proposed by Zollmann and Venugopal (2006) uses syntactic categories extracted from target-side parse trees to augment nonterminals in hierarchical rules. Unfortunately, there is a data sparseness problem in this model due to thousands of extracted syntactic categories. One solution to address this issue is to reduce the number of syntactic categories. Zollmann and Vogel (2011) use word tags, generated by either POS tagger or unsupervised word class induction, instead of syntactic categories. Hanneman and Lavie (2013) coarsen the label set by introducing a label collapsing algorithm to SAMT grammars (Zollmann and Venugopal, 2006). Yet another solution is easing restrictions on label matching. Shen et al. (2009) penalize substitution with unmatched labels while Chiang (2010) uses soft match features to model substitutions with various labels. Similar to Zollmann and Venugopal (2006), Hoang and Koehn (2010) decorate some hierarchical rules with source-side syntax information and use undecorated, decorated, and partially decorated rules in their translation model. Mylonakis and Sima'an (2011) employ source-side syntax-based labels to define a joint probability synchronous grammar. Combinatory Categorial Grammar (CCG) labels or CCG contextual labels are also used to enrich nonterminals (Almaghout et al., 2011; Weese et al., 2012). Li et al. (2012) incorporate head information extracted from source-side dependency structures into translation rules. Besides, semantic knowledge is also used to refine nonterminals. Gao and Vogel (2011) utilize target-side semantic roles to form SRL-aware SCFG rules. Most of approaches introduced here explicitly require syntactic or semantic parsers trained on manually labeled data.

On the other hand, efforts have also been directed towards attaching distributional linguistic knowledge to nonterminals. Venugopal et al. (2009) propose a preference grammar to annotate nonterminals based on preference distributions of syntactic categories. Huang et al. (2010) learn latent syntactic distributions for each nonterminal. They use these distributions to decorate nonterminal Xs in SCFG rules with a real-valued feature vectors and utilize these vectors to measure the similarities between source phrases and applied rules. Similar to this work, Huang et al. (2013) utilize treebank tags based on dependency parsing to learn latent distributions. Cao et al. (2014) attach translation rules with dependency knowledge, which contains both dependency relations inside rules and dependency relations between rules and their contexts.

The difference of our work from these studies is that our semantic representations are learned from unlabeled bilingual (or monolingual) data and do not depend on any linguistic resources, e.g., parsers. We also believe that our model is able to exploit both syntactic and semantic information for nonterminals since vector representations learned in our way are able to capture both syntactic and semantic properties (Turian et al., 2010; Socher et al., 2010).

3 Learning Semantic Representations for Nonterminals

In our framework, semantic representations for nonterminal Xs are automatically induced from word-aligned parallel corpus. In this section, we detail the essential component of our approach, i.e., how to learn semantic vectors for nonterminals and how to project source semantic vectors onto target language semantic space. Before discussing nonterminal representations, we briefly introduce vector representations for words and phrases.

3.1 Prerequisite: Learning Words and Phrases Representations

We employ a neural method, specifically the continuous bag-of-words model (Mikolov et al., 2013a) to learn high-quality vector representations for words. Once we complete the training of the continuous bag-of-words model, word embeddings form an embedding matrix $M \in \mathbb{R}^{d\times|V|}$, where d is a pre-determined embedding dimensionality and each word w in the vocabulary V corresponds to a vector $\vec{v} \in \mathbb{R}^d$. Given the embedding matrix M, mapping words to vectors can be done by simply looking up their respective columns in M.

We further feed these learned word embeddings to recursive autoencoders (RAE) (Socher et al., 2011) for learning phrase representations. In traditional RAE (shown in Figure 1), given two input children representation vectors $\vec{c_1} \in \mathbb{R}^d$ and $\vec{c_2} \in \mathbb{R}^d$, their parent representation $\vec{p}$ can be calculated as follows:

$$\vec{p} = f^{(1)}(W^{(1)}[\vec{c_1}; \vec{c_2}] + b^{(1)}) \qquad (1)$$

where $[\vec{c_1}; \vec{c_2}] \in \mathbb{R}^{2d}$ is the concatenation of vectors of two children, $W^{(1)} \in \mathbb{R}^{d\times 2d}$ is a weight matrix, $b^{(1)} \in \mathbb{R}^d$ is a bias term, and $f^{(1)}$ is an element-wise activation function such as $tanh$. The above output representation $\vec{p}$ can be used as a child vector to construct the representation for a larger subphrase. This process is repeated until a binary tree covering the whole input phrase is generated.

In order to evaluate how well the parent vector represents its children, we can reconstruct the children in a reconstruction layer:

$$[\vec{c_1}'; \vec{c_2}'] = f^{(2)}(W^{(2)}\vec{p} + b^{(2)}) \qquad (2)$$

where $\vec{c_1}'$ and $\vec{c_2}'$ are the reconstructed children, $W^{(2)}$ is a weight matrix for reconstruction, $b^{(2)}$ is a bias term for reconstruction, and $f^{(2)}$ is an element-wise activation function.

For each node in the generated binary tree, we compute Euclidean distance between the original input vectors and the reconstructed vectors to measure the reconstruction error:

$$E_{rec}([\vec{c_1}; \vec{c_2}]) = \frac{1}{2}\|[\vec{c_1}; \vec{c_2}] - [\vec{c_1}'; \vec{c_2}']\|^2 \qquad (3)$$

By minimizing the total reconstruction error over all nonterminal nodes, we can learn parameters of RAE.

Socher et al. (2011) propose a greedy unsupervised RAE as an extension to the above traditional RAE. The main difference is that in the unsupervised RAE there is no tree structure which is given for traditional RAE. It can learn both representations and tree structures of phrases or sentences. In this work, we adopt the unsupervised RAE to learn vector representations for phrases.

3.2 Inducing Nonterminal Representations from Phrase Representations

As we extract hierarchical rules from phrases by replacing subphrases with nonterminal symbols, a nonterminal X is generalized from a number of

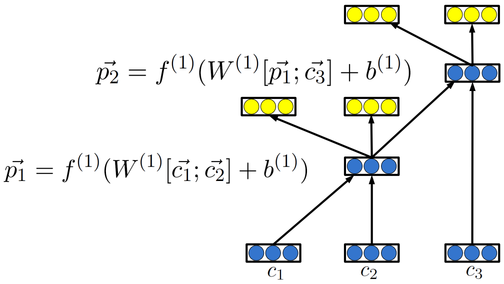
$\vec{p_2} = f^{(1)}(W^{(1)}[\vec{p_1}; \vec{c_3}] + b^{(1)})$
$\vec{p_1} = f^{(1)}(W^{(1)}[\vec{c_1}; \vec{c_2}] + b^{(1)})$
c_1
c_2
c_3

Figure 1: The architecture of a recursive autoencoder, adapted from (Socher et al., 2011). Blue nodes are original vectors and yellow nodes are reconstructed vectors which are used to compute reconstruction errors.

subphrases. We believe that these subphrases determine syntactic and semantic properties of the nonterminal X. We therefore enrich each nonterminal X with a semantic vector induced from vector representations of phrases that are replaced by the nonterminal during rule extraction.

For an SCFG rule, we can learn semantic vectors for nonterminals on both the source and target side. Due to the space limitation, we introduce the procedure of learning nonterminal vectors on the source side. Semantic vectors on the target side can be learned analogically.

For each source-side nonterminal X of a hierarchical rule, we collect all source subphrases replaced by X in a source subphrase set $P = \{p_1, p_2, \cdots, p_m\}$. We also count the number of times of these phrases being replaced by nonterminal X on training data during rule extraction. We collect these numbers in a count set $C = \{c_1, c_2, \cdots, c_m\}$. Based on the phrase set P, count set C and learned phrase vector representations in P, we can compute a semantic vector $\vec{v_x}$ for nonterminal X in each SCFG rule.

We propose two general approaches to obtain semantic vectors for nonterminals: a weighted mean value method and a minimum distance method. Given phrase vector representations $\vec{P_r} = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_m}\}$, we calculate the semantic vector for a nonterminal generalized from these phrases as follows.

Weighted mean value method (MV) computes semantic vector $\vec{v_x}$ as:

$$\vec{v_x} = \frac{\sum_{i=1}^{m} c_i \cdot \vec{p_i}}{\sum_{i=1}^{m} c_i} \qquad (4)$$

Minimum distance method (MD) finds a point in semantic space to minimize the sum of Euclidean distances of vectors in $\vec{P_r}$ to this point. Formally,

$$\vec{v_x} = \operatorname*{argmin}_{\vec{v_x}} \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{d} (p_{ij} - v_{xj})^2} \qquad (5)$$

We use the stochastic gradient descent algorithm to find the minimal distance and the point $\vec{v_x}$. The component v_{xj} can be updated by $v_{xj} \leftarrow v_{xj} + \lambda \frac{\partial f}{\partial v_{xj}}$ where f is $\sum_{i=1}^{m} \sqrt{\sum_{j=1}^{d} (p_{ij} - v_{xj})^2}$ and λ is the learning rate.

Similar to the center of gravity, the semantic vector $\vec{v_x}$ learned by this method acts as a semantic centroid for all vectors of phrases that are substituted by X. Nonterminals in different hierarchical translation rules will have different semantic centroids. These centroids will help translation model capture semantic diversity to a certain degree.

3.3 Mapping Source-Side Representations onto Target-Side Semantic Space

As we discussed in Section 1, directly learning vector representations for target phrases is very costly in practice. Inspired by Mikolov et al. (2013b), we adopt vector projection to alleviate this problem. Different from mapping representations from the source side to the target side by learning a linear matrix on word alignments (Mikolov et al., 2013b), we project source multi-word phrase representations onto the target semantic space in a nonlinear manner as we believe that nonlinear relations between languages are more reasonable. Specifically, we use a neural network to achieve this goal. Our neural network is a multilayer feed-forward neural network with one hidden layer. The functional form can be written in the following equation:

$$\vec{p} = \tanh(W^{(4)}(\tanh(W^{(3)}\vec{src}) + b^{(3)}) + b^{(4)}) \qquad (6)$$

where $\vec{src}$ is the input vector which is learned in the source semantic space, $W^{(3)}$ denotes the weight matrix for connections between input and hidden neurons and $W^{(4)}$ denotes the weight matrix for links between hidden neurons and output, $b^{(3)}$ and $b^{(4)}$ are bias terms. To train the neural network, we optimize the following objective:

$$J = \operatorname*{argmin}_{W^{(3)}, W^{(4)}} \frac{1}{N} \sum_{i=1}^{N} \|\vec{trg_i} - \vec{p_i}\|^2 + R(\theta) \qquad (7)$$

where N is the number of training examples, $\vec{trg_i}$ is the target vector representation for the ith example learned by RAE and $\vec{p_i}$ is the output of the neural network for the source vector representation $\vec{src_i}$ of ith example. $R(\theta)$ is the regularizer on parameters:

$$R(\theta) = \frac{\lambda_L}{2}\|W\|^2 \tag{8}$$

where W denotes parameters for parameter matrices $W^{(3)}$, $W^{(4)}$ and bias terms $b^{(3)}$, $b^{(4)}$.

4 Semantic Nonterminal Refinement Model

In this section, we describe our semantic nonterminal refinement model on the basis of induced real-valued semantic vectors for nonterminals.

4.1 Nonterminal Representations in Hierarchical Rules

We incorporate learned semantic representations of nonterminals into hierarchical rules. In particular, ordinary hierarchical rules take the following form:

$$X \rightarrow \langle aX_sb, cX_td \rangle \tag{9}$$

where *a*/*b*, *c*/*d* are strings of terminals on the source and target side, s and t are placeholders denoting the nonterminal X on the source or target side, X_s and X_t are aligned to each other.

Representations for nonterminals can be on either the source or target side. They are attached to hierarchical rules as follows:

$$X \rightarrow \langle aX_sb, cX_td, \vec{v_{xs}}, \vec{v_{xt}} \rangle \tag{10}$$

where $\vec{v_{x\cdot}}$ is the source- or target-side semantic representation for nonterminal. In this way, we keep original translation rules intact and decorate nonterminals with their semantic representations.

4.2 The Model

The proposed semantic nonterminal refinement model estimates the semantic similarity between a phrase p and nonterminal X. The phrase p and nonterminal X will have a high similarity score in the representation space if they are semantically similar. The higher semantic similarity scores are, the more compatible nonterminals are with corresponding phrases.

There is another nonterminal S in glue rules, which are formalized as follows:

$$S \rightarrow \langle S_1X_2, S_1X_2 \rangle \tag{11}$$

$$S \rightarrow \langle X_1, X_1 \rangle \tag{12}$$

This nonterminal S is different from X. We therefore treat it as a special case in the computation of semantic similarity.

In this work, we explore two approaches to compute similarity: one based on cosine similarity and the other based on Euclidean distance.

Given a phrase vector representation $\vec{p}$ and nonterminal X semantic vector $\vec{v_x}$, **Cosine Similarity** (CS) is computed as:

$$cos(\vec{p}, \vec{v_x}) = \frac{\vec{p} \cdot \vec{v_x}}{\|\vec{p}\|\|\vec{v_x}\|} \tag{13}$$

We set α for the Cosine Similarity between the glue rule and its corresponding phrase as follows:

$$SeSim = \begin{cases} cos(\vec{p}, \vec{v_x}) & hierarchical\, rules \\ \alpha & glue\, rules \end{cases} \tag{14}$$

As for **Euclidean Distance** (ED), it is computed according to the following formula:

$$dist(\vec{p}, \vec{v_x}) = \sqrt{\sum_{i=1}^{d}(p_i - v_{xi})^2} \tag{15}$$

and similarly we set β for glue rules:

$$SeSim = \begin{cases} dist(\vec{p}, \vec{v_x}) & hierarchical\, rules \\ \beta & glue\, rules \end{cases} \tag{16}$$

5 Decoding

We incorporate the proposed model as a new feature into the hierarchical phrase-based translation system. Specifically, two features are added into the baseline system:

1. Source-side semantic similarity between source phrases and nonterminals

2. Target-side semantic similarity between target phrases and nonterminals

We compute source- and target-side similarities based on representations of nonterminals and phrasal substitutions for each applied rule, and sum up these similarities to calculate the total score of a derivation on the two features.

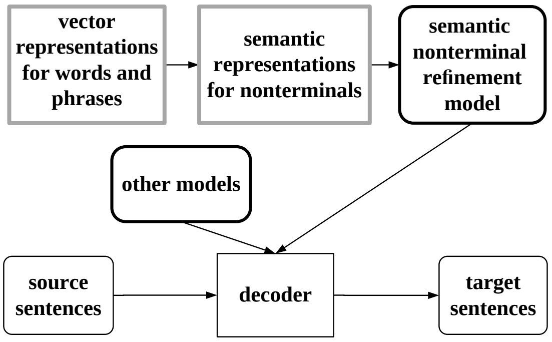
vector representations for words and phrases
semantic representations for nonterminals
semantic nonterminal refinement model
other models
source sentences
decoder
target sentences

Figure 2: Architecture of SMT system with the proposed semantic nonterminal refinement model.

The integration of the source-side semantic nonterminal refinement model into the decoder is trivial. For the target-side model, however, we have to consider the efficiency issue as we mentioned in Section 1. We introduce two different methods to integrate the target-side model into the decoder: 1) projection and 2) two-pass decoding. In the first integration method, a mapping neural network is trained to map source phrase representations onto the target semantic space as described in Section 3.3. The projection can be linear if we remove the hidden layer in the projection neural network. This is similar to the mapping matrix learned by Mikolov et al. (2013b). We calculate semantic similarities between projected representations of phrases and those of nonterminals. In the two-pass decoding, we collect target phrase candidates from 100-best translations for each source sentence generated by the baseline in the first pass and learn vector representations for these target phrase candidates. Then in the second pass, we decode source sentence with our target semantic nonterminal refinement model using learned target phrase vector representations. If a target phrase appears in the collected set, the target-side semantic nonterminal refinement model will calculate the semantic similarity between the target phrase and the corresponding nonterminal on the target semantic space; otherwise the model will give a penalty. This is because this phrase is not a desirable phrase as it is not used in 100-best translations.

The weights of these two features are tuned by the Minimum Error Rate Training (MERT)(Och, 2003), together with weights of other sub-models on a development set. Figure 2 shows the architecture of SMT system with the proposed semantic nonterminal refinement model.

6 Experiment

In this section, we conducted a series of experiments on Chinese-to-English translation using large-scale bilingual training data, aiming at the following questions:

1. Which approach is better for learning nonterminal representations, weighted mean value or minimum distance?

2. Can the target-side semantic nonterminal refinement model improve translation quality? And which method is better for integrating the target-side semantic model into translation, projection or two-pass decoding?

3. Does the combination of source and target semantic nonterminal refinement models provide further improvement?

6.1 Setup

Our training corpus contains 2.9M sentence pairs with 80.9M Chinese words and 86.4M English words from LDC data[1]. We used NIST MT03 as our development set, NIST MT06 as our development test set and MT08 as our final test set.

We ran Giza++ on the training corpus in both Chinese-to-English and English-to-Chinese directions and applied the “grow-diag-final” refinement rule (Koehn et al., 2003) to obtain word alignments. We used the SRI Language Modeling Toolkit[2] (Stolcke and others, 2002) to train our language models. MERT (Och, 2003) was adopted to tune feature weights of the decoder. We used the case-insensitive BLEU[3] as our evaluation metric. In order to alleviate the instability of MERT , we followed Clark et al. (2011) to perform three runs of MERT and reported average BLEU scores over the three runs for all our experiments.

We used word2vec toolkit[4] to train our word embeddings and set the vector dimension d to 30. In our training experiment, we used the continuous bag-of-words model with a context window of size 5. The monolingual corpus, which was used to pre-train word embeddings, is extracted from

[1] The corpora include LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08 (Hong Kong Hansards/Laws/News).

[2] http://www.speech.sri.com/projects/srilm/download.html

[3] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl

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

the above parallel corpus in SMT. To train vector representations for multi-word phrases, we randomly selected 1M bilingual sentences [5] as training set and used the unsupervised greedy RAE following (Socher et al., 2011). We used a learning rate of 10^{-3} for our minimum distance method that learned the centroid of phrase representations as the vector representation of the corresponding nonterminal.

For projection neural network in Section 3.3, we set 300 units for the hidden layer and dimensionality of 30 for both input and output vectors. Learning rate was set to 10^{-3} and the regularization coefficient λ_L was set to 10^{-3}. To construct the training set for the projection neural network, we selected phrase pairs from our rule table and used their representations on the source and target side as training examples. We randomly selected 5M examples as training set, 10k examples as development set and 10k examples as test set. The multi-layer projection neural network was trained with the back-propagation and stochastic gradient descent algorithm with a mini-batch size of 5k.

Our baseline system is an in-house hierarchical phrase-based system (Chiang, 2007). The features used in the baseline system includes a 4-gram language model trained on the Xinhua section of the English Gigaword corpus, a 3-gram language model trained on the target part of the bilingual training data, bidirectional translation probabilities, bidirectional lexical weights, a word count, a phrase count and a glue rule count.

In order to compare our proposed models with previous methods on nonterminal refinement, we re-implemented a syntax mismatch model (Syn-Mis) which was used by Huang et al. (2013) and integrated it into hierarchical phrase-based system. Syn-Mis model decorates each nonterminal with a distribution of head POS tags and uses this distribution to measure the degree of syntactic compatibility of translation rules with corresponding source spans. In order to obtain head POS tags for Syn-Mis model, we used the Stanford dependency parser [6] (Chang et al., 2009) to parse Chinese sentences in our training corpus and NIST development/test sets.

	MT06	MT08	Avg
Baseline	30.54	23.58	27.06
Syn-Mis	31.23*	24.38*	27.81
MV + CS α = 1.0	31.44$^+$	24.23*	27.84
MV + CS α = 0	31.63*	24.51*	28.07
MV + CS α = -1.0	31.13	24.07*	27.60
MD + ED β = 0	31.02$^+$	23.74	27.38
MD + ED β = 0.5	31.35$^+$	24.08*	27.72
MD + ED β = 1.0	31.06	23.90$^+$	27.48

Table 1: BLEU scores of our models against the baseline and Syn-Mis model. “*” and “+” : significantly better than Baseline at significance level $p < 0.01$ and $p < 0.05$ respectively.

6.2 Different Approaches to Learn Vector Representations for Nonterminals

Our first group of experiments were carried out to investigate which approach is more appropriate to learn semantic vectors for nonterminals. We only used the source-side semantic nonterminal refinement model in these experiments. In order to validate the effectiveness of the proposed approaches for learning nonterminal semantic vectors, we combined the minimum distance method (MD) with the Euclidean Distance (ED) because both of them are distance-based, and combined the weighted mean value method (MV) with the Cosine Similarity model (CS) as they belong to vector-based approaches. We chose α = 1.0, 0, -1.0 and β = 0, 0.5, 1.0 for glue rules to study the impact of these parameters. We compared our model with the baseline and Syn-Mis model.

Results are shown in Table 1. From Table 1, we observe that the proposed two approaches are able to achieve significant improvements over the baseline. (MV + CS) and (MD + ED) achieve up to an absolute improvement of 1.09 and 0.81 (when α = 0 and β = 0.5) BLEU points respectively over the baseline on the development test set MT06. And the approach (MV + CS) with α = 0 outperforms Syn-Mis by 0.4 BLEU points on MT06 without using any syntactic information. The approach (MV + CS) achieves better performance and it is more efficient than (MD + ED) where the computation of semantic centroids is time-consuming. Therefore, we adopt the approach (MV + CS) with α = 0 to learn semantic vectors for nonterminals and compute semantic similarities in the following experiments.

[5] We choose bilingual sentences because we want to obtain bilingual training examples to train our projection neural network as described in Section 3.3.

[6] http://nlp.stanford.edu/software/lex-parser.shtml

	MT06	MT08	Avg
Baseline	30.54	23.58	27.06
Linear Projection	30.70	23.66	27.18
Nonlinear Projection	31.16	24.11*	27.64
Two-pass decoding	31.29+	24.24*	27.77

Table 2: Comparison of two-pass decoding, linear and nonlinear projection methods for integrating the target-side semantic nonterminal refinement model in terms of BLEU scores. "*" and "+" : significantly better than Baseline at significance level $p < 0.01$ and $p < 0.05$ respectively.

6.3 Effect of the Target Semantic Nonterminal Refinement Models

In the second set of experiments, we further validate the effectiveness of semantic nonterminal vectors learned on the target side. In these experiments, learning vector representations and computing semantic similarities were performed on the target language semantic space. We also compared the two integration methods discussed in Section 5 for the target-side model. With regard to the projection method, we further compared the linear projection (the projection neural network without hidden layer) with the nonlinear projection (with hidden layer). Experiment results are shown in Table 2.

From Table 2, we can see that

- Two-pass decoding achieves the highest BLEU scores, which are higher than those of the baseline by 0.75 and 0.66 BLEU points on MT06 and MT08 respectively. The reason may be that noisy translation candidates are filtered out in the first pass. This finding is consistent with many other multiple-pass systems in natural language processing, e.g., two-pass parsing (Zettlemoyer and Collins, 2007).

- Nonlinear projection achieves an improvement of 0.62 BLEU points over the baseline on MT06. It outperforms linear projection method on both sets. These empirical results support our assumption that nonlinear relations between languages are more reasonable than linear relations.

- The results prove that the target-side semantic nonterminal refinement model is also able to improve the baseline system, although the gain is less than that of the source-side counterpart.

	MT06	MT08	Avg
Baseline	30.54	23.58	27.06
Syn-Mis	31.23*	24.38*	27.81
Src Model [1]	31.63*	24.51*	28.07
Trg Model [2]	31.16	24.11*	27.64
Combined-Model	31.71*	24.72*	28.22

[1] (MV + CS $\alpha = 0$) is used.
[2] Nonlinear Projection is used.

Table 3: BLEU scores of the combination of the source- and target-side semantic nonterminal refine model. "*" and "+" : significantly better than Baseline at significance level $p < 0.01$ and $p < 0.05$ respectively.

6.4 Combination of the Source and Target Models

Finally, we integrated both the source- and target-side semantic nonterminal refinement models into the baseline system. In this experiment, we adopted nonlinear projection to obtain target semantic vector representations for target phrases. These two models collectively achieve a gain of up to 1.16 BLEU points over the baseline and 0.41 BLEU points over Syn-Mis model on average, which is shown in Table 3.

7 Conclusion

We have presented a framework to refine nonterminal X in hierarchical translation rules with semantic representations. The semantic vectors are derived from vector representations of phrasal substitutions, which are automatically learned using an unsupervised RAE. As the semantic nonterminal refinement model is capable of selecting more semantically similar translation rules, it achieves statistically significant improvements over the baseline on Chinese-to-English translation. Experiment results have shown that

- Using (MV + CS) approach to learn semantic representations for nonterminals can achieve better performance than (MD + ED) in terms of BLEU scores.

- Target-side semantic nonterminal refinement model is able to substantially improve translation quality over the baseline. Two-pass decoding method is superior to the projection method.

- The simultaneous incorporation of the source- and target-side models can achieve further improvements over a single-side model.

For the future work, we are interested in learning bilingual representations (Lauly et al., 2014; Gouws et al., 2014) for nonterminals. We also would like to extend our work by using more contextual lexical information to derive semantic vectors for nonterminals.

Acknowledgment

The work was sponsored by the National Natural Science Foundation of China (Grants No. 61403269, 61432013 and 61333018) and Natural Science Foundation of Jiangsu Province (Grant No. BK20140355). We would like to thank three anonymous reviewers for their insightful comments.

References

Hala Almaghout, Jie Jiang, and Andy Way. 2011. Ccg contextual labels in hierarchical phrase-based smt. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*.

Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Hailong Cao, Dongdong Zhang, Ming Zhou, and Tiejun Zhao. 2014. Soft dependency matching for hierarchical phrase-based machine translation. In *COLING*.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452. Association for Computational Linguistics.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2011. Utilizing target-side semantic role labels to assist hierarchical phrase-based machine translation. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 107–115. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2014. Bilbowa: Fast bilingual distributed representations without word alignments. *arXiv preprint arXiv:1410.2455*.

Greg Hanneman and Alon Lavie. 2013. Improving syntax-augmented machine translation by coarsening the label set. In *HLT-NAACL*, pages 288–297.

Hieu Hoang and Philipp Koehn. 2010. Improved translation with source syntax labels. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 409–417. Association for Computational Linguistics.

Zhongqiang Huang, Martin Čmejrek, and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 138–147. Association for Computational Linguistics.

Zhongqiang Huang, Jacob Devlin, and Rabih Zbib. 2013. Factored soft source syntactic constraints for hierarchical machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 556–566.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861.

Junhui Li, Zhaopeng Tu, Guodong Zhou, and Josef van Genabith. 2012. Head-driven hierarchical phrase-based translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 33–37. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

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

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 46st Annual Meeting on Association for Computational*, pages 236–244.

Markos Mylonakis and Khalil Sima'an. 2011. Learning hierarchical translation structure with linguistic annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 642–652. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 72–80. Association for Computational Linguistics.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Ashish Venugopal, Andreas Zollmann, Noah A Smith, and Stephan Vogel. 2009. Preference grammars: Softening syntactic constraints to improve statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–244. Association for Computational Linguistics.

Jonathan Weese, Chris Callison-Burch, and Adam Lopez. 2012. Using categorial grammar to label translation rules. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 222–231. Association for Computational Linguistics.

Luke S Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 678–687.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141. Association for Computational Linguistics.

Andreas Zollmann and Stephan Vogel. 2011. A word-class approach to labeling pscfg rules for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1–11. Association for Computational Linguistics.