Bilingually-Constrained Recursive Neural Networks with Syntactic Constraints for Hierarchical Translation Model

Wei Chen(✉) and Bo Xu

IDMTech, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{wei.chen.media,xubo}@ia.ac.cn

Abstract. Hierarchical phrase-based translation models have advanced statistical machine translation (SMT). Because such models can improve leveraging of syntactic information, two types of methods (leveraging source parsing and leveraging shallow parsing) are applied to introduce syntactic constraints into translation models. In this paper, we propose a bilingually-constrained recursive neural network (BC-RNN) model to combine the merits of these two types of methods. First we perform supervised learning on a manually parsed corpus using the standard recursive neural network (RNN) model. Then we employ unsupervised bilingually-constrained tuning to improve the accuracy of the standard RNN model. Leveraging the BC-RNN model, we introduce both source parsing and shallow parsing information into a hierarchical phrase-based translation model. The evaluation demonstrates that our proposed method outperforms other state-of-the-art statistical machine translation methods for National Institute of Standards and Technology 2008 (NIST 2008) Chinese-English machine translation testing data.

1 Introduction

Hierarchical phrase-based models [1] have advanced statistical machine translation (SMT) by employing hierarchical rules. Formally, a hierarchical phrase-based model is a synchronous context-free grammar that is learned from a bitext without any syntactic information. Thus, such models can be considered to be a shift in the formal machinery of syntax-based translation systems without any linguistic commitment, which enables their convenient and extensive application.

Numerous studies have leveraged syntactic information in SMT systems. Some of these studies have introduced linguistic syntax via source parsing to direct word reordering. For example, [2] used dependency tree to add syntactic cohesion. [3] proposed to parse and to translate jointly by taking tree-based translation as parsing. [4] propose a nonparametric Bayesian method for inducing Part-of-Speech (POS) tags in dependency trees to improve the performance of statistical machine translation. Such methods are performed within the unit of tree nodes and efficiently address some mistakes such as word reordering in SMT. However, they cause data sparseness and are vulnerable to parsing errors because

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 388–395, 2015.
DOI: 10.1007/978-3-319-25207-0_34

of their strict constraints on the parse tree. Other studies have employed shallow parsing (also chunking) to justify the selection of translation rules. [5] presented a chunk-to-string translation model where the decoder generates a translation by first translating the words in each chunk, then reordering the translation of chunks. [6] present a hierarchical chunk-to-string translation model, which can be seen as a compromise between the hierarchical phrase-based model and the tree-to-string model. The constraints on the syntactic information that are applied in these studies are significantly weaker. However, these methods tend to suffer from the conflict between the different definitions of phrases in SMT and traditional chunking methods: phrases in SMT are grammar-free, whereas traditional chunking methods require phrases to be intact in terms of grammar.

In this paper, we present a bilingually-constrained recursive neural network (BC-RNN) model to combine the merits of the two types of studies. First, we propose a standard recursive neural network (RNN) model to perform supervised learning to determine how to parse phrases and how to represent phrases in a continuous vector space of features [7] for source- and target- languages, respectively. A simple softmax layer is employed in this model to predict syntactic categories (also chunk labels). Second, we propose a bilingually-constrained learning model to fine-tune the parameters of the standard RNN to improve the accuracy of the representation and chunk labels of the phrases. Finally, by leveraging the BC-RNN model, we extract information about parsing and chunking from the source sentence and efficiently add extra syntactic features to state-of-the-art hierarchical phrase-based translation systems.

Using the 2008 National Institute of Standards and Technology (NIST) Chinese-English MT translation test set, the results of the experiments demonstrate that our model can significantly improve the performance of hierarchical phrase-based translation models and outperform other state-of-the-art SMT methods that leverage syntactic information.

2 Bilingually-Constrained RNN

In this section, we describe the structure of the BC-RNN model. We also define the objective function and the inferences of the parameters of the BC-RNN model.

2.1 The BC-RNN Model

Assume that we are given the phrase $w_1 w_2 ... w_m$; it is projected onto a list of vectors $(x_1, x_2, ..., x_m)$ using word vector representation. The standard RNN learns the parsing tree and the distributed representation of the phrase by recursively combining two child vectors in a bottom-up manner. Given the distributed representation p of the phrase $w_1 w_2 ... w_m$, it is convenient to add a simple softmax layer to predict chunk labels, such as NP and VP. The details of structure prediction and Category Classifier using standard RNN can be got from [7] and we don't introduce it in this paper.

Given two standard RNN models, we propose a bilingually-constrained optimization to fine-tune the parameters of both standard RNN models. The structure of the BC-RNN model is illustrated in Figure 1.

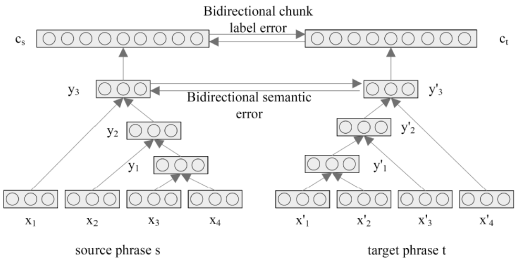

Fig. 1. Structure of bilingually-constrained recursive neural network

2.2 The Objective Function

To fine-tune the standard RNN models for the source and target languages, for a bilingual phrase pair (s, t), two types of errors are involved:

(1) Semantic error: this is quantified in terms of the semantic distance of the distributed representation p_s and p_t of the bilingual phrase pair (s, t) [9].

Because word embeddings for two languages are learned separately and located in different vector spaces, a transformation must be performed to calculate the semantic distance. Thus, the semantic distance is bidirectional: there is both the distance between p_t and the transformation of p_s, and the distance between p_s and the transformation of p_t. Consequently, the total semantic error becomes

$$E_{sem}(s, t; \theta) = E_{sem}(s|t, \theta) + E_{sem}(t|s, \theta) \tag{1}$$

where θ denotes the parameters of the BC-RNN model and we calculate $E_{sem}(s|t, \theta)$ using the Euclidean distance:

$$E_{sem}(s|t, \theta) = \frac{1}{2}||p_t - f(W_{en}^{ch} p_s + b_{en}^{ch})||^2 \tag{2}$$

$E_{sem}(t|s, \theta)$ can be calculated in exactly the same manner.

(2) Chunk label error: this is quantified by the difference between the predicted chunk labels of the distributed representations p_s and p_t.

After applying the simple softmax layer of each standard RNN models, the output vector representations c_s and c_t denote the probability distribution of the chunk labels. Consequently, similar to semantic error, the total chunk label error becomes bidirectional as follows:

$$E_{chunk}(s, t; \theta) = E_{chunk}(s|t, \theta) + E_{chunk}(t|s, \theta) \tag{3}$$

where θ denotes the parameters of the BC-RNN model and $E_{chunk}(s|t,\theta)$ is calculate as follows:

$$E_{chunk}(s|t,\theta) = \frac{1}{2}||c_t - f(W^{ch}_{chunk}p_s + b^{ch}_{chunk})||^2 \tag{4}$$

Thus, for a bilingual phrase pair (s,t), the joint error is

$$E(s,t;\theta) = \alpha E_{sem}(s,t;\theta) + (1-\alpha)E_{chunk}(s,t;\theta) \tag{5}$$

The hyper-parameter α weights the semantic and chunk label errors. The final BC-RNN objective function over the phrase pairs training set (S,T) becomes:

$$J_{BC-RNN} = \sum_{(s,t)\in(S,T)} E(s,t;\theta) + \frac{\lambda}{2}||\theta||^2 \tag{6}$$

2.3 Parameter Inference

The parameter θ can be divided into the source-side parameter θ_s and the target-side parameter θ_t [9]. We apply the stochastic gradient descent (SGD) algorithm to optimize each parameter. Word vector representations θ_L are initialized with the DNN toolkit Word2Vec [8] using large-scale monolingual data, and other parameters are randomly initialized. The details of optimization of the parameters can be got from [9].

3 A Hierarchical Phrase-Based Translation Model that Leverages Syntactic Information

In this section, we leverage two types of syntactic information in the hierarchical phrase-based translation model.

3.1 Feature1: The Score of the Parse Tree

First, we calculate the score of this tree by applying the fine-tuned parameter θ^*_s in the BC-RNN model.

Given the fine-tuned parameters W^{r*}_s and b^{r*}_s, the distributed representation of each nonterminal in this tree is calculated as

$$p = f(W^{r*}_s[c_1 : c_2] + b^{r*}_s) \tag{7}$$

where the concatenation of two children $[c_1 : c_2]$ is provided by word vector representation. Then, we calculate the parsing score similarly to [7] as follows:

$$s = W_{score}p \tag{8}$$

Let $T(y_i)$ denote the set of spans coming from all nonterminal nodes of this parse tree. The total parsing score of this tree is calculated as the sum of the scores of each span:

$$s_{parse}(f,e,a) = \sum_{d \in T(y_i)} s_d(c_1, c_2) \tag{9}$$

where "f" represents the target sentence and "e" represents the source sentence and "a" represents the word alignment. The more details of parsing score can be got from [7] and we don't introduce it in this paper.

3.2 Feature2: Hierarchical Rules with Syntactic Categories

We now discuss how to extract chunk-based hierarchical rules, as Basic phrases are defined using the same heuristic as in previous systems [11][12]. Chunk-based hierarchical phrases are extracted as follows:

1. If $< f_{j_1}^{j_2}, e_{i_1}^{i_2} >$ is a basic phrase with a chunk label c (source-side only), then a rule is extracted:

$$X \rightarrow < f_{j_1}^{j_2}, e_{i_1}^{i_2} > \tag{10}$$

2. Assume that $X \rightarrow < \alpha, \beta >$ is a rule with $\alpha = \alpha_1 f_{j_1}^{j_2} \alpha_2$ and $\beta = \beta_1 e_{i_2}^{i_1} \beta_2$. Let $C(X)$ denotes the set of chunk labels of the nonterminals in this rule, then we update the rule as:

$$X \rightarrow < \alpha_1 X_1 \alpha_2, \beta_1 X_1 \beta_2 > \qquad X_1 \in C(X) \cup c \tag{11}$$

The chunk labels are tagged using the fine-tuned parameter θ_s^* in the BC-RNN model. We evaluate the distribution of these rules in the same manner as [13].

Leveraging chunk-based hierarchical rules, we introduce information about chunk labels into the hierarchical translation model. In the translating decoding process, we select a penalization of incorrect chunk labels as our loss function and add a penalization term to each incorrect decision. Let T_X denote the set of applied hierarchical rules in the decoding process. The penalization can be derived as

$$s_{chunk}(f,e,a) = \sum_{X \in T_X} \sum_{n \in N(X)} 1\{c(n) \notin C(X)\} \tag{12}$$

where $N(X)$ represents the set of the nonterminals of the hierarchical rule X and $c(n)$ represents the chunk labels (source-side only) of the subphrase that covers the nonterminal n and is given by BC-RNN model.

The Hierarchical Translation Model with Syntactic Information. Finally, we introduce both the features into the standard hierarchical translation model [1]. The formula can be derived as follows:

$$\begin{aligned} s_{trans}(f,e,a) = & \sum_i \lambda_i log(s_i(f,e,a)) \\ & + \lambda_{parse} log(s_{parse}(f,e,a)) + \lambda_{chunk} log(s_{chunk}(f,e,a)) \end{aligned} \tag{13}$$

where $s_i(f, e, a)$ represent the traditional features used in standard hierarchical translation model which is as same as [1]. The weights λ_i and λ_{parse} and λ_{chunk} are learned via minimum error-rate training [14] using the development dataset.

4 Experiments

4.1 Data Preparation and Tools

We got training data for source-side standard RNN model from the standard Chinese Treebank (CTB) 6.0, which has 780k words with several categories. The English Treebank corpus (ETB), which contains 2,881k tokens is used to train the target-side standard RNN model.

We used the NIST training set for Chinese-English translation tasks excluding the Hong Kong Law and Hong Kong Hansard as the training data, which contains 470K sentence pairs. For the training data set, we first performed word alignment in both directions using GIZA++ toolkit [10] then refined the alignments using "final-and". We trained a 5-gram language model with modified Kneser-Ney smoothing on The English Gigaword corpus, section AFP which contains 611,506,174 words. we employ an out-of-the-box toolkit Moses (v3.0) framework and minimum error rate training [14] to train and tune the feature weights of SMT systems. We used our in-house Chinese-English data set as the development set and used the 2008 NIST Chinese-English MT test set (1859 sentences) as the test set. Our evaluation metric is BLEU-4.

We employed the Stanford Chinese word segmentation tools to segment the Chinese sentences in the training and testing process.

4.2 Machine Translation Performance

First, we evaluate the performance of our method and compare it with other state-of-the-art methods, including a phrase-based machine translation model [11], a standard hierarchical phrase-based machine translation model [1], a tree-to-string machine translation model that leverages source parsing [4], and a chunk-to-string machine translation model that leverages shallow parsing [6]. The comparison of the performance is shown in Table.1.

Table 1. Translation performance of different methods on NIST 2008

Methods	NIST2008 %
phrase-based	23.25
hierarchical phrase-based	23.94
Tree-to-string	24.1
chunk-to-string	24.8
Feature1 only	25.56
Feature2 only	25.35
Feature1+Feature2	26.02

Table 2. Translation performance using different chunkers on NIST 2008

Methods	NIST2008 %
Feature2 with Left corner PCFG	23.5
Feature2 with Standford parser	24.0
Feature2 with BC-RNN	25.35

The results indicate that using either the score for parsing tree or the chunk-based penalization function can effectively improve the performance of the standard hierarchical translation model. When we integrate both features, the model outperforms the other translation model and can significantly improve the performance of machine translation.

Moreover, for additional analysis, we use the traditional chunker in the Feature2 instead of our BC-RNN model and compare the translation performance with our method in Table.2. The method "Left corner PCFG" is obtained from [15]. The Stanford parser is an out-of-the-box parsing system [16] with the latest version.

The results show that our BC-RNN model can integrate syntactic information into hierarchical translation models more effectively and accurately than traditional chunkers.

5 Conclusion

In this paper, we propose a bilingually-constrained RNN model to introduce high-quality syntactic information into the standard hierarchical translation model. We combine the merits of the two types of studies and propose a bilingually-constrained tuning to improve the quality of syntactic information. The evaluation demonstrate that our method outperforms other state-of-the-art SMT systems.

References

1. Chiang, D.: Hierarchical phrase-based translation. Computational Linguistics **33**(2), 201–228 (2007)
2. Cherry, C.: Cohesive phrase-based decoding for statistical machine translation. In: ACL 2008, pp. 72–80 (2008)
3. Liu, Y., Liu, Q.: Joint parsing and translation. In: ACL 2010, pp. 707–715 (2010)
4. Tamura, A., Watanabe, T., Sumita, E., et al.: Part-of-speech induction in dependency trees for statistical machine translation. In: ACL (1) 2013 (2013)
5. Watanabe, T., Sumita, E., Okuno, H.G.: Chunk-based statistical translation. In: ACL 2003, pp. 303–310 (2003)
6. Feng, Y., Zhang, D., Li, M., et al.: Hierarchical chunk-to-string translation. In: Association for computational linguistics 2012, pp. 950–958 (2012)
7. Socher, R., Manning, C.D., Ng, A.Y.: Learning continuous phrase representations and syntactic parsing with recursive neural networks. In: NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop 2010, pp. 1–9 (2010)

8. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 2013 (2013)
9. Zhang, J., Liu, S., Li, M., Zhou, M., Zong, C.: Bilingually-constrained phrase embeddings for machine translation. In: EMNLP 2014 (2014)
10. Och, F.J., Ney, H.: Improved statistical alignment models. In: Proceedings of ACL, pp. 440–447 (2000)
11. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 NAACL (2003)
12. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: ACL 2005 (2005)
13. Liu, Y., Liu, Q., Lin, S.: Tree-to-string alignment template for statistical machine translation. In: ACL 2006 (2006)
14. Och, F.J.: Minimum error rate training in statistical machine translation. In: ACL 2003 (2003)
15. Manning, C.D., Carpenter, B.: Probabilistic parsing using left corner language models. In: 5th ACL International Workshop (1997)
16. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL, pp. 423–430 (2003)