Learning Hierarchical Translation Spans

Jingyi Zhang[1,2], **Masao Utiyama**[3], **Eiichro Sumita**[3], **Hai Zhao**[1,2]
[1]Center for Brain-Like Computing and Machine Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong Unviersity, Shanghai, 200240, China
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong Unviersity, Shanghai, 200240, China
[3]National Institute of Information and Communications Technology
3-5Hikaridai, Keihanna Science City, Kyoto, 619-0289, Japan
zhangjingyizz@gmail.com, mutiyama/eiichiro.sumita@nict.go.jp,
zhaohai@cs.sjtu.edu.cn

Abstract

We propose a simple and effective approach to learn translation spans for the hierarchical phrase-based translation model. Our model evaluates if a source span should be covered by translation rules during decoding, which is integrated into the translation system as soft constraints. Compared to syntactic constraints, our model is directly acquired from an aligned parallel corpus and does not require parsers. Rich source side contextual features and advanced machine learning methods were utilized for this learning task. The proposed approach was evaluated on NTCIR-9 Chinese-English and Japanese-English translation tasks and showed significant improvement over the baseline system.

1 Introduction

The hierarchical phrase-based (HPB) translation model (Chiang, 2005) has been widely adopted in statistical machine translation (SMT) tasks. The HPB translation rules based on the synchronous context free grammar (SCFG) are simple and powerful.

One drawback of the HPB model is the applications of translation rules to the input sentence are highly ambiguous. For example, a rule whose English side is "X1 by X2" can be applied to any word sequence that has "*by*" in them. In Figure 1, this rule can be applied to the whole sentence as well as to "*experiment by tomorrow*".

In order to tackle rule application ambiguities, a few previous works used syntax trees. Chiang (2005) utilized a syntactic feature in the HPB model, which represents if the source span covered by a translation rule is a syntactic constituent. However, the experimental results showed this feature gave no significant improvement. Instead of using the undifferentiated constituency feature, (Marton and Resnik, 2008) defined different soft syntactic features for different constituent types and obtained substantial performance improvement. Later, (Mylonakis and Sima'an, 2011) introduced joint probability synchronous grammars to integrate flexible linguistic information. (Liu et al., 2011) proposed the soft syntactic constraint model based on discriminative classifiers for each constituent type and integrated all of them into the translation model. (Cui et al., 2010) focused on hierarchical rule selection using many features including syntax constituents.

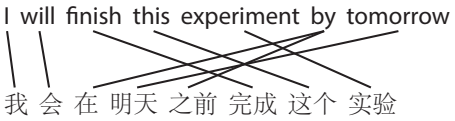

Figure 1: A translation example.

These works have demonstrated the benefits of using syntactic features in the HPB model. However, high quality syntax parsers are not always easily obtained for many languages. Without this problem, word alignment constraints can also be used to guide the application of the rules.

Suppose that we want to translate the English sentence into the Chinese sentence in Figure 1, a translation rule can be applied to the source span "*finish this experiment by tomorrow*". Nonetheless, if a rule is applied to "*experiment by*", then the Chinese translation can not be correctly obtained, because the target span projected from "*ex-*

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 183–188,
October 25-29, 2014, Doha, Qatar.

periment by" contains words projected from the source words outside "*experiment by*".

In general, a translation rule projects one continuous source word sequence (source span) into one continuous target word sequence. Meanwhile, the word alignment links between the source and target sentence define the source spans where translation rules are applicable. In this paper, we call a source span that can be covered by a translation rule without violating word alignment links a *translation span*.

Translation spans that have been correctly identified can guide translation rules to function properly, thus (Xiong et al., 2010) attempted to use extra machine learning approaches to determine boundaries of translation spans. They used two separate classifiers to learn the beginning and ending boundaries of translation spans, respectively. A source word is marked as beginning (ending) boundary if it is the first (last) word of a translation span. However, a source span whose first and last words are both boundaries is not always a translation span. In Figure 1, "*I*" is a beginning boundary since it is the first word of translation span "*I will*" and "*experiment*" is an ending boundary since it is the last word of translation span "*finish this experiment*" , but "*I will finish this experiment*" is not a translation span. This happens because the translation spans are nested or hierarchical. Note that (He et al., 2010) also learned phrase boundaries to constrain decoding, but their approach identified boundaries only for monotone translation.

In this paper, taking fully into account that translation spans being nested, we propose an approach to learn hierarchical translation spans directly from an aligned parallel corpus that makes more accurate identification over translation spans.

The rest of the paper is structured as follows: In Section 2, we briefly review the HPB translation model. Section 3 describes our approach. We describe experiments in Section 4 and conclude in Section 5.

2 Hierarchical Phrase-based Translation

Chiang's HPB model is based on a weighted SCFG. A translation rule is like: $X \rightarrow \langle \gamma, \alpha, \sim \rangle$, where X is a nonterminal, γ and α are source and target strings of terminals and nonterminals, and $\sim$ is a one-to-one correspondence between nonterminals in γ and α. The weight of each rule is:

$$w\left(X \rightarrow \langle \gamma, \alpha, \sim \rangle\right) = \prod_t h_t(X \rightarrow \langle \gamma, \alpha, \sim \rangle)^{\lambda_t} \quad (1)$$

where h_t are the features defined on the rules.

Rewriting begins with a pair of linked start symbols and ends when there is no nonterminal left. Let D be a derivation of the grammar, $f\left(D\right)$ and $e\left(D\right)$ be the source and target strings generated by D. D consists of a set of triples $\langle r, i, j \rangle$, each of which stands for applying a rule r on a span $f\left(D\right)_i^j$. The weight of D is calculated as:

$$w\left(D\right) = \prod_{\langle r,i,j \rangle \in D} w\left(r\right) \times P_{lm}(e)^{\lambda_{lm}} \times \exp\left(-\lambda_{wp}\left|e\right|\right) \quad (2)$$

where $w\left(r\right)$ is the weight of rule r, the last two terms represent the language model and word penalty, respectively.

3 Learning Translation Spans

We will describe how to learn translation spans in this section.

3.1 Our Model

We make a series of binary classifiers $\{C_1, C_2, C_3, ...\}$ to learn if a source span $f\left(D\right)_i^j$ should be covered by translation rules during translation. C_k is trained and tested on source spans whose lengths are k, i.e., $k = j - i + 1$.[1]

C_k learns the probability

$$P_k\left(v | f\left(D\right), i, j\right) \quad (3)$$

where $v \in \{0, 1\}$, $v = 1$ represents a rule is applied on $f\left(D\right)_i^j$, otherwise $v = 0$.

Training instances for these classifiers are extracted from an aligned parallel corpus according to Algorithm 1. For example, "*I will*" and "*will finish*" are respectively extracted as positive and negative instances in Figure 1.

Note that our model in Equation 3 only uses the source sentence $f\left(D\right)$ in the condition. This means that the probabilities can be calculated before translation. Therefore, the predicted probabilities can be integrated into the decoder conveniently as soft constraints and no extra time is added during decoding. This enables us to use rich source contextual features and various machine learning methods for this learning task.

[1] We indeed can utilize just one classifier for all source spans. However, it will be difficult to design features for such a classifier unless only boundary word features are adopted. On the contrary, we can fully take advantage of rich information about inside words as we turn to the fixed span length approach.

3.2 Integration into the decoder

It is straightforward to integrate our model into Equation 2. It is extended as

$$w(D) = \prod_{\langle r,i,j\rangle \in D} w(r) \times P_{lm}(e)^{\lambda_{lm}} \times \exp(-\lambda_{wp}|e|) \times P_k(v=1|f(D),i,j)^{\lambda_k} \quad (4)$$

where λ_k is the weight for C_k.

During decoding, the decoder looks up the probabilities P_k calculated and stored before decoding.

Algorithm 1 Extract training instances.

Input: A pair of parallel sentence f_1^n and e_1^m with word alignments A.

Output: Training examples for $\{C_1, C_2, C_3, ...\}$.

```
for i = 1 to n do
  for j = i to n do
    if ∃e_p^q, 1 ≤ p ≤ q ≤ m
    & ∃(k,t) ∈ A, i ≤ k ≤ j, p ≤ t ≤ q
    & ∀(k,t) ∈ A, i ≤ k ≤ j ↔ p ≤ t ≤ q
    then
      f_i^j is a positive instance for C_{j-i+1}
    else
      f_i^j is a negative instance for C_{j-i+1}
    end if
  end for
end for
```

3.3 Classifiers

We compare two machine learning methods for learning a series of binary classifiers.

For the first method, each C_k is individually learned using the maximum entropy (ME) approach (Berger et al., 1996):

$$P_k(v|f(D),i,j) = \frac{\exp\left(\sum_t \mu_t h_t(v, f(D), i, j)\right)}{\sum_{v'} \exp\left(\sum_t \mu_t h_t(v', f(D), i, j)\right)} \quad (5)$$

where h_t is a feature function and μ_t is weight of h_t. We use rich source contextual features: unigram, bigram and trigram of the phrase $[f_{i-3}, ..., f_{j+3}]$.

As the second method, these classification tasks are learned in the continuous space using feed-forward neural networks (NNs). Each C_k has the similar structure with the NN language model (Vaswani et al., 2013). The inputs to the NN are indices of the words: $[f_{i-3}, ..., f_{j+3}]$. Each source word is projected into an N dimensional vector. The output layer has two output neurons, whose values correspond to $P_k(v=0|f(D),i,j)$ and $P_k(v=1|f(D),i,j)$.

For both ME and NN approaches, words that occur only once or never occur in the training corpus are treated as a special word "UNK" (unknown) during classifier training and predicting, which can reduce training time and make the classifier training more smooth.

4 Experiment

We evaluated the effectiveness of the proposed approach for Chinese-to-English (CE) and Japanese-to-English (JE) translation tasks. The datasets officially provided for the patent machine translation task at NTCIR-9 (Goto et al., 2011) were used in our experiments. The detailed training set statistics are given in Table 1. The development and test

		SOURCE	TARGET
CE	#Sents	954k	
	#Words	37.2M	40.4M
	#Vocab	288k	504k
JE	#Sents	3.14M	
	#Words	118M	104M
	#Vocab	150k	273k

Table 1: Data sets.

sets were both provided for CE task while only the test set was provided for JE task. Therefore, we used the sentences from the NTCIR-8 JE test set as the development set. Word segmentation was done by BaseSeg (Zhao et al., 2006; Zhao and Kit, 2008; Zhao et al., 2010; Zhao and Kit, 2011; Zhao et al., 2013) for Chinese and Mecab [2] for Japanese.

To learn the classifiers for each translation task, the training set and development set were put together to obtain symmetric word alignment using GIZA++ (Och and Ney, 2003) and the *grow-diag-final-and* heuristic (Koehn et al., 2003). The source span instances extracted from the aligned training and development sets were used as the training and validation data for the classifiers.

The toolkit Wapiti (Lavergne et al., 2010) was adopted to train ME classifiers using the classical quasi-newton optimization algorithm with limited memory. The NNs are trained by the toolkit NPLM (Vaswani et al., 2013). We chose "rectifier" as the activation function and the logarithmic loss function for NNs. The number of epochs was set to 20. Other parameters were set to default

[2] http://sourceforge.net/projects/mecab/files/

Span length	CE					JE				
		ME		NN			ME		NN	
	Rate	P	N	P	N	Rate	P	N	P	N
1	2.67	0.93	0.63	0.93	0.64	1.08	0.85	0.79	0.86	0.80
2	1.37	0.83	0.70	0.82	0.75	0.73	0.69	0.84	0.71	0.87
3	0.86	0.70	0.80	0.73	0.83	0.52	0.56	0.89	0.63	0.90
4	0.62	0.57	0.81	0.67	0.88	0.36	0.48	0.93	0.54	0.93
5	0.48	0.52	0.90	0.61	0.91	0.26	0.30	0.96	0.47	0.95
6	0.40	0.47	0.91	0.58	0.92	0.20	0.25	0.97	0.41	0.96
7	0.34	0.40	0.93	0.53	0.93	0.16	0.14	0.98	0.33	0.97
8	0.28	0.35	0.94	0.46	0.94	0.13	0	1	0.32	0.97
9	0.22	0.28	0.96	0.37	0.96	0.10	0	1	0.25	0.98
10	0.15	0.21	0.97	0.28	0.97	0.08	0	1	0.23	0.99

Table 2: Classification accuracies. The Rate column represents ratio of positive instances to negative instances; the P and N columns give classification accuracies for positive and negative instances.

values. The training time of one classifier on a 12-core 3.47GHz Xeon X5690 machine was 0.5h (2.5h) using ME (NN) approach for CE task; 1h (4h) using ME (NN) approach for JE task .

The classification results are shown in Table 2. Instead of the undifferentiated classification accuracy, we present separate classification accuracies for positive and negative instances. The big difference between classification accuracies for positive and negative instances was caused by the unbalanced rate of positive and negative instances in the training corpus. For example, if there are more positive training instances, then the classifier will tend to classify new instances as positive and the classification accuracy for positive instances will be higher. In our classification tasks, there are less positive instances for longer span lengths.

Since the word order difference of JE task is much more significant than that of CE task, there are more negative Japanese translation span instances than Chinese. In JE tasks, the ME classifiers C_8, C_9 and C_{10} predicted all new instances to be negative due to the heavily unbalanced instance distribution.

As shown in Table 2, NN outperformed ME approach for our classification tasks. As the span length growing, the advantage of NN became more significant. Since the classification accuracies deceased to be quite low for source spans with more than 10 words, only $\{C_1, ..., C_{10}\}$ were integrated into the HPB translation system.

For each translation task, the recent version of Moses HPB decoder (Koehn et al., 2007) with the training scripts was used as the baseline (Base). We used the default parameters for Moses, and a 5-gram language model was trained on the target side of the training corpus by IRST LM Toolkit [3] with improved Kneser-Ney smoothing. $\{C_1, ..., C_{10}\}$ were integrated into the baseline with different weights, which were tuned by MERT (Och, 2003) together with other feature weights (language model, word penalty,...) under the log-linear framework (Och and Ney, 2002).

	Method	TER	BLEU-n 4	n-gram precisions 1 2 3 4
CE	Base	49.39- -	33.07- -	69.9/40.7/25.8/16.9
	BLM	48.60	33.93	70.0/41.4/26.6/17.6
	ME	49.02-	33.63-	70.0/41.2/26.3/17.4
	NN	48.09++	34.35++	70.1/41.9/27.0/18.0
JE	Base	57.39- -	30.13- -	67.1/38.3/23.0/14.0
	BLM	56.79	30.81	67.7/38.9/23.6/14.5
	ME	56.48	31.01	67.6/39.0/23.8/14.7
	NN	55.96++	31.77++	67.8/39.7/24.6/15.4

Table 3: Translation results. The symbol ++ (- -) represents a significant difference at the $p < 0.01$ level and - represents a significant difference at the $p < 0.05$ level against the BLM.

We compare our method with the baseline and the boundary learning method (BLM) (Xiong et al., 2010) based on Maximum Entropy Markov Models with Markov order 2. Table 3 reports BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores. Significance tests are conducted using bootstrap sampling (Koehn, 2004). Our ME classifiers achieve comparable translation improvement with the BLM and NN classifiers enhance translation system significantly compared to others. Table 3 also shows that the relative gain was higher for higher n-grams, which is reasonable since the higher n-grams have higher ambiguities in the translation rule application.

It is true that because of multiple parallel sentences, a source span can be applied with transla-

[3] http://hlt.fbk.eu/en/irstlm

tion rules in one sentence pair but not in another sentence pair. So we used the probability score as a feature in the decoding. That is, we did not use classification results directly but use the probability score for softly constraining the decoding process.

5 Conclusion

We have proposed a simple and effective translation span learning model for HPB translation. Our model is learned from aligned parallel corpora and predicts translation spans for source sentence before translating, which is integrated into the translation system conveniently as soft constraints. We compared ME and NN approaches for this learning task. The results showed that NN classifiers on the continuous space model achieved both higher classification accuracies and better translation performance with acceptable training times.

Acknowledgments

Hai Zhao were partially supported by CSC fund (201304490199), the National Natural Science Foundation of China (Grant No.60903119, Grant No.61170114, and Grant No.61272248), the National Basic Research Program of China (Grant No.2013CB329401), the Science and Technology Commission of Shanghai Municipality (Grant No.13511500200), the European Union Seventh Framework Program (Grant No.247619), and the art and science interdiscipline funds of Shanghai Jiao Tong University, a study on mobilization mechanism and alerting threshold setting for online community, and media image and psychology evaluation: a computational intelligence approach.

References

Adam Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2010. A joint rule selection model for hierarchical phrase-based translation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 6–11. Association for Computational Linguistics.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K Tsou. 2011. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proceedings of NTCIR*, volume 9, pages 559–578.

Zhongjun He, Yao Meng, and Hao Yu. 2010. Learning phrase boundaries for hierarchical phrase-based translation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 383–390. Association for Computational Linguistics.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden, July. Association for Computational Linguistics.

Lemao Liu, Tiejun Zhao, Chao Wang, and Hailong Cao. 2011. A unified and discriminative soft syntactic constraint model for hierarchical phrase-based translation. In *the Thirteenth Machine Translation Summit*, pages 253–260. Asia-Pacific Association for Machine Translation.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *ACL*, pages 1003–1011.

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 and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA, October. Association for Computational Linguistics.

Deyi Xiong, Min Zhang, and Haizhou Li. 2010. Learning translation boundaries for phrase-based decoding. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 136–144. Association for Computational Linguistics.

Hai Zhao and Chunyu Kit. 2008. Exploiting unlabeled text with different unsupervised segmentation criteria for chinese word segmentation. *Research in Computing Science*, 33:93–104.

Hai Zhao and Chunyu Kit. 2011. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*, 181(1):163–183.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165. Sydney: July.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for chinese word segmentation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(2):5.

Hai Zhao, Masao Utiyama, Eiichiro Sumita, and Bao-Liang Lu. 2013. An empirical study on word segmentation for chinese machine translation. In *Computational Linguistics and Intelligent Text Processing*, pages 248–263. Springer.