Soft Dependency Matching for Hierarchical Phrase-based Machine Translation

Hailong Cao[1], Dongdong Zhang[2], Ming Zhou[2] and Tiejun Zhao[1]
[1]Harbin Institute of Technology, Harbin, P.R. China
[2]Microsoft Research Asia, Beijing, P.R. China
{hailong, tjzhao}@mtlab.hit.edu.cn
{Dongdong.Zhang, mingzhou}@microsoft.com

Abstract

This paper proposes a soft dependency matching model for hierarchical phrase-based (HPB) machine translation. When a HPB rule is extracted, we enrich it with dependency knowledge automatically learnt from the training data. The dependency knowledge not only encodes the dependency relations between the components inside the rule, but also contains the dependency relations between the rule and its context. When a rule is applied to translate a sentence, the dependency knowledge is used to compute the syntactic structural consistency of the rule against the dependency tree of the sentence. We characterize the structure consistency by three features and integrate them into the standard SMT log-linear model to guide the translation process. Our method is evaluated on multiple Chinese-to-English machine translation test sets. The experimental results show that our soft matching model achieves 0.7-1.4 BLEU points improvements over a strong baseline of an in-house implemented HPB translation system.

1 Introduction

HPB model (Chiang, 2007) is widely used and has consistently delivered state-of-the-art performance. This model extends the phrase-based model (Koehn et al., 2003) by using the formal synchronous grammar to well capture the recursiveness of language during translation. In a formal synchronous grammar, the syntactic unit could be any sequence of contiguous terminals and non-terminals, which may not necessarily satisfy the linguistic constraints. HPB model is powerful to cover any format of translation pairs, but it might introduce ungrammatical rules and produce poor quality translations.

To generate grammatical translations, lots of syntax-based models have been proposed by Galley et al. (2004), Liu et al. (2006), Huang et al. (2006), Mi et al. (2008), Shen et al. (2008), Xie et al. (2011), Zhang et al. (2008), etc. In these models, the syntactic units should be compatible with the syntactic structure of either the source sentence or the target sentence. These approaches can generate more grammatical translations by capturing the structural difference between language pairs. However, these models need special efforts to capture non-syntactic translation knowledge to improve the translation performance.

It is desired to combine the advantages of syntax-based models and the HPB model (Stein et al., 2010). There has been much work trying to improve HPB model by incorporating syntax information. Marton and Resnik (2008) leverage linguistic constituents to constrain the decoding softly. Some work go further to augment the non-terminals in HPB rules with syntactic tags which depend on the syntactic structure covered by the non-terminals (Zollmann and Venugopal, 2006; Chiang, 2010; Li et al., 2012; Huang et al., 2013). For example, given below HPB rules (1-4), the source non-terminal X could be refined into NP or PP as shown in rules (5-8) respectively.

(1) <借 了 X, borrowed X> (2) <借 了 X, lent X>
(3) <X_1 借 了 X_2, borrowed X_2 X_1> (4) <X_1 借 了 X_2, X_1 borrowed X_2>

(5) <借了 NP, borrowed X> (6) <借了 NP, lent X>
(7) <PP 借了 NP, borrow X_2 X_1> (8) <NP 借了 NP, X_1 lent X_2>

Although augmenting the non-terminals with syntactic tags in these methods achieved better results for HPB model, they have limitations that the syntax information on the non-terminals are not discrim-

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2227–2236, Dublin, Ireland, August 23-29 2014.

inative enough due to the limited context covered by the HPB rule. For example, rule (5) and (6) are still not discriminative when translating below two sentences (9) and (10).

(9) 我向他借了一本书(I borrowed a book from him) (10) 我借了一本书给他(I lent a book to him)

where the common phrase "借了一本书" appear in both sentences. Obviously, although rule (5) and (6) share same source sides, rule (5) can only be applied to the translation of sentence (9) and rule (6) to sentence (10). Otherwise, inappropriate application will lead to wrong translations. Rule (5) and (6) are not discriminative due to no consideration of their outside context during the translation.

Motivated by such observation, we proposed an alternative approach, called soft dependency matching model, to incorporate into each HPB rule the source syntactic dependencies connecting the contents inside the rule with the context outside the rule. The dependency knowledge associated with HPB rules is automatically learnt from bilingual training corpus. They make HPB rules discriminative according to global context.

Figure 1. Dependency information associated with two rules. LC and RC mean the source context on the left and right of the rule respectively.

Figure 1 shows two rules associated with different dependencies. The first one is applicable to the case when some word on the left side depends on the word "借" in the rule, and the second one is applicable to the case when the word "借" in the rule depends on some word on the right side.

During SMT decoding, first we parse the source sentence to get the dependency tree. When a HPB rule is applied to translate the sentence, we calculate structural consistency between the dependency knowledge associated with the rule and dependency tree structure of the source sentence. The consistency degree is integrated into the SMT log-linear model as features to encourage syntactic hypotheses and penalize the hypotheses violating syntactic constraints.

Compared with previous work that incorporate syntax knowledge into HPB model, the advantage of our soft dependency matching model is:

- It not only captures the dependency relations between the components inside the rule, but also models the dependency relations between the rule and its context from a global view.
- Without increasing the amount of rules or the searching space, our model can capture the syntactic variation for all of the rules (syntactic or non-syntactic, well-formed or ill-formed).
- Our model can take advantage of the dependency knowledge on both terminals and non-terminals.

We evaluate the performance of our soft dependency matching model on Chinese-to-English translation task. Experimental results show that our method can achieve the improvements of 0.7-1.4 BLEU points over the baseline HPB model on multiple NIST MT evaluation test sets.

2 Related Work

Ever since the invention of phrase-based model, a lot of efforts have been made to incorporate linguistic syntax. Cherry(2008) and Marton and Resnik (2008) leverage linguistic constituent to constrain the decoding softly. In their methods, a translation hypothesis gets an extra credit if it respects the parse tree but may incur a cost if it violates a constituent boundary. The soft constrain based methods achieved promising results on various language pairs. One problem of these methods is that exactly matching syntactic constraints cannot always guarantee a good translation, and violating syntactic structure does not always induce a poor translation. It could be more reasonable if the credit and penalty is learnt from the parallel training data. In this work, we learn this kind of constrain knowledge directly from the syntactic structures over the training corpus.

Xiong et al. (2009) present a method that automatically learns syntactic constraints from training data for the ITG based translation (Wu, 1997; Xiong et al., 2006). They utilize the syntactic constraints to estimates the extent to which a span is bracketable. Though the effect was demonstrated on the ITG based model, the method is also applicable to the HPB model. The main difference between Xiong et al. (2009) and our work is that we try to estimate the structural consistency of each rule

against the source syntax tree. For rules which are same in the source side but different in the target side, our method will distinguish the inconsistency degree for different rules. While, for such rules, Xiong et al. (2009) will give a same score which will be used to compete with rules in other spans.

More recently, Huang et al. (2013) associate each non-terminal with the distribution of tags that is used to measure the consistency of syntactic compatibility of the translation rule on source spans. Our work is similar to Huang et al. (2013) since we also represent the syntactic variation of translation rules in the form of distribution. The main difference is that they annotate non-terminals with head POS tags while we use dependency triples (over both terminals and non-terminals) to explicitly represent both the dependency relations inside the rule, and that between the rule and its context.

Both above related work and our work need parse the source sentence to get syntactic context before decoding. There are also some methods incorporating syntax information without the need of online parsing the source sentences (Zollmann and Venugopal, 2006; Shen et al, 2009; Chiang, 2010). They parse the training data to label the non-terminals with syntactic tags. During the bottom-up decoding, the tags are used to model the substitution of non-terminals in a soft way (Shen et al, 2009; Chiang, 2010) or in a hard way (Zollmann and Venugopal, 2006).

Gao et al. (2011) derive soft constraints from the source dependency parsing for the HPB translation. They focus on the relative order of each dependent word and its head word after translation, while our method models whether the dependency information of a rule matches the context or not.

Our work utilizes contextual information around translation rules. In this sense, it is similar to He et al. (2008) and Liu et al. (2008). The main difference between their work and our work is that they leverage lexical context for rule selection while we focus on the syntactic contextual information.

3 Hierarchical Phrase based Machine Translation

Our model proposed in this paper is an extension of the HPB model (Chiang, 2007). Formally, HPB model is a weighted synchronous context free grammar. It employs a generalization of the standard plain phrase extraction approach in order to acquire the synchronous rules of the grammar directly from word-aligned parallel text. Rules have the form of:

$$X \rightarrow < \gamma, \alpha, \sim >$$

where X is a nonterminal, γ and α are both strings of terminals and non-terminals from source and target side respectively, and $\sim$ is a one-to-one correspondence between nonterminal occurrences in γ and α. Associated with each rule is a set of feature functions with the form $f_i(\gamma, \alpha)$. These feature functions are combined into a log-linear model. When a rule is applied during SMT decoding, its score is calculated as:

$$\sum_i \lambda_i \cdot f_i(\gamma, \alpha)$$

where λ_i is the weight associated with feature function $f_i(\gamma, \alpha)$. The feature weights are typically optimized using minimum error rate training algorithm (Och, 2003).

4 Soft Dependency Matching Model

In order to incorporate syntactic knowledge to refine both the word ordering and word sense disambiguation for HPB model, we propose a soft dependency matching model (SDMM). It extends HPB rule into a form which is named as SDMM rule:

$$X \rightarrow < \gamma, \alpha, \sim, \mathrm{RDT} >$$

where RDT(rule's dependency triples) is a set of dependency triples defined on source string γ. Each element in RDT is a triple representing dependency knowledge in the form:

$$\{m\text{-}h\text{-}l\}$$

where *m* and *h* are the dependent and head respectively, *l* is the label of the dependency relation type. *m* and *h* could be any of terminals, non-terminals, LC and RC, where LC denotes the left context and RC the right context.

In the following two sub-sections, we will explain the details of SDMM rule extensions for both plain phrases (i.e., there are no non-terminals in both γ and α) and hierarchical rules (i.e., there are at

least one non-terminal in both γ and α) respectively. For simplicity, we ignore the correspondence ~ in the representations of both HPB rules and SDMM rules.

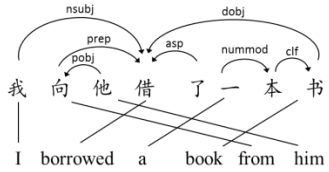


Figure 2: An illustration of a dependency parse tree for the source side of a word-aligned parallel sentences pair.

4.1 SDMM Over Plain Phrase Rules

Figure 2 illustrates a parallel sentence together with word alignments and source dependency parse tree, from which we can extract the phrase pairs of HPB rules like:

(11) < 一 本 书, a book > (12) < 借 了 一 本 书, borrowed a book >

By incorporating syntactic knowledge, we can extend these HPB rules into SDMM rules as shown in Figure 3(a) and Figure 3(b) respectively.

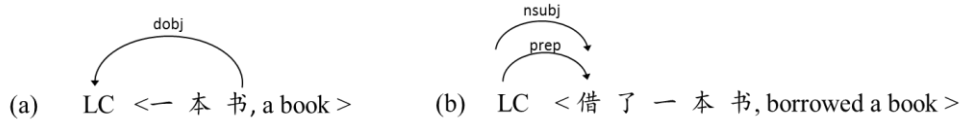


Figure 3: An illustration of two phrase pairs annotated with a set of dependency triples.

Formally, the RDT corresponding to phrase pair (11) is {书-LC-dobj}. The RDT corresponding to phrase pair (12) is {LC-借-nsubj, LC-借-prep}.

Now we describe how to build the RDT when a phrase pair is extracted from a sentence pair during the training step. First, we initialize RDT to be empty. Then, for each dependency triple "*m*-*h*-*l*" in the parse tree of the source sentence, if either *m* or *h* is covered by the source phrase in the rule, we add it to RDT. However, if both *m* and *h* are covered by the source phrase, we will ignore it because it holds less syntactic information beyond HPB rule itself. For example, the dependency triple "一-本-nummod" is excluded from RDT for both phrase pair (11) and phrase pair (12). In addition, we do not add the dependency triple "*m*-*h*-*l*" into RDT if both *m* and *h* are not contained in source phrase, because it is not related to phrase pair at all. The dependency triple "他-向-pobj" is such a case for both phrase pair (11) and phrase pair (12).

Finally, we normalize the word in RDT that is not covered by the source phrase with either LC (stands for the left context) or RC (stands for the right context) according to its relative position to the source phrase. For example, in the RDT for phrase pair (11), we normalize "书-借-dobj" as "书-LC-dobj" since the word "借" is not covered by the source phrase and it is treated as left context.

Note that for each context word outside the source phrase, we only record whether it is on the left or on the right of phrase. We do not further consider its lexical form and its distance to the source phrase. For example, in the two dependency triples in Figure 3(b), both the dependent word "我" and "向" are normalized into LC. In this way, we can generalize the dependency triples in RDT and alleviate the data sparseness problem. In fact, there might be duplicated dependency triples for a phrase pair. In this case, we only keep one of them.

4.2 SDMM over Hierarchical Rules

Hierarchical rules are usually generated by substituting sub-phrases with non-terminals from plain phrase pairs. For example, given the parallel sentence and the two phrase pairs in Section 4.1, we can get a hierarchical rule like:

<借 了 X, borrowed X>

To extend hierarchical rules into SDMM rules, we add dependency information to source terminals or non-terminals in RDT. Figure 4 shows an example representing an SDMM rule:

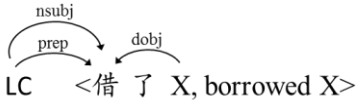


Figure 4: An illustration of a hierarchical rule annotated with a set of dependency triples.

The generation of SDMM rules over hierarchical rules is similar to that of plain phrase rules. The only difference lies in processing the non-terminals, whose dependencies are inferred from the words they covered. For example, the RDT of the above SDMM rule would be:{LC-借-nsubj, LC-借-prep, X-借-dobj}

Similarly, any dependencies over two terminals contained in the source rule are not included in RDT, and dependencies inferred from same non-terminals are excluded as well. In addition, dependencies between two non-terminals are ignored.

4.3 SDMM Rule Composing

A same HPB rule (either plain phrase pair or a hierarchical rule) can be extracted from different bilingual sentences. Therefore, the same HPB rule could be extended into multiple SDMM rules. For example, given a parallel sentence pair shown in Figure 5,

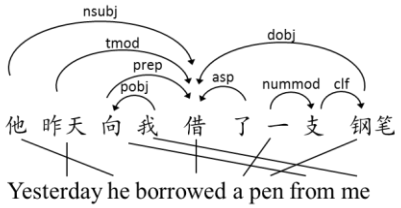


Figure 5: An example of a dependency tree over the source sentence together with the word-aligned target sentence.

we might get a SDMM rule as shown in Figure 6. Compared to the SDMM rule in Figure 4, there is an additional dependency triple "LC-借-tmod" in RDT.

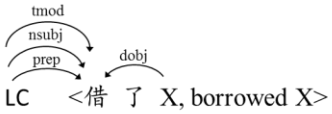


Figure 6: An illustration of dependency triples associated to a hierarchical rule.

Intuitively, we can process SDMM rules independently although they share the same information of HPB rules. However, this will exacerbate the data sparseness problem and make the computation inefficient due to dramatically increased model size. An alternative way is only to keep the most frequent RDT information for the same HPB rules. Though this can get a very concise model, a lot of useful syntactic information might be lost.

We propose a balanced composing method to make a trade-off between knowledge representation and computation efficiency of SDMM rules. Suppose there are more than one SDMM rules with different RDT_i but the same HPB rule, we compose them by the union and get the new form of RDT as:

$$\mathrm{RDT} = \bigcup_i \mathrm{RDT_i}$$

In addition, we record the frequency of HPB rule as well as that of each dependency triple in RDT as:

$$\#(\mathrm{X} \rightarrow < \gamma, \alpha, \sim >) \ , \ \#(t_i, \mathrm{X} \rightarrow < \gamma, \alpha, \sim >)$$

where $\#(\mathrm{X} \rightarrow < \gamma, \alpha, \sim >)$ is the number of times that HPB rule $\mathrm{X} \rightarrow < \gamma, \alpha, \sim >$ is extracted from the training data, and $\#(t_i, \mathrm{X} \rightarrow < \gamma, \alpha, \sim >)$ is the frequency that t_i and $\mathrm{X} \rightarrow < \gamma, \alpha, \sim >$ co-occur. For example, suppose SDMM rules in Figure 4 and Figure 6 occurs 9 and 1 times respectively, we can compose them into the form as shown in Figure 7.

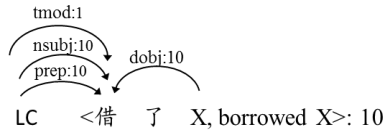


Figure 7: Composed form of the dependency annotation of a rule. The integers following the colons denote occurring times.

Therefore, the composed SDMM rule will be represented by the original HPB rule <借 了 X, borrowed X> together with RDT and its frequency information shown in Table 1.

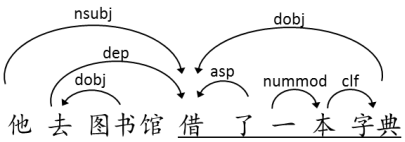
RDT	#
{ LC-借-tmod,	1
LC-借-nsubj,	10
LC-借-prep,	10
X - 借-dobj }	10

Table 1. The RDT and its frequency information of a composed SDMM rule.

4.4 Consistency of SDMM Rules

So far we have described how to enrich a rule with RDT in the training step. Now we introduce how to use the RDT of each rule to guide the translation process.

In the decoding, we parse the source sentence to get the dependency parse tree as shown in Figure 8. When we apply a rule to get a partial translation for a span, we also extract a set of dependency triples based on the parse tree in the exact same way that is used in the training step. We denote this by CDT (context dependency triples). Suppose the rule <借 了 X, borrowed X> is applied to translate the underlined span in Figure 8, then the CDT for the rule is: {LC-借-nsubj, LC-借- dep, X-借-dobj}.



Ref: He went to the library to borrow a dictionary

Figure 8: A sentence to be translated and its dependency parse tree.

In order to evaluate whether a SDMM rule is applicable to translate a sentence or not from the syntactic view, we model the structural consistency of SDMM rule against source dependency tree by calculating the matching degree between RDT and CDT. The example in Figure 9 illustrates how we compute the matching degree between the SDMM rule in Figure 7 and CDT over the source dependency tree in Figure 8. We estimate the matching degree based on three sets including the relative complement set of CDT in RDT, the intersection set of RDT and CDT, and the relative complement set of RDT in CDT.

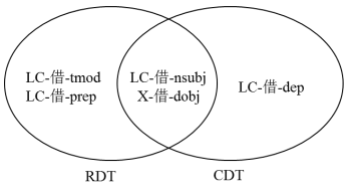


Figure 9: Three different sets of dependency triples to model the structural consistency of syntactic matching.

The statistics over above three sets are leveraged to design three features which are incorporated into SMT log-linear model to encourage and penalize various syntactic motivated hypotheses. The first feature is called as the lost dependency triple feature f_l. It is calculated based on the set RDT\CDT as:

$$f_l = \sum_{t \in RDT \setminus CDT} \delta(\#(t, X \rightarrow < \gamma, \alpha, \sim >) == \#(X \rightarrow < \gamma, \alpha, \sim >))$$

where δ is the indicator function whose value is one if and only if the condition is true, otherwise its value is zero. The motivation of f_l is that: if a dependency triple which always co-occur with the HPB rule is not observed in CDT, it indicates the current SDMM rule may mismatch with the source sentence and therefore we need to penalize its application. In Figure 9, "LC-借-prep" is such a dependency triple. However, for the less frequent dependency triples in RDT such as "LC-借-tmod" in Figure 8, there is no penalty on it although it is not found in CDT.

The second feature is the unexpected dependency triple feature f_u, which is computed as :

$$f_u = |\mathrm{CDT} \backslash \mathrm{RDT}|$$

This feature is the number of dependency triples in CDT that never co-occur with the rule in the training data. In Figure 9, "LC-借-dep" is such a case. Intuitively, the higher the value f_u is, the higher inconsistency degree is, because it means that many dependency triples in CDT are never observed in the training corpus. We should discourage the application of the corresponding SDMM rule.

The third feature is the matched dependency triple feature f_m which is calculated based on RDT∩CDT. It is directly used to model the structural consistency over all the dependency triples in RDT∩CDT for the application of HPB rule $\mathrm{X} \rightarrow < \gamma, \alpha, \sim >$. Formally, f_m is defined as the sum of log probability of each dependency triple in RDT∩CDT conditioned on the HPB rule:

$$f_m = \sum_{t \in \mathrm{RDT} \cap \mathrm{CDT}} \log(P(t|\mathrm{X} \rightarrow < \gamma, \alpha, \sim >))$$

where $\mathrm{P}(t|\mathrm{X} \rightarrow < \gamma, \alpha, \sim >)$ is the probability of a dependency triple t associated to a HPB rule $\mathrm{X} \rightarrow < \gamma, \alpha, \sim >$. We estimate it based on the relative frequency and experimentally use the adding 0.5 smoothing.

5 Experiments

5.1 Experimental Settings

Our baseline is the re-implementation of the Hiero system (Chiang, 2007). When our soft dependency matching model is integrated, the HPB rule is extended into the form of $\mathrm{X} \rightarrow < \gamma, \alpha, \sim, \mathrm{RDT} >$ and the score is calculated by:

$$\sum_i \lambda_i \cdot f_i(\gamma, \alpha) + \lambda_l \cdot f_l(\gamma, \alpha, \mathrm{RDT}) + \lambda_u \cdot f_u(\gamma, \alpha, \mathrm{RDT}) + \lambda_m \cdot f_m(\gamma, \alpha, \mathrm{RDT})$$

where the additional three features are defined in Section 4.3, λ_l, λ_u and λ_m are corresponding feature weights.

We test our soft dependency matching model on a Chinese-English translation task. The NIST06 evaluation data was used as our development set to tune the feature weights, and NIST04, NIST05 and NIST08 evaluation data are our test sets. We first conduct experiments by using the FBIS parallel corpus, and then further test the performance of our method on a large scale training corpus.

Word alignment is performed by GIZA++ (Och and Ney, 2000) in both directions with the default setting. 4-gram language model is trained over the Xinhua portion of LDC English Gigaword Version 3.0 and the English part of the bilingual training data. Feature weights are tuned with the minimum error rate training algorithm (Och, 2003).Translation performance is measured with case-insensitive BLEU4 score (Papineni et al., 2002).

All the Chinese sentences in the training set, development set and test set are parsed by an in-house developed dependency parser based on shift-reduce algorithm (Zhang and Nivre, 2011). There are 45 named grammatical relations plus a default relation representing unknown cases. The detailed descriptions about dependency parsing are explained in Chang et al. (2009).

5.2 Experimental Results on FBIS Corpus

We first conduct experiments by using the FBIS parallel corpus to train the model of both the baseline and the soft dependency matching model. Table 2 shows the statistics of FBIS corpus after the preprocessing.

	#sentences	#words
Chinese	128,832	3,016,570
English	128,832	3,922,816

Table 2. The statistics of FBIS corpus

The evaluation results over FBIS corpus are reported in Table 3. The first row shows the results of baseline, the next three rows show the effect of three features respectively and the last row gives the result when all features are integrated together. Based on Table 3, we can see that each individual feature improves the performance. Among all integrated features, the third feature f_m is the most effective one. The best performance is achieved when using all three features, where we get 1.4, 0.9 and 1.2 BLEU points improvements respectively over the baseline on three test sets.

	NIST04	NIST05	NIST08
Baseline	33.53	32.97	25.08
Baseline+f_l	34.59	33.44	25.69
Baseline+f_u	34.48	33.59	25.51
Baseline+f_m	34.73	33.74	25.76
Baseline+f_l+f_u+f_m	**34.96**	**33.91**	**26.28**

Table 3. Translation performance over BLEU% when models are trained on the FBIS corpus.

5.3 Experimental Results on Large Scale Corpus

To further test the effect of our soft dependency matching model, we use a large scale corpus released by LDC. The catalog number of them is LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10, LDC2005E83, LDC2006E26, LDC2006E34, LDC2006E85 and LDC2006E92. There are 498K sentence pairs, 12.1M Chinese words and 13.8M English words. Table 4 summarizes the translation performance on the large scale of corpus. Our model is still effective when we train the translation system on large scale data. We get 1.3, 0.7 and 1.0 BLEU point improvements over the baseline on three test sets respectively, which shows that our method can consistently improve HPB system over different sized training corpus.

	NIST04	NIST05	NIST08
Baseline	38.72	37.59	29.03
Baseline+f_l+f_u+f_m	**40.00**	**38.34**	**30.06**

Table 4. Translation performance over BLEU% when models are trained on a large scale parallel corpus.

5.4 Decoding Cost

Incorporating syntax can improve the translation performance, but it might increase the SMT decoding complexity. One advantage of our method is that it does not increase the amount of translation rules, so the searching space is not enlarged. Table 5 shows the decoding time comparison with the baseline when models are trained on the FBIS corpus. The average decoding time per sentence is only increased by about 12% due to the parsing of source sentences and the computation of the features. We believe that this is acceptable given the performance gain.

	NIST04	NIST05	NIST08
Baseline	0.67sec	0.78sec	0.50sec
Baseline+f_l+f_u+f_m	0.88sec	0.87sec	0.56sec

Table 5. The average decoding time per sentence, measured in second/sentence.

6 Conclusion and Future Work

We proposed a soft dependency matching model for HPB machine translation. We enrich the HPB rule with dependency knowledge learnt from the training data. The dependency knowledge allows our model to capture the both the dependency relations inside the rule and the dependency relations between the rule and its context from a global view. During decoding, the syntax structural consistency of rules against source dependency tree is calculated and converted into SMT log-linear model fea-

tures to guide the translation process. The experimental results show that our soft matching model achieves significant improvements over a strong baseline of an in-house implemented HPB system.

In future work, there is much room to improve the performance via our method. First, we can discriminatively learn the contribution of the dependency knowledge of each rule based on the training data. Second, we can go beyond the current "bag of dependency triples" representation by composing them hierarchically to capture deep syntactic information. Third, section 2 has discussed the theoretical difference with related work on adding source syntax into the HPB model, we are interested in empirically comparing our method with them and combining it with them to get further improvement.

Acknowledgments

We thank anonymous reviewers for insightful comments. The work of Hailong Cao is sponsored by Microsoft Research Asia Star Track Visiting Young Faculty Program. The work of HIT is also funded by the project of National Natural Science Foundation of China (No. 61173073) and International Science & Technology Cooperation Program of China (No. 2014DFA11350).

Reference

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative Reordering with Chinese Grammatical Relations Features. In Proceedings of *NAACL* Workshop on SSST.

Colin Cherry. 2008. Cohesive Phrase-based Decoding for Statistical Machine Translation. In Proceedings of *ACL*.

David Chiang. 2007. Hierarchical Phrase-based Translation. Computational Linguistics, 33(2):201–228.

Yang Gao, Philipp Koehn, and Alexandra Birch. 2011. Soft Dependency Constraints for Reordering in Hierarchical Phrase-Based Translation. In Proceedings of *EMNLP*.

Zhongjun He, Qun Liu, Shouxun Lin. 2008. Improving Statistical Machine Translation using Lexicalized Rule Selection. In Proceedings of *COLING*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended-domain of locality. In Proceedings of *AMTA.*

Zhongqiang Huang, Martin Čmejrek, and Bowen Zhou. 2010. Soft Syntactic Constraints for Hierarchical Phrase-based Translation Using Latent Syntactic Distributions. In Proceedings of *EMNLP*.

Zhongqiang Huang, Jacob Devlin, and Rabih Zbib. 2013. Factored Soft Syntactic Contraints for Hierarchical Machine Translation. In Proceedings of *EMNLP*.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In Proceedings of *ACL*.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In Proceedings of *ACL*.

Philipp Koehn, Franz Josef Och, Daniel Marcu. 2003. Statistical phrase based translation. In Proceedings of *NAACL.*

Junhui Li, Zhaopeng Tu, Guodong Zhou, and Josef van Genabith. 2012. Using Syntactic Head Information in Hierarchical Phrase-based Translation. In Proceedings of *WMT*.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree to-string alignment template for statistical machine translation. In Proceedings of *ACL.*

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In Proceedings of *ACL*.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In Proceedings of *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of *ACL*.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In Proceedings of *ACL.*

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 *EMNLP*.

Daniel Stein, Stephan Peitz, David Vilar, and Hermann Ney. 2010. A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation. In Conference of the Association for Machine Translation in the Americas.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377–404.

Jun Xie, Haitao Mi and Qun Liu. 2011. A Novel Dependency-to-String Model for Statistical Machine Translation. In Proceedings of *EMNLP*.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In Proceedings of *ACL*.

Deyi Xiong, Min Zhang, Aiti Aw, Haizhou Li. 2009. A Syntax-Driven Bracketing Model for Phrase-Based Translation. In Proceedings of *ACL*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In Proceedings of *NAACL* Workshop on SMT.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In Proceedings of *ACL*.

Yue Zhang and Joakim Nivre. 2011. Transition-based Dependency Parsing with Rich Non-local Features In Proceedings of *ACL*.