Factored Soft Source Syntactic Constraints for Hierarchical Machine Translation

Zhongqiang Huang
Raytheon BBN Technologies
50 Moulton St
Cambridge, MA, USA
zhuang@bbn.com

Jacob Devlin
Raytheon BBN Technologies
50 Moulton St
Cambridge, MA, USA
jdevlin@bbn.com

Rabih Zbib
Raytheon BBN Technologies
50 Moulton St
Cambridge, MA, USA
rzbib@bbn.com

Abstract

This paper describes a factored approach to incorporating soft source syntactic constraints into a hierarchical phrase-based translation system. In contrast to traditional approaches that directly introduce syntactic constraints to translation rules by explicitly decorating them with syntactic annotations, which often exacerbate the data sparsity problem and cause other problems, our approach keeps translation rules intact and factorizes the use of syntactic constraints through two separate models: 1) a syntax mismatch model that associates each nonterminal of a translation rule with a distribution of tags that is used to measure the degree of syntactic compatibility of the translation rule on source spans; 2) a syntax-based reordering model that predicts whether a pair of sibling constituents in the constituent parse tree of the source sentence should be reordered or not when translated to the target language. The features produced by both models are used as soft constraints to guide the translation process. Experiments on Chinese-English translation show that the proposed approach significantly improves a strong string-to-dependency translation system on multiple evaluation sets.

1 Introduction

Hierarchical phrase-based translation models (Chiang, 2007) are widely used in machine translation systems due to their ability to achieve local fluency through phrasal translation and handle non-local phrase reordering using synchronous context-free grammars. A large number of previous works have tried to introduce grammaticality to the translation process by incorporating syntactic constraints into hierarchical translation models. Despite some differences in the granularity of syntax units (e.g., tree fragments (Galley et al., 2004; Liu et al., 2006), treebank tags (Shen et al., 2008; Chiang, 2010), and extended tags (Zollmann and Venugopal, 2006)), most previous work incorporates syntax into hierarchical translation models by explicitly decorating translation rules with syntactic annotations. These approaches inevitably exacerbate the data sparsity problem and cause other problems such as increased grammar size, worsened derivational ambiguity, and unavoidable parsing errors (Hanneman and Lavie, 2013).

In this paper, we propose a factored approach that incorporates soft source syntactic constraints into a hierarchical string-to-dependency translation model (Shen et al., 2008). The general ideas are applicable to other hierarchical models as well. Instead of enriching translation rules with explicit syntactic annotations, we keep the original translation rules intact, and factorize the use of source syntactic constraints through two separate models.

The first is a *syntax mismatch model* that introduces source syntax into the nonterminals of translation rules, and measures the degree of syntactic compatibility between a translation rule and the source spans it is applied to during decoding. When a hierarchical translation rule is extracted from a parallel training sentence pair, we determine a tag for each nonterminal based on the dependency parse of the source sentence. Instead of fragmenting rule statistics by directly labeling nonterminals with tags,

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 556–566,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

we keep the original string-to-dependency translation rules intact and associate each nonterminal with a distribution of tags. That distribution is then used to measure the syntactic compatibility between the syntactic context from which the translation rule is extracted and the syntactic analysis of a test sentence.

The second is a *syntax-based reordering model* that takes advantage of phrasal cohesion in translation (Fox, 2002). The reordering model takes a pair of sibling constituents in the source parse tree as input, and uses source syntactic clues to predict the ordering distribution (straight vs. inverted) of their translations on the target side. The resulting ordering distribution is used in the decoder at the word pair level to guide the translation process. This separate reordering model allows us to utilize source syntax to improve reordering in hierarchical translation models without having to explicitly annotate translation rules with source syntax.

Our results show that both the syntax mismatch model and the syntax-based reordering model are able to achieve significant gains over a strong Chinese-English MT baseline. The rest of the paper is organized as follows. Section 2 discusses related work in the literature. Section 3 provides an overview of our baseline string-to-dependency translation system. Section 4 describes the details of the syntax mismatch and syntax-based reordering models. Experimental results are presented in Section 5. The last section concludes the paper.

2 Related Work

Attempts to use rich syntactic annotations do not always result in improved performance when compared to purely hierarchical models that do not use linguistic guidance. For example, as shown in (Mi and Huang, 2008), tree-to-string translation models (Huang et al., 2006) only start to outperform purely hierarchical models when significant efforts were made to alleviate parsing errors by using forest-based approaches in both rule extraction and decoding. Using only syntactic phrases is too restrictive in phrasal translation as many useful phrase pairs are not syntactic constituents (Koehn et al., 2003). The syntax-augmented translation model of Zollmann and Venugopal (2006) annotates nonterminals in hierarchical rules with thousands of extended syntactic categories in order to capture the syntactic variations of phrase pairs. This results in exacerbated data sparsity problems, partially due to the requirement of exact matches in nonterminal substitutions between translation rules in the derivation. Several solutions were proposed. Shen et al. (2009) and Chiang (2010) used soft match features to explicitly model the substitution of nonterminals with different labels; Venugopal et al. (2009) used a preference grammar to soften the syntactic constraints through the use of a preference distribution of syntactic categories; and recently Hanneman and Lavie (2013) proposed a clustering approach to reduce the number of syntactic categories. Our proposed syntax mismatch model associates nonterminals with a distribution of tags. It is similar to the preference grammar in (Venugopal et al., 2009); however, we use treebank tags and focus on the syntactic compatibility between translation rules and the source sentence. The work of Huang et al. (2010) is most similar to ours, with the main difference being that their syntactic categories are latent and learned automatically in a data driven fashion while we simply use treebank tags based on dependency parsing. Marton and Resnik (2008) also exploited soft source syntax constraints without modifying translation rules. However, they focused on the quality of translation spans based on the syntactic analysis of the source sentence, while our method explicitly models the syntactic compatibility between translation rules and source spans.

Most research on reordering in machine translation focuses on phrase-based translation models as they are inherently weak at non-local reordering. Previous efforts to improve reordering for phrase-based systems can be largely classified into two categories. Approaches in the first category try to reorder words in the source sentence in a preprocessing step to reduce reordering in both word alignment and MT decoding. The reordering decisions are either made using manual or automatically learned rules (Collins et al., 2005; Xia and McCord, 2004; Xia and McCord, 2004; Genzel, 2010) based on the syntactic analysis of the source sentence, or constructed through an optimization procedure that uses feature-based reordering models trained on a word-aligned parallel corpus (Tromble and Eisner, 2009;

Khapra et al., 2013). Approaches in the second category try to explicitly model phrase reordering in the translation process. These approaches range from simple distance based distortion models (Koehn et al., 2003) that globally penalizes reordering based on the distorted distance, to lexicalized reordering models (Koehn et al., 2005; Al-Onaizan and Papineni, 2006) that assign reordering preferences of adjacent phrases for individual phrases, and to hierarchical reordering models (Galley and Manning, 2008; Cherry, 2013) that handle reordering preferences beyond adjacent phrases. Although hierarchical translation models are capable of handling nonlocal reordering, their accuracy is far from perfect. Xu et al. (2009) showed that the syntax-augmented hierarchical model (Zollmann and Venugopal, 2006) also benefits from reordering source words in a preprocessing step. Explicitly adding syntax to translation rules helps with reordering in general, but it introduces additional complexities, and is still limited by the context-free nature of hierarchical rules. Our work exploits an alternative direction that uses an external reordering model to improve word reordering of hierarchical models. Gao et al. (2011), Xiong et al. (2012), and Li et al. (2013) also studied external reordering models for hierarchical models. However, they focused on specific word pairs such as a word and its dependents or a predicate and its arguments, while our proposed general framework considers all word pairs in a sentence. Our syntax-based reordering model exploits phrasal cohesion in translation (Fox, 2002) by modeling the reordering of sibling constituents in the source parse tree, which is similar to the recent work of Yang et al. (2012). However, the latter focuses on finding the optimal reordering of sibling constituents before MT decoding, while our proposed model generates reordering features that are used together with other MT features to determine the optimal reordering during MT decoding.

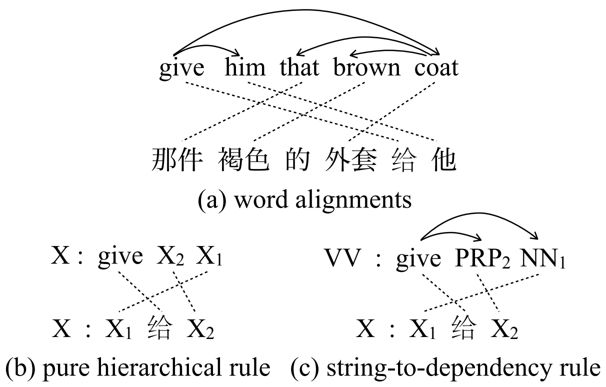
3 String-to-Dependency Translation

Our baseline translation system is based on a string-to-dependency translation model similar to the implementation in (Shen et al., 2008). It is an extension of the hierarchical translation model of Chiang et al. (2006) that requires the target side of a phrase pair to have a well-formed dependency structure, defined as either of the two types:

- *fixed structure*: a single rooted dependency sub-tree with each child being a complete constituent. In this case, the phrase has a unique head word inside the phrase, i.e., the root of the dependency sub-tree. Each dependent of the head word, together with all of its descendants, is either completely inside the phrase or completely outside the phrase. For example, the phrase *give him* in Figure 1 (a) has a fixed dependency structure with head word *give*.
- *floating structure*: a sequence of siblings with each being a complete constituent. In this case, the phrase is composed of a sequence of sibling constituents whose common parent is outside the phrase. For example, the phrase *him that brown coat* in Figure 1 is a floating structure whose common parent *give* is not in the phrase.

Requiring the target side to have a well-formed dependency structure is less restrictive than requiring it to be a syntactic constituent, allowing more translation rules to be extracted. However, it still results in fewer rules than pure hierarchical translation models and might hurt MT performance. The well-formed dependency structure on the target side makes it possible to introduce syntax features during decoding. Shen et al. (2008) obtained significant improvements from including a dependency language model score in decoding, outweighing the negative effect of the dependency constraint. Shen et al. (2009) proposed an approach to label each nonterminal, which can be either on the left-hand-side (LHS) or the right-hand-side (RHS) of the rule, with the head POS tag of the underlying target phrase if it has a fixed dependency structure[1], and measure the mismatches between nonterminal labels when a RHS nonterminal of a rule is substantiated with the LHS nonterminal of another rule during decoding. This also resulted in further improvements in MT performance. Figure 1 (c) shows an example string-to-dependency translation rule in our baseline system.

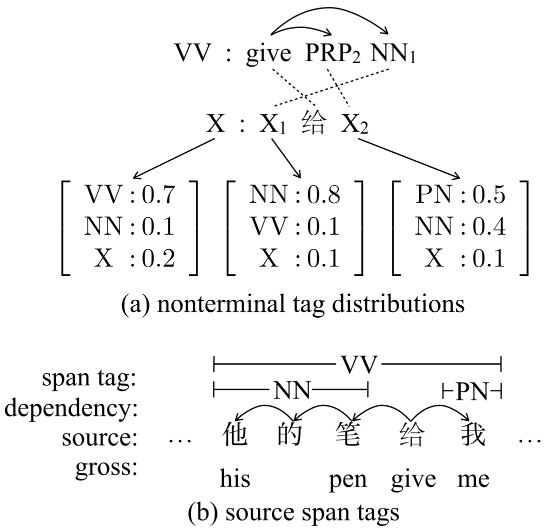
[1] Nonterminals corresponding to floating structures keep their default label "X" as experiments show that it is not beneficial to label them differently.



(a) word alignments

(b) pure hierarchical rule (c) string-to-dependency rule

Figure 1: An example of extracting a string-to-dependency translation rule from word alignments. The nonterminals on the target side of the hierarchical rule (b) all correspond to fixed dependency structures and so they are labeled by the respective head tag in the string-to-dependency rule (c).



(a) nonterminal tag distributions

(b) source span tags

Figure 2: Example distribution of tags for nonterminals on the source side (a) and example tags for source spans (b)

4 Factored Syntactic Constraints

Although the string-to-dependency formulation helps to improve the grammaticality of translations, it lacks the ability to incorporate source syntax into the translation process. We next describe a factored approach to address this problem by utilizing source syntax through two models: one that introduces syntactic awareness to translation rules themselves, and another that focuses on reordering based on the syntactic analysis of the source.

4.1 Syntax Mismatch Model

A straightforward method to introduce awareness of source syntax to translation rules is to apply the same well-formed dependency constraint and head POS annotation on the target side of string-to-dependency translation rules to the source side. However, as discussed earlier, this would significantly reduce the number of rules that can be extracted, exacerbate data sparsity, and cause other problems, especially given that the target side is already constrained by the dependency requirement.

A relaxed method is to bypass the dependency constraint and only annotate source nonterminals whose underlying phrase is a fixed dependency structure with the head POS tag of the phrase. This method would still extract all of the rules that can be extracted from the baseline string-to-dependency translation model, but the extra annotation on nonterminals can split a rule into multiple rules, with the only difference being the nonterminal labels on the source side. Unfortunately, our experiments have shown that even this moderate annotation results in significantly lower translation quality due to the fragmentation of translation rules, and the increased derivational ambiguity. We have also tried to include some source tag mismatch features (with details described later) to measure the syntactic compatibility between the nonterminal labels of a translation rule and the corresponding tags of source spans. This improves translation accuracy, but not enough to compensate for the performance drop caused by annotating source nonterminals.

Our proposed method introduces syntax to translation rules without sacrificing performance. Instead of imposing dependency constraints or explicitly annotating source nonterminals, we keep the original string-to-dependency translation rules intact and associate each nonterminal on the source side with a distribution of tags. The tags are determined based on the dependency structure of training samples. If the underlying source phrase of a nonterminal is a fixed dependency structure in a training sample, we use the head POS tag of the phrase as the tag. Otherwise, we use the default tag “X” to denote float-

Feature	**Condition**	**Value**
f_1	$t_s = \mathrm{X}$	$\mathrm{P}(t_r = \mathrm{X})$
f_2	$t_s = \mathrm{X}$	$\mathrm{P}(t_r \neq \mathrm{X})$
f_3	$t_s \neq \mathrm{X}$	$\mathrm{P}(t_r = \mathrm{X})$
f_4	$t_s \neq \mathrm{X}$	$\mathrm{P}(t_r = t_s)$
f_5	$t_s \neq \mathrm{X}$	$\mathrm{P}(t_r \neq \mathrm{X}, t_r \neq t_s)$

Table 1: Source tag mismatch features. The default value of each feature is zero if the source span tag t_s does not match the condition

ing structures and dependency structures that are not well formed. As a result, we still extract the same set of rules as in the baseline string-to-dependency translation model, and also obtain a distribution of tags for each nonterminal. Figure 2 (a) illustrates the example tag distributions of a string-to-dependency translation rule. The tag distributions provide an approximation of the source syntax of the training data from which the translation rules are extracted. They are used to measure the syntactic compatibility between a translation rule and the source spans it is applied to. At decoding time, we parse the source sentence and assign each span a tag in the same way as it is done during rule extraction, as shown in the example in Figure 2 (b). When a translation rule is used to expand a derivation, for each nonterminal (which can be on the LHS or RHS) on the source side of the rule, five source tag mismatch features are computed based on the distribution of tags $\mathrm{P}(t_r)$ on the rule nonterminal, and the tag t_s on the corresponding source span. The features are defined in Table 1. We use soft features instead of hard syntactic constraints, and allow the tuning process to choose the appropriate weight for each feature. As shown in Section 5, these source syntax mismatch features help to improve the baseline system.

4.2 Syntax-based Reordering Model

Most previous research on reordering models has focused on improving word reordering for statistical phrase-based translation systems (e.g., (Collins et al., 2005; Al-Onaizan and Papineni, 2006; Tromble and Eisner, 2009)). There has been less work on improving the reordering of hierarchical phrase-based translation systems (see (Xu et al., 2009; Gao et al., 2011; Xiong et al., 2012) for a few exceptions), except through explicit syntactic annotation of translation rules. It is generally assumed that hierarchical models are inherently capable of handling both local and non-local reorderings. However, many hierarchical translation rules are noisy and have limited context, and so may not be able to produce translations in the right order.

We propose a general framework that incorporates external reordering information into the decoding process of hierarchical translation models. To simplify the presentation, we make the assumption that every source word translates to one or more target words, and that the translations for a pair of source words is either straight or inverted. We discuss the general case later. Given a sentence $w_1, \cdots, w_n$, suppose we have a separate reordering model that predicts $\mathrm{P}_{\text{order}}(o_{ij})$, the probability distribution of ordering $o_{ij} \in \{\text{straight}, \text{inverted}\}$ between the translations of any source word pair (w_i, w_j). We can measure the goodness of a given hypothesis h with respect to the ordering predicted by the reordering model as the sum of log probabilities[2] for ordering each pair of source words, as defined in Equation 1:

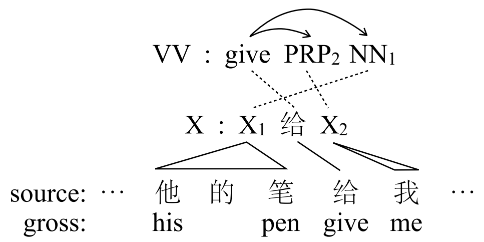
$$f_{\text{order}}(h) = \sum_{1 \leq i < j \leq n} \log \mathrm{P}_{\text{order}}(o_{ij} = o_{ij}^{h}) \qquad (1)$$

where o_{ij}^{h} is the ordering between the translations of source word pair (w_i, w_j) in hypothesis h. The reordering score $f_{\text{order}}(h)$ can be computed efficiently through recursion during hierarchical decoding as follows:

- Base case: for phrasal (i.e. non-hierarchical) rules, the ordering of translations for any word pair covered by the source phrase can be determined based on the word alignment of the rule. The value of the reordering score can be simply computed according to Equation 1.
- Recursive case: when a hierarchical rule is used to expand a partial derivation, two types of word pairs are encountered: a) word pairs that are covered exclusively by one of the nonterminals on the RHS of the rule, and b) other

[2] In practice, the log probability is thresholded to avoid negative infinity, which would otherwise result in a hard constraint.

(a)



(b)

word pair	translation order
(他, 给)	inverted
(他, 我)	inverted
(的, 给)	inverted
(的, 我)	inverted
(笔, 给)	inverted
(笔, 我)	inverted
(给, 我)	straight
(他, 的)	previously considered
(他, 笔)	previously considered
(的, 笔)	previously considered

Figure 3: An example rule application (a) with the translation order of new source word pairs covered by the rule shown in (b). The translation order of word pairs covered by X_1 is previously considered and is thus not shown.

word pairs. The reordering scores of the former would be already computed in previous rule applications, and can simply be retrieved from the partial derivation. Word pairs of the latter case are new word pairs introduced by the hierarchical rule, and their ordering can be determined based on the alignment of the hierarchical rule. The value of the reordering score of the new derivation is the sum of the reordering scores retrieved from the partial derivations for the nonterminals and the reordering scores of the new word pairs.

Figure 3 shows an example of determining the ordering of translations when applying a string-to-dependency rule. The alignment in the translation rule is able to fully determine the translation order for all new word pairs introduced by the rule. For example, “笔/pen” is covered by X_1 in the rule and the translation order for X_1 and “给/give” is inverted on the target side. Since “笔/pen” is translated together with other words covered by X_1 as a group, we can determine that the translation order between the source word pair “笔/pen” and “给/give” is also inverted on the target side. The words “他/his”, “的”, “笔 /pen” are all covered by the same nonterminal X_1 and thus their pairwise reordering scores have already been considered in previous rule applications.

Reordering features
The syntactic production rule
The syntactic labels of the nodes in the context
The head POS tags of the nodes in the context
The dep. labels of the nodes in the context
The seq. of dep. labels connecting the two nodes
The length of the nodes in the context

Table 2: Features in the reordering model

In practice, not all source words in a translation rule are translated to a target word; sometimes there is no clear ordering between the translations of two source words. In such cases we use a binary discount feature instead of the reordering feature.

This reordering framework relies on an external model to provide the ordering probability distribution of source word pairs. In this paper, we investigate a simple maximum-entropy reordering model based on the syntactic parse tree of the source sentence. This allows us to take advantage of the source syntax to improve reordering without using syntactic annotations in translation rules. The syntax-based reordering model attempts to predict the reordering probability of a pair of sibling constituents in the source parse tree, building on the fact that syntactic phrases tend to move in a group during translation (Fox, 2002). The reordering model is trained on a word-aligned corpus. For each pair of sibling constituents in the source parse tree, we determine the translation order on the target side based on word alignments. If there is a clear ordering[3], i.e., either straight or inverted, on the target side, we include the context of the constituent pair and its translation order as a sample for training or evaluating the maximum entropy reordering model. Table 2 lists the features of the reordering model.

The ordering distributions of source word pairs are determined based on the ordering distributions of sibling constituent pairs. For each pair of sib-

[3] If the translations overlap with other, the non-overlapping parts are used to determine the translation order.

ling constituents[4] in the parse tree of a source sentence, we compute its distribution of translation order using the reordering model. The distribution is shared among all word pairs covered by the respective constituents, which guarantees that the ordering distribution of any source word pair is computed exactly once. The ordering distributions of source word pairs are then used through the general reordering framework in the decoder to guide the decoding process.

5 Experiments

5.1 Experimental Setup

Our main experiments use the Chinese-English parallel training data and development sets released by the LDC, and made available to the DARPA GALE and BOLT programs. We train the translation model on 100 million words of parallel data. We use a 8 billion words of English monolingual data to train two language models: a trigram language model used in chart decoding, and a 5-gram language model used in n-best rescoring. The systems are tuned and evaluated on a mixture of newswire and web forum text from the development sets available for the DARPA GALE and BOLT programs, with up to 4 independent references for each source sentence. We also evaluate our final systems on both newswire and web text from the NIST MT06 and MT08 evaluations using an experimental setup compatible with the NIST MT12 Chinese-English constrained track. In this setup, the translation and language models are trained on 35 million words of parallel data and 3.8 billion words of English monolingual data, respectively. The systems are tuned on the MT02-05 development sets. All systems are tuned and evaluated on IBM BLEU (Papineni et al., 2002). The baseline string-to-dependency translation system uses more than 10 core features and a large number of sparse binary features similar to the method described in (Chiang et al., 2009). It achieves translation accuracies comparable to the top ranked systems in the NIST MT12 evaluation.

GIZA++ (Och and Ney, 2003) is used for automatic word alignment in all of the experiments. We use Charniak's parser (Charniak and Johnson, 2005) on the English side to obtain string-to-dependency translation rules, and use a latent variable PCFG parser (Huang and Harper, 2009) to parse the source side of the parallel training data as well as the test sentences for extracting syntax mismatch and reordering features. For both languages, dependency structures are read off constituency trees using manual head word percolation rules. We use a lexicon-based longest-match-first word segmenter to tokenize source Chinese sentences. Since the source tokenization used in our MT system is different from the treebank tokenization used to train the Chinese parser, the source sentences are first tokenized using the treebank-trained Stanford Chinese segmenter (Tseng et al., 2005), then parsed with the Chinese parser, and finally projected to MT tokenization based on the character alignment between the tokens. The syntax-based reordering model is trained on a set of Chinese-English manual word alignment corpora released by the LDC[5].

5.2 Syntax Mismatch Model

We first conduct experiments on the GALE/BOLT data sets to evaluate different strategies of incorporating source syntax into string-to-dependency translation rules. As mentioned in Section 4.1, constraining the source side of translation rules to only well-formed dependency structures is too restrictive given that our baseline system already has dependency constraint on the target side. We evaluate the relaxed method that only annotates source nonterminals with the head POS tag of the underlying phrase if the phrase is a fixed dependency structure. As shown in Table 3, nonterminal annotation results in a big drop in performance, decreasing the BLEU score of the baseline from 27.82 to 25.54. This suggests that it is undesirable to further fragment the translation rules. Introducing the syntax mismatch features described in Section 4.1 helps to improve

[4] Note that the constituent pairs used to train the reordering model are filtered to only contain these with clear ordering on the target side, while no such pre-filtering is applied to constituent pairs when applying the reordering model in translation. We leave it to future work to address this mismatch problem.

[5] The alignment corpora are LDC2012E24, LDC2012E72, LDC2012E95, and LDC2013E02. The reordering model can also be trained on automatically aligned data; however, our experiments show that using manual alignments results in a better accuracy for the reordering model itself and more improvements for the MT system.

	BLEU
baseline	27.82
+ tag annotation only	25.54
+ tag annotation, mismatch feat.	25.90
+ tag distribution, mismatch feat.	28.23

Table 3: Effects of tag annotation, tag distribution, and syntax mismatch features on MT performance on the GALE/BOLT data set.

BLEU from 25.54 to 25.90. This improvement is not large enough to compensate for the performance drop caused by annotating the nonterminals.

Our proposed approach, on the other hand, does not modify the translation rules in the baseline system, but only associates each nonterminal with a distribution of tags. For that reason, it does not suffer from the aforementioned problem. It achieves exactly the same performance as the baseline system if no source syntactic constraints are imposed during decoding. When the source syntax mismatch features are used, the proposed approach is able to achieve a gain of 0.41 in BLEU over the baseline system. Table 4 lists the learned weights of the syntax mismatch features after MT tuning. The negative weights of f_1 and f_2 mean that the MT system penalizes source spans that do not have a fixed dependency structure, and it assigns a higher penalty to rules whose nonterminals have a high probability of being extracted from source phrases that do not have a fixed dependency structure. When the source span has a fixed dependency structure, the MT system prefers translation rules that have a high probability of matching the tag on the source span (feature f_4) over the ones that do not match (features f_3 and f_5). This result is consistent with our expectations of the syntax mismatch features.

Feature	**Description**	**Weight**
f_1	$t_s = \mathrm{X}$, $t_r = \mathrm{X}$	–1.543
f_2	$t_s = \mathrm{X}$, $t_r \neq \mathrm{X}$	–0.676
f_3	$t_s \neq \mathrm{X}$, $t_r = \mathrm{X}$	0.380
f_4	$t_s \neq \mathrm{X}$, $t_r \neq \mathrm{X}$, $t_r = t_s$	1.677
f_5	$t_s \neq \mathrm{X}$, $t_r \neq \mathrm{X}$, $t_r \neq t_s$	0.232

Table 4: Learned syntax mismatch feature weight

5.3 Syntax-based Reordering Model

Before evaluating the syntax-based reordering model, we would like to establish the upper bound improvement that could be achieved using the general reordering framework for hierarchical translation models. Towards that goal, we conduct an oracle experiment on the GALE/BOLT development set that uses the oracle translation order from the reference as the external reordering model. For each source sentence in the development set, we pair it with the first reference translation (out of up to 4 independent translations). We then add the sentence pairs from the development set to the parallel training data and run GIZA++ to obtain word alignments. We consider the GIZA++ word alignments for the development set to be all correct, and use it to determine the oracle order in the reference translation. For the ordering distribution, we set the log probability of the reference translation order to 0 and the reverse order to -1 to avoid negative infinity. As shown in Table 5, the system tuned and evaluated with the oracle reordering model significantly outperforms the baseline by a large margin of 2.32 in BLEU on the GALE/BOLT test set. This suggests that there is room for potential improvement by using a fairly trained reordering model.

	BLEU
baseline	27.82
+ oracle reorder	30.14
+ syntax reorder	28.40

Table 5: Effects of external reordering features on MT performance on the GALE/BOLT test set.

We next evaluate the syntax-based reordering model. We train the model on manually aligned Chinese-English corpora. Since the tokenization used in the manual alignment corpora is different from the tokenization used in our MT system, the manual alignment is projected to the MT tokenization based on the character alignment between the tokens. Some extraneously tagged alignment links in the manual alignment corpora are not useful for machine translation and are thus removed before projecting the alignment. As described in Section 4.2, the syntax-based reordering method models the translation order of sibling constituent pairs in the source parse tree. As a result of strong phrasal cohesion (Fox, 2002), we find that 94% of constituent pairs have a clear ordering on the target side. We only retain these constituent pairs for training and evaluating the reordering model. In order to evaluate the accuracy of the maximum entropy reordering model, we divide the manual alignment corpora into 2/3 for training and 1/3 for evaluation. A baseline that only chooses the majority order (i.e. straight) has an accuracy of 69%, while the syntax-based reordering model improves the accuracy to 79%.

The final reordering model used in MT is trained on all of the samples extracted from the manual alignment corpora. As shown in Table 5, the syntax-based reordering feature improves the baseline by 0.58 in BLEU, which is a good improvement given our strong baseline. Table 6 lists the number of shifting errors in TER measurement (Snover et al., 2006) of various systems on the GALE/BOLT test set. The syntax-based reordering model achieves a 6.1% reduction in the number of shifting errors in the baseline system, and its combination with the syntax mismatch model achieves an additional reduction of 0.6%. This suggests that the proposed method helps to improve word reordering in translation.

	Shifting errors
baseline	3205
+ syntax mismatch	3089
+ syntax reorder	3010
+ syntax mismatch and reorder	2990

Table 6: Number of shifting errors in TER measurement of multiple systems on the GALE/BOLT test set

5.4 Final Results

Table 7 shows the final results on the GALE/BOLT test set, as well as the NIST MT06 and MT08 test sets. Both the syntax mismatch and the syntax-based reordering features improve the baseline system, resulting in moderate to significant gains in all of the five test sets. The two features are complementary to each other and their combination results in better improvement in four out of the five test sets compared to adding them separately. In three out of the five test sets, the improvement from the combination of the two features is statistically significant at the 95% confidence level over the baseline, with the largest absolute improvement of 1.43 in BLEU obtained on MT08 web.

6 Conclusion

In this paper, We have discussed problems resulting from explicitly decorating translation rules with syntactic annotations. We presented a factored approach to incorporate soft source syntax mismatch and reordering constraints to hierarchical machine translation, and showed how our models avoid the pitfalls of the explicit decoration approach. Experiments on Chinese-English translation show that the proposed approach significantly improves a strong string-to-dependency translation baseline on multiple evaluation sets. There are many directions in which this work can be continued. The syntax mismatch model can be extended to dynamically adjust the translation distribution based on the syntactic compatibility between a translation rule and a source sentence. It also might be beneficial to look beyond syntactic constituent pairs when modeling reordering, given that phrasal cohesion does not always hold in translation. The general framework that uses an external reordering model in hierarchical models via features can also be naturally extended to use multiple reordering models.

Acknowledgments

This work was supported in part by DARPA/IPTO Contract No. HR0011-12-C-0014 under the BOLT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The authors would like to thank the anonymous reviewers for their helpful comments.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

	MT06 news	**MT06 web**	**MT08 news**	**MT08 web**	**GALE/BOLT**
baseline	43.76	36.13	40.52	27.78	27.82
+ syntax mismatch	43.89	36.72	40.82$^{\wedge}$	28.54$^{\overset{\wedge}{\wedge}}$	28.23$^{\overset{\wedge}{\wedge}}$
+ syntax reorder	44.01	36.40	41.23$^{\overset{\wedge}{\overset{\wedge}{\wedge}}}$	28.95$^{\overset{\wedge}{\overset{\wedge}{\wedge}}}$	28.40$^{\overset{\wedge}{\overset{\wedge}{\wedge}}}$
+ syntax mismatch and reorder	44.28$^{\wedge}$	36.43$^{\wedge}$	41.14$^{\overset{\wedge}{\wedge}}$	29.21$^{\overset{\wedge}{\overset{\wedge}{\wedge}}}$	28.62$^{\overset{\wedge}{\overset{\wedge}{\wedge}}}$
improvement over baseline	+0.52	+0.30	+0.62	+1.43	+0.8

Table 7: Results on Chinese-English MT. The symbols $\wedge$, $\overset{\wedge}{\wedge}$, and $\overset{\wedge}{\overset{\wedge}{\wedge}}$ indicate that the system is better than the baseline at the 85%, 95%, and 99% confidence levels, respectively, as defined in (Koehn, 2004).

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic dialects. In *Conference of the European Chapter of the Association for Computational Linguistics*.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2004. What's in a translation rule. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Yang Gao, Philipp Koehn, and Alexandra Birch. 2011. Soft dependency constraints for reordering in hierarchical phrase-based translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the International Conference on Computational Linguistics*.

Greg Hanneman and Alon Lavie. 2013. Improving syntax-augmented machine translation by coarsening the label set. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Zhongqiang Huang and Mary Harper. 2009. Self-training PCFG grammars with latent annotations across languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A syntax-directed translator with extended domain of locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*.

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

Mitesh M. Khapra, Ananthakrishnan Ramanathan, and Karthik Visweswariah. 2013. Improving reordering performance using higher order and structural features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

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

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*.

Philipp Koehn. 2004. Pharaoh: A bean search decoder for phrase-based statistical machine translation models. In *Proceedings of the Conference of Association for Machine Translation in the Americas*.

Junhui Li, Philip Resnik, and Hal Daume. 2013. Modeling syntactic and semantic structures in hierarchical phrase-based translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computional 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 Annual Meeting on Association for Computational Linguistics*.

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 the Annual Meeting of the 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 Conference on Empirical Methods in Natural Language Processing*.

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*.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Proceedings of the SIGHAN Workshop on Chinese Language Processing*.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2009. Preference grammars: softening syntactic constraints to improve statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the International Conference on Computational Linguistics*.

Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceeding of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Nan Yang, Mu Li, Dongdong Zhang, and Nenghai Yu. 2012. A ranking-based approach to word reordering for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*.