Topological Ordering of Function Words in Hierarchical Phrase-based Translation

Hendra Setiawan[1] and **Min-Yen Kan**[2] and **Haizhou Li**[3] and **Philip Resnik**[1]
[1]University of Maryland Institute for Advanced Computer Studies
[2]School of Computing, National University of Singapore
[3]Human Language Technology, Institute for Infocomm Research, Singapore
{hendra,resnik}@umiacs.umd.edu,
kanmy@comp.nus.edu.sg, hli@i2r.a-star.edu.sg

Abstract

Hierarchical phrase-based models are attractive because they provide a consistent framework within which to characterize both local and long-distance reorderings, but they also make it difficult to distinguish many implausible reorderings from those that are linguistically plausible. Rather than appealing to annotation-driven syntactic modeling, we address this problem by observing the influential role of function words in determining syntactic structure, and introducing soft constraints on function word relationships as part of a standard log-linear hierarchical phrase-based model. Experimentation on Chinese-English and Arabic-English translation demonstrates that the approach yields significant gains in performance.

1 Introduction

Hierarchical phrase-based models (Chiang, 2005; Chiang, 2007) offer a number of attractive benefits in statistical machine translation (SMT), while maintaining the strengths of phrase-based systems (Koehn et al., 2003). The most important of these is the ability to model long-distance reordering efficiently. To model such a reordering, a hierarchical phrase-based system demands no additional parameters, since long and short distance reorderings are modeled identically using synchronous context free grammar (SCFG) rules. The same rule, depending on its topological ordering – i.e. its position in the hierarchical structure – can affect both short and long spans of text. Interestingly, hierarchical phrase-based models provide this benefit without making any linguistic commitments beyond the structure of the model.

However, the system's lack of linguistic commitment is also responsible for one of its greatest drawbacks. In the absence of linguistic knowledge, the system models linguistic structure using an SCFG that contains only one type of nonterminal symbol[1]. As a result, the system is susceptible to the *overgeneration* problem: the grammar may suggest more reordering choices than appropriate, and many of those choices lead to ungrammatical translations.

Chiang (2005) hypothesized that incorrect reordering choices would often correspond to hierarchical phrases that violate syntactic boundaries in the source language, and he explored the use of a "constituent feature" intended to reward the application of hierarchical phrases which respect source language syntactic categories. Although this did not yield significant improvements, Marton and Resnik (2008) and Chiang et al. (2008) extended this approach by introducing soft syntactic constraints similar to the constituent feature, but more fine-grained and sensitive to distinctions among syntactic categories; these led to substantial improvements in performance. Zollman et al. (2006) took a complementary approach, constraining the application of hierarchical rules to respect syntactic boundaries in the target language syntax. Whether the focus is on constraints from the source language or the target language, the main ingredient in both previous approaches is the idea of constraining the spans of hierarchical phrases to respect syntactic boundaries.

In this paper, we pursue a different approach to improving reordering choices in a hierarchical phrase-based model. Instead of biasing the model toward hierarchical phrases whose spans respect syntactic boundaries, we focus on the topological ordering of phrases in the hierarchical structure. We conjecture that since incorrect reordering choices correspond to incorrect topological orderings, boosting the probability of correct topo-

[1] In practice, one additional nonterminal symbol is used in "glue rules". This is not relevant in the present discussion.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 324–332,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

logical ordering choices should improve the system. Although related to previous proposals (correct topological orderings lead to correct spans and vice versa), our proposal incorporates broader context and is structurally more aware, since we look at the topological ordering of a phrase relative to other phrases, rather than modeling additional properties of a phrase in isolation. In addition, our proposal requires no monolingual parsing or linguistically informed syntactic modeling for either the source or target language.

The key to our approach is the observation that we can approximate the topological ordering of hierarchical phrases via the topological ordering of function words. We introduce a statistical reordering model that we call the *pairwise dominance model*, which characterizes reorderings of phrases around a pair of function words. In modeling function words, our model can be viewed as a successor to the function words-centric reordering model (Setiawan et al., 2007), expanding on the previous approach by modeling pairs of function words rather than individual function words in isolation.

The rest of the paper is organized as follows. In Section 2, we briefly review hierarchical phrase-based models. In Section 3, we first describe the overgeneration problem in more detail with a concrete example, and then motivate our idea of using the topological ordering of function words to address the problem. In Section 4, we develop our idea by introducing the pairwise dominance model, expressing function word relationships in terms of what we call the the *dominance* predicate. In Section 5, we describe an algorithm to estimate the parameters of the dominance predicate from parallel text. In Sections 6 and 7, we describe our experiments, and in Section 8, we analyze the output of our system and lay out a possible future direction. Section 9 discusses the relation of our approach to prior work and Section 10 wraps up with our conclusions.

2 Hierarchical Phrase-based System

Formally, a hierarchical phrase-based SMT system is based on a weighted synchronous context free grammar (SCFG) with one type of nonterminal symbol. Synchronous rules in hierarchical phrase-based models take the following form:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \qquad (1)$$

where X is the nonterminal symbol and γ and α are strings that contain the combination of lexical items and nonterminals in the source and target languages, respectively. The $\sim$ symbol indicates that nonterminals in γ and α are synchronized through co-indexation; i.e., nonterminals with the same index are aligned. Nonterminal correspondences are strictly one-to-one, and in practice the number of nonterminals on the right hand side is constrained to at most two, which must be separated by lexical items.

Each rule is associated with a score that is computed via the following log linear formula:

$$w(X \rightarrow \langle \gamma, \alpha, \sim \rangle) = \prod_i f_i^{\lambda_i} \qquad (2)$$

where f_i is a feature describing one particular aspect of the rule and λ_i is the corresponding weight of that feature. Given $\tilde{e}$ and $\tilde{f}$ as the source and target phrases associated with the rule, typical features used are rule's translation probability $P_{trans}(\tilde{f}|\tilde{e})$ and its inverse $P_{trans}(\tilde{e}|\tilde{f})$, the lexical probability $P_{lex}(\tilde{f}|\tilde{e})$ and its inverse $P_{lex}(\tilde{e}|\tilde{f})$. Systems generally also employ a word penalty, a phrase penalty, and target language model feature. (See (Chiang, 2005) for more detailed discussion.) Our pairwise dominance model will be expressed as an additional rule-level feature in the model.

Translation of a source sentence e using hierarchical phrase-based models is formulated as a search for the most probable derivation D^* whose source side is equal to e:

$$D^* = \text{argmax } P(D), \text{where source}(D)\text{=}e.$$

$D = X^i, i \in 1...|D|$ is a set of rules following a certain topological ordering, indicated here by the use of the superscript.

3 Overgeneration and Topological Ordering of Function Words

The use of only one type of nonterminal allows a flexible permutation of the topological ordering of the same set of rules, resulting in a huge number of possible derivations from a given source sentence. In that respect, the overgeneration problem is not new to SMT: Bracketing Transduction Grammar (BTG) (Wu, 1997) uses a single type of nonterminal and is subject to overgeneration problems, as well.[2]

[2] Note, however, that overgeneration in BTG can be viewed as a feature, not a bug, since the formalism was origi-

The problem may be less severe in hierarchical phrase-based MT than in BTG, since lexical items on the rules' right hand sides often limit the span of nonterminals. Nonetheless overgeneration of reorderings is still problematic, as we illustrate using the hypothetical Chinese-to-English example in Fig. 1.

Suppose we want to translate the Chinese sentence in Fig. 1 into English using the following set of rules:

1. $X_a \rightarrow \langle$ 电脑 和 X_1, computers and $X_1\rangle$
2. $X_b \rightarrow \langle X_1$ 是 X_2, X_1 are $X_2\rangle$
3. $X_c \rightarrow \langle$ 手机 , cell phones $\rangle$
4. $X_d \rightarrow \langle X_1$ 的 发明 , inventions of $X_1\rangle$
5. $X_e \rightarrow \langle$ 上个世纪 , the last century $\rangle$

Co-indexation of nonterminals on the right hand side is indicated by subscripts, and for our examples the label of the nonterminal on the left hand side is used as the rule's unique identifier. To correctly translate the sentence, a hierarchical phrase-based system needs to model the subject noun phrase, object noun phrase and copula constructions; these are captured by rules X_a, X_d and X_b respectively, so this set of rules represents a hierarchical phrase-based system that can be used to correctly translate the Chinese sentence. Note that the Chinese word order is correctly preserved in the subject (X_a) as well as copula constructions (X_b), and correctly inverted in the object construction (X_d).

However, although it can generate the correct translation in Fig. 2, the grammar has no mechanism to prevent the generation of an incorrect translation like the one illustrated in Fig. 3. If we contrast the topological ordering of the rules in Fig. 2 and Fig. 3, we observe that the difference is small but quite significant. Using precede symbol ($\prec$) to indicate the first operand immediately dominates the second operand in the hierarchical structure, the topological orderings in Fig. 2 and Fig. 3 are $X_a \prec X_b \prec X_c \prec X_d \prec X_e$ and $X_d \prec X_a \prec X_b \prec X_c \prec X_e$, respectively. The only difference is the topological ordering of X_d: in Fig. 2, it appears below most of the other hierarchical phrases, while in Fig. 3, it appears above all the other hierarchical phrases.

Modeling the topological ordering of hierarchical phrases is computationally prohibitive, since there are literally millions of hierarchical rules in the system's automatically-learned grammar and millions of possible ways to order their application. To avoid this computational problem and still model the topological ordering, we propose to use the topological ordering of function words as a practical approximation. This is motivated by the fact that function words tend to carry crucial syntactic information in sentences, serving as the "glue" for content-bearing phrases. Moreover, the positional relationships between function words and content phrases tends to be fixed (e.g., in English, prepositions invariably precede their object noun phrase), at least for the languages we have worked with thus far.

In the Chinese sentence above, there are three function words involved: the conjunction 和 (and), the copula 是 (are), and the noun phrase marker 的 (of).[3] Using the function words as approximate representations of the rules in which they appear, the topological ordering of hierarchical phrases in Fig. 2 is 和(and) $\prec$ 是(are) $\prec$ 的(of), while that in Fig. 3 is 的(of) $\prec$ 和(and) $\prec$ 是(are).[4] We can distinguish the correct and incorrect reordering choices by looking at this simple information. In the correct reordering choice, 的(of) appears at the lower level of the hierarchy while in the incorrect one, 的(of) appears at the highest level of the hierarchy.

4 Pairwise Dominance Model

Our example suggests that we may be able to improve the translation model's sensitivity to correct versus incorrect reordering choices by modeling the topological ordering of function words. We do so by introducing a predicate capturing the *dominance* relationship in a derivation between pairs of neighboring function words.[5]

Let us define a predicate $d(Y', Y'')$ that takes two function words as input and outputs one of

nally introduced for bilingual analysis rather than generation of translations.

[3] We use the term "noun phrase marker" here in a general sense, meaning that in this example it helps tell us that the phrase is part of an NP, not as a technical linguistic term. It serves in other grammatical roles, as well. Disambiguating the syntactic roles of function words might be a particularly useful thing to do in the model we are proposing; this is a question for future research.

[4] Note that for expository purposes, we designed our simple grammar to ensure that these function words appear in separate rules.

[5] Two function words are considered neighbors iff no other function word appears between them in the source sentence.

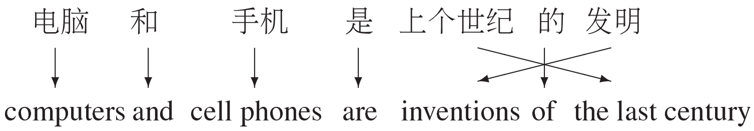

Figure 1: A running example of Chinese-to-English translation.

$X_a \Rightarrow \langle$ 电脑 和 X_b, computers and $X_b\rangle$
$\Rightarrow \langle$ 电脑 和 X_c 是 X_d, computers and X_c are $X_d\rangle$
$\Rightarrow \langle$ 电脑 和 手机 是 X_d, computers and cell phones are $X_d\rangle$
$\Rightarrow \langle$ 电脑 和 手机 是 X_e 的 发明 , computers and cell phones are inventions of $X_e\rangle$
$\Rightarrow \langle$ 电脑 和 手机 是 上个世纪 的 发明 , computers and cell phones are inventions of the last century$\rangle$

Figure 2: The derivation that leads to the correct translation

$X_d \Rightarrow \langle X_a$ 的 发明 , inventions of $X_a\rangle$
$\Rightarrow \langle$电脑 和 X_b 的 发明 , inventions of computers and $X_b\rangle$
$\Rightarrow \langle$电脑 和 X_c 是 X_e 的 发明 , inventions of computers and X_c are $X_e\rangle$
$\Rightarrow \langle$电脑 和 手机 是 X_e 的 发明 , inventions of computers and cell phones are $X_e\rangle$
$\Rightarrow \langle$电脑 和 手机 是 上个世纪 的 发明 , inventions of computers and cell phones are the last century$\rangle$

Figure 3: The derivation that leads to the incorrect translation

four values: $\{$leftFirst, rightFirst, dontCare, neither$\}$, where Y' appears to the left of Y'' in the source sentence. The value leftFirst indicates that in the derivation's topological ordering, Y' precedes Y'' (i.e. Y' dominates Y'' in the hierarchical structure), while rightFirst indicates that Y'' dominates Y'. In Fig. 2, $d(Y', Y'') =$ leftFirst for $Y' =$ the copula 是 (are) and $Y'' =$ the noun phrase marker 的 (of).

The dontCare and neither values capture two additional relationships: dontCare indicates that the topological ordering of the function words is flexible, and neither indicates that the topological ordering of the function words is disjoint. The former is useful in cases where the hierarchical phrases suggest the same kind of reordering, and therefore restricting their topological ordering is not necessary. This is illustrated in Fig. 2 by the pair 和(and) and the copula 是(are), where putting either one above the other does not change the final word order. The latter is useful in cases where the two function words do not share a same parent.

Formally, this model requires several changes in the design of the hierarchical phrase-based system.

1. To facilitate topological ordering of function words, the hierarchical phrases must be subcategorized with function words. Taking X_b in Fig. 2 as a case in point, subcategorization using function words would yield:[6]

$$X_b(\text{是} \prec \text{的}) \rightarrow X_c \text{ 是 } X_d(\text{的}) \tag{3}$$

The subcategorization (indicated by the information in parentheses following the nonterminal) propagates the function word 是(are) of X_b to the higher level structure together with the function word 的(of) of X_d. This propagation process generalizes to other rules by maintaining the ordering of the function words according to their appearance in the source sentence. Note that the subcategorized nonterminals often resemble genuine syntactic categories, for instance X(的) can frequently be interpreted as a noun phrase.

2. To facilitate the computation of the dominance relationship, the coindexing in synchronized rules (indicated by the $\sim$ symbol in Eq. 1) must be expanded to include information not only about the nonterminal correspondences but also about the alignment of the lexical items. For example, adding lexical alignment information to rule X_d would yield:

$$X_d \rightarrow \langle X_1\text{的}_2\text{发明}_3,\ \text{inventions}_3\ \text{of}_2\ X_1\rangle \tag{4}$$

[6] The target language side is concealed for clarity.

> The computation of the dominance relationship using this alignment information will be discussed in detail in the next section.

Again taking X_b in Fig. 2 as a case in point, the dominance feature takes the following form:

$$f_{dom}(X_b) \approx dom(d(是, 的)|是, 的)) \quad (5)$$

$$dom(d(Y_L, Y_R)|Y_L, Y_R)) \quad (6)$$

where the probability of 是 $\prec$ 的 is estimated according to the probability of d(是, 的).

In practice, both 是(are) and 的(of) may appear together in one same rule. In such a case, a dominance score is not calculated since the topological ordering of the two function words is unambiguous. Hence, in our implementation, a dominance score is only calculated at the points where the topological ordering of the hierarchical phrases needs to be resolved, i.e. the two function words always come from two different hierarchical phrases.

5 Parameter Estimation

Learning the dominance model involves extracting *d* values for every pair of neighboring function words in the training bitext. Such statistics are not directly observable in parallel corpora, so estimation is needed. Our estimation method is based on two facts: (1) the topological ordering of hierarchical phrases is tightly coupled with the span of the hierarchical phrases, and (2) the span of a hierarchical phrase at a higher level is always a superset of the span of all other hierarchical phrases at the lower level of its substructure. Thus, to establish soft estimates of dominance counts, we utilize alignment information available in the rule together with the consistent alignment heuristic (Och and Ney, 2004) traditionally used to guess phrase alignments.

Specifically, we define the span of a function word as a maximal, consistent alignment in the source language that either starts from or ends with the function word. (Requiring that spans be maximal ensures their uniqueness.) We will refer to such spans as Maximal Consistent Alignments (MCA). Note that each function word has two such Maximal Consistent Alignments: one that ends with the function word (MCA$_R$)and another that starts from the function word (MCA$_L$).

Y'	Y''	left- First	right- First	dont- Care	nei- ther
和 (and)	是 (are)	0.11	0.16	**0.68**	0.05
是 (are)	的 (of)	**0.57**	0.15	0.06	0.22

Table 1: The distribution of the dominance values of the function words involved in Fig. 1. The value with the highest probability is in **bold**.

Given two function words Y' and Y'', with Y' preceding Y'', we define the value of *d* by examining the MCAs of the two function words.

$$d(Y', Y'') = \begin{cases} \textsf{leftFirst}, & Y' \notin \text{MCA}_R(Y'') \wedge Y'' \in \text{MCA}_L(Y') \\ \textsf{rightFirst}, & Y' \in \text{MCA}_R(Y'') \wedge Y'' \notin \text{MCA}_L(Y') \\ \textsf{dontCare}, & Y' \in \text{MCA}_R(Y'') \wedge Y'' \in \text{MCA}_L(Y') \\ \textsf{neither}, & Y' \notin \text{MCA}_R(Y'') \wedge Y'' \notin \text{MCA}_L(Y') \end{cases} \quad (6)$$

Fig. 4a illustrates the leftFirst dominance value where the intersection of the MCAs contains only the second function word (的(of)). Fig. 4b illustrates the dontCare value, where the intersection contains both function words. Similarly, rightFirst and neither are represented by an intersection that contains only Y', or by an empty intersection, respectively. Once all the *d* values are counted, the pairwise dominance model of neighboring function words can be estimated simply from counts using maximum likelihood. Table 1 illustrates estimated dominance values that correctly resolve the topological ordering for our running example.

6 Experimental Setup

We tested the effect of introducing the pairwise dominance model into hierarchical phrase-based translation on Chinese-to-English and Arabic-to-English translation tasks, thus studying its effect in two languages where the use of function words differs significantly. Following Setiawan et al. (2007), we identify function words as the N most frequent words in the corpus, rather than identifying them according to linguistic criteria; this approximation removes the need for any additional language-specific resources. We report results for $N = 32, 64, 128, 256, 512, 1024, 2048$.[7] For

[7] We observe that even $N = 2048$ represents less than 1.5% and 0.8% of the words in the Chinese and Arabic vocabularies, respectively. The validity of the frequency-based strategy, relative to linguistically-defined function words, is discussed in Section 8

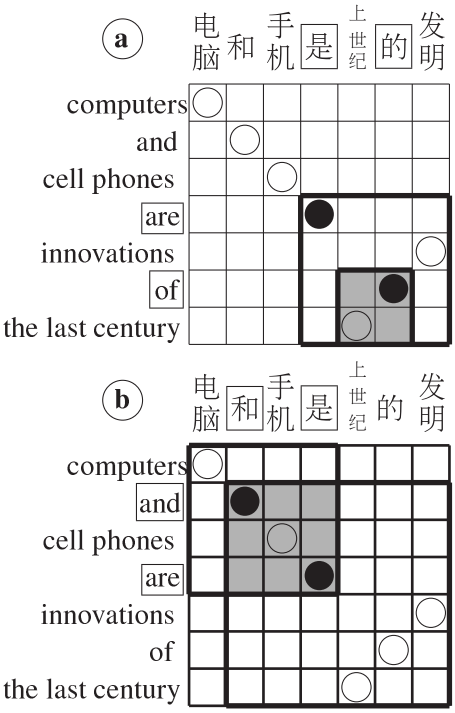

Figure 4: Illustrations for: a) the leftFirst value, and b) the dontCare value. Thickly bordered boxes are MCAs of the function words while solid circles are the alignment points of the function words. The gray boxes are the intersections of the two MCAs.

all experiments, we report performance using the BLEU score (Papineni et al., 2002), and we assess statistical significance using the standard bootstrapping approach introduced by (Koehn, 2004).

Chinese-to-English experiments. We trained the system on the NIST MT06 Eval corpus excluding the UN data (approximately 900K sentence pairs). For the language model, we used a 5-gram model with modified Kneser-Ney smoothing (Kneser and Ney, 1995) trained on the English side of our training data as well as portions of the Gigaword v2 English corpus. We used the NIST MT03 test set as the development set for optimizing interpolation weights using minimum error rate training (MERT; (Och and Ney, 2002)). We carried out evaluation of the systems on the NIST 2006 evaluation test (MT06) and the NIST 2008 evaluation test (MT08). We segmented Chinese as a preprocessing step using the Harbin segmenter (Zhao et al., 2001).

Arabic-to-English experiments. We trained the system on a subset of 950K sentence pairs from the NIST MT08 training data, selected by "subsampling" from the full training data using a method proposed by Kishore Papineni (personal communication). The subsampling algorithm selects sentence pairs from the training data in a way that seeks reasonable representation for all n-grams appearing in the test set. For the language model, we used a 5-gram model trained on the English portion of the whole training data plus portions of the Gigaword v2 corpus. We used the NIST MT03 test set as the development set for optimizing the interpolation weights using MERT. We carried out the evaluation of the systems on the NIST 2006 evaluation set (MT06) and the NIST 2008 evaluation set (MT08). Arabic source text was preprocessed by separating clitics, the definiteness marker, and the future tense marker from their stems.

7 Experimental Results

Chinese-to-English experiments. Table 2 summarizes the results of our Chinese-to-English experiments. These results confirm that the pairwise dominance model can significantly increase performance as measured by the BLEU score, with a consistent pattern of results across the MT06 and MT08 test sets. Modeling $N = 32$ drops the performance marginally below baseline, suggesting that perhaps there are not enough words for the pairwise dominance model to work with. Doubling the number of words ($N = 64$) produces a small gain, and defining the pairwise dominance model using $N = 128$ most frequent words produces a statistically significant 1-point gain over the baseline ($p < 0.01$). Larger values of N yield statistically significant performance above the baseline, but without further improvements over $N = 128$.

Arabic-to-English experiments. Table 3 summarizes the results of our Arabic-to-English experiments. This set of experiments shows a pattern consistent with what we observed in Chinese-to-English translation, again generally consistent across MT06 and MT08 test sets although modeling a small number of lexical items ($N = 32$) brings a marginal improvement over the baseline. In addition, we again find that the pairwise dominance model with $N = 128$ produces the most significant gain over the baseline in the MT06, although, interestingly, modeling a much larger number of lexical items ($N = 2048$) yields the strongest improvement for the MT08 test set.

	MT06	MT08
baseline	30.58	24.08
+$dom(N = 32)$	30.43	23.91
+$dom(N = 64)$	30.96	24.45
+$dom(N = 128)$	**31.59**	**24.91**
+$dom(N = 256)$	**31.24**	24.26
+$dom(N = 512)$	**31.33**	24.39
+$dom(N = 1024)$	**31.22**	**24.79**
+$dom(N = 2048)$	30.75	23.92

Table 2: Experimental results on Chinese-to-English translation with the pairwise dominance model (dom) of different N. The baseline (the first line) is the original hierarchical phrase-based system. Statistically significant results ($p < 0.01$) over the baseline are in **bold**.

	MT06	MT08
baseline	41.56	40.06
+$dom(N = 32)$	41.66	40.26
+$dom(N = 64)$	**42.03**	**40.73**
+$dom(N = 128)$	**42.66**	**41.08**
+$dom(N = 256)$	**42.28**	40.69
+$dom(N = 512)$	41.97	**40.95**
+$dom(N = 1024)$	42.05	40.55
+$dom(N = 2048)$	**42.48**	**41.47**

Table 3: Experimental results on Arabic-to-English translation with the pairwise dominance model (dom) of different N. The baseline (the first line) is the original hierarchical phrase-based system. Statistically significant results over the baseline ($p < 0.01$) are in **bold**.

8 Discussion and Future Work

The results in both sets of experiments show consistently that we have achieved a significant gains by modeling the topological ordering of function words. When we visually inspect and compare the outputs of our system with those of the baseline, we observe that improved BLEU score often corresponds to visible improvements in the subjective translation quality. For example, the translations for the Chinese sentence “军情$_1$ 观察$_2$:$_3$ 伊朗$_4$ 在$_5$ 美军$_6$ 空袭$_7$ 下$_8$ 能$_9$ 撑$_{10}$ 多$_{11}$ 久$_{12}$?$_{13}$”, taken from Chinese MT06 test set, are as follows (co-indexing subscripts represent reconstructed word alignments):

- baseline: “military$_1$ intelligence$_2$ under_observation$_8$ in$_5$ u.s.$_6$ air_raids$_7$:$_3$ iran$_4$ to$_9$ how$_{11}$ long$_{12}$?$_{13}$ ”
- *+dom(N=128)*: “ military$_1$ survey$_2$:$_3$ how$_{11}$ long$_{12}$ iran$_4$ under$_8$ air_strikes$_7$ of_the_u.s$_6$ can$_9$ hold_out$_{10}$?$_{13}$ ”

In addition to some lexical translation errors (e.g. 美军$_6$ should be translated to U.S. Army), the baseline system also makes mistakes in reordering. The most obvious, perhaps, is its failure to capture the wh-movement involving the interrogative word 多$_{11}$ (how); this should move to the beginning of the translated clause, consistent with English wh-fronting as opposed to Chinese wh in $situ$. The pairwise dominance model helps, since the dominance value between the interrogative word and its previous function word, the modal verb 能$_9$(can) in the baseline system's output, is neither, rather than rightFirst as in the better translation.

The fact that performance tends to be best using a frequency threshold of $N = 128$ strikes us as intuitively sensible, given what we know about word frequency rankings.[8] In English, for example, the most frequent 128 words include virtually all common conjunctions, determiners, prepositions, auxiliaries, and complementizers – the crucial elements of “syntactic glue” that characterize the types of linguistic phrases and the ordering relationships between them – and a very small proportion of content words. Using Adam Kilgarriff's lemmatized frequency list from the British National Corpus, http://www.kilgarriff.co.uk/bnc-readme.html, the most frequent 128 words in English are heavily dominated by determiners, “functional” adverbs like *not* and *when*, “particle” adverbs like *up*, prepositions, pronouns, and conjunctions, with some arguably “functional” auxiliary and light verbs like *be*, *have*, *do*, *give*, *make*, *take*. Content words are generally limited to a small number of frequent verbs like *think* and *want* and a very small handful of frequent nouns. In contrast, ranks 129-256 are heavily dominated by the traditional content-word categories, i.e. nouns, verbs, adjectives and adverbs, with a small number of left-over function words such as less frequent conjunctions *while*, *when*, and *although*.

Consistent with these observations for English, the empirical results for Chinese suggest that our

[8] In fact, we initially simply chose $N = 128$ for our experimentation, and then did runs with alternative N to confirm our intuitions.

approximation of function words using word frequency is reasonable. Using a list of approximately 900 linguistically identified function words in Chinese extracted from (Howard, 2002), we observe that that the performance drops when increasing N above 128 corresponds to a large increase in the number of non-function words used in the model. For example, with $N = 2048$, the proportion of non-function words is 88%, compared to 60% when $N = 128$.[9]

One natural extension of this work, therefore, would be to tighten up our characterization of function words, whether statistically, distributionally, or simply using manually created resources that exist for many languages. As a first step, we did a version of the Chinese-English experiment using the list of approximately 900 genuine function words, testing on the Chinese MT06 set. Perhaps surprisingly, translation performance, 30.90 BLEU, was around the level we obtained when using frequency to approximate function words at $N = 64$. However, we observe that many of the words in the linguistically motivated function word list are quite infrequent; this suggests that data sparseness may be an additional factor worth investigating.

Finally, although we believe there are strong motivations for focusing on the role of function words in reordering, there may well be value in extending the dominance model to include content categories. Verbs and many nouns have subcategorization properties that may influence phrase ordering, for example, and this may turn out to explain the increase in Arabic-English performance for $N = 2048$ using the MT08 test set. More generally, the approach we are taking can be viewed as a way of selectively lexicalizing the automatically extracted grammar, and there is a large range of potentially interesting choices in how such lexicalization could be done.

9 Related Work

In the introduction, we discussed Chiang's (2005) constituency feature, related ideas explored by Marton and Resnik (2008) and Chiang et al. (2008), and the target-side variation investigated by Zollman et al. (2006). These methods differ from each other mainly in terms of the specific linguistic knowledge being used and on which side the constraints are applied.

Shen et al. (2008) proposed to use linguistic knowledge expressed in terms of a dependency grammar, instead of a syntactic constituency grammar. Villar et al. (2008) attempted to use syntactic constituency on both the source and target languages in the same spirit as the constituency feature, along with some simple pattern-based heuristics – an approach also investigated by Iglesias et al. (2009). Aiming at improving the selection of derivations, Zhou et al. (2008) proposed prior derivation models utilizing syntactic annotation of the source language, which can be seen as smoothing the probabilities of hierarchical phrase features.

A key point is that the model we have introduced in this paper does not require the linguistic supervision needed in most of this prior work. We estimate the parameters of our model from parallel text without any linguistic annotation. That said, we would emphasize that our approach is, in fact, motivated in linguistic terms by the role of function words in natural language syntax.

10 Conclusion

We have presented a pairwise dominance model to address reordering issues that are not handled particularly well by standard hierarchical phrase-based modeling. In particular, the minimal linguistic commitment in hierarchical phrase-based models renders them susceptible to overgeneration of reordering choices. Our proposal handles the overgeneration problem by identifying hierarchical phrases with function words and by using function word relationships to incorporate soft constraints on topological orderings. Our experimental results demonstrate that introducing the pairwise dominance model into hierarchical phrase-based modeling improves performance significantly in large-scale Chinese-to-English and Arabic-to-English translation tasks.

Acknowledgments

This research was supported in part by the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-2-001. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the sponsors.

[9] We plan to do corresponding experimentation and analysis for Arabic once we identify a suitable list of manually identified function words.

References

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii, October.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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

Jiaying Howard. 2002. *A Student Handbook for Chinese Function Words.* The Chinese University Press.

Gonzalo Iglesias, Adria de Gispert, Eduardo R. Banga, and William Byrne. 2009. Rule filtering by pattern for efficient hierarchical translation. In *Proceedings of the 12th Conference of the European Chapter of the Association of Computational Linguistics (to appear).*

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing95*, pages 181–184, Detroit, MI, May.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, Edmonton, Alberta, Canada, May. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1003–1011, Columbus, Ohio, June.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

Hendra Setiawan, Min-Yen Kan, and Haizhou Li. 2007. Ordering phrases with function words. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 712–719, Prague, Czech Republic, June.

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 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 577–585, Columbus, Ohio, June.

David Vilar, Daniel Stein, and Hermann Ney. 2008. Analysing soft syntax features and heuristics for hierarchical phrase based machine translation. *International Workshop on Spoken Language Translation 2008*, pages 190–197, October.

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

Tiejun Zhao, Yajuan Lv, Jianmin Yao, Hao Yu, Muyun Yang, and Fang Liu. 2001. Increasing accuracy of chinese segmentation with strategy of multi-step processing. *Journal of Chinese Information Processing (Chinese Version)*, 1:13–18.

Bowen Zhou, Bing Xiang, Xiaodan Zhu, and Yuqing Gao. 2008. Prior derivation models for formally syntax-based translation using linguistically syntactic parsing and tree kernels. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 19–27, Columbus, Ohio, June.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June.