A Discriminative Syntactic Word Order Model for Machine Translation

Pi-Chuan Chang[*]
Computer Science Department
Stanford University
Stanford, CA 94305
pichuan@stanford.edu

Kristina Toutanova
Microsoft Research
Redmond, WA
kristout@microsoft.com

Abstract

We present a global discriminative statistical word order model for machine translation. Our model combines syntactic movement and surface movement information, and is discriminatively trained to choose among possible word orders. We show that combining discriminative training with features to detect these two different kinds of movement phenomena leads to substantial improvements in word ordering performance over strong baselines. Integrating this word order model in a baseline MT system results in a 2.4 points improvement in BLEU for English to Japanese translation.

1 Introduction

The machine translation task can be viewed as consisting of two subtasks: predicting the collection of words in a translation, and deciding the order of the predicted words. For some language pairs, such as English and Japanese, the ordering problem is especially hard, because the target word order differs significantly from the source word order.

Previous work has shown that it is useful to model target language order in terms of movement of syntactic constituents in constituency trees (Yamada and Knight, 2001; Galley et al., 2006) or dependency trees (Quirk et al., 2005), which are obtained using a parser trained to determine linguistic constituency. Alternatively, order is modelled in terms of movement of automatically induced hierarchical structure of sentences (Chiang, 2005; Wu, 1997).

The advantages of modeling how a target language syntax tree moves with respect to a source language syntax tree are that (*i*) we can capture the fact that constituents move as a whole and generally respect the phrasal cohesion constraints (Fox, 2002), and (*ii*) we can model broad syntactic reordering phenomena, such as subject-verb-object constructions translating into subject-object-verb ones, as is generally the case for English and Japanese.

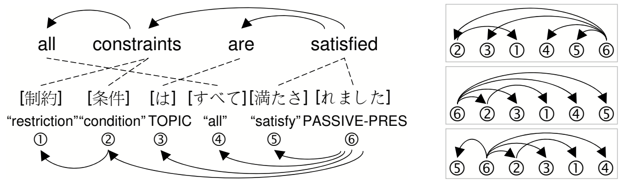
On the other hand, there is also significant amount of information in the surface strings of the source and target and their alignment. Many state-of-the-art SMT systems do not use trees and base the ordering decisions on surface phrases (Och and Ney, 2004; Al-Onaizan and Papineni, 2006; Kuhn et al., 2006). In this paper we develop an order model for machine translation which makes use of both syntactic and surface information.

The framework for our statistical model is as follows. We assume the existence of a dependency tree for the source sentence, an unordered dependency tree for the target sentence, and a word alignment between the target and source sentences. Figure 1 (a) shows an example of aligned source and target dependency trees. Our task is to order the target dependency tree.

We train a statistical model to select the best order of the unordered target dependency tree. An important advantage of our model is that it is global, and does not decompose the task of ordering a target sentence into a series of local decisions, as in the recently proposed order models for Machine Transition (Al-Onaizan and Papineni, 2006; Xiong et al., 2006; Kuhn et al., 2006). Thus we are able to define features over complete target sentence orders, and avoid the independence assumptions made by these

[*] This research was conducted during the author's internship at Microsoft Research.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 9–16,
Prague, Czech Republic, June 2007.



(a) (b)

Figure 1: (*a*) A sentence pair with source dependency tree, projected target dependency tree, and word alignments. (*b*) Example orders violating the target tree projectivity constraints.

models. Our model is discriminatively trained to select the best order (according to the BLEU measure) (Papineni et al., 2001) of an unordered target dependency tree from the space of possible orders.

Since the space of all possible orders of an unordered dependency tree is factorially large, we train our model on N-best lists of possible orders. These N-best lists are generated using approximate search and simpler models, as in the re-ranking approach of (Collins, 2000).

We first evaluate our model on the task of ordering target sentences, given correct (reference) unordered target dependency trees. Our results show that combining features derived from the source and target dependency trees, distortion surface order-based features (like the distortion used in Pharaoh (Koehn, 2004)) and language model-like features results in a model which significantly outperforms models using only some of the information sources.

We also evaluate the contribution of our model to the performance of an MT system. We integrate our order model in the MT system, by simply re-ordering the target translation sentences output by the system. The model resulted in an improvement from 33.6 to 35.4 BLEU points in English-to-Japanese translation on a computer domain.

2 Task Setup

The ordering problem in MT can be formulated as the task of ordering a target bag of words, given a source sentence and word alignments between target and source words. In this work we also assume a source dependency tree and an unordered target dependency tree are given. Figure 1(a) shows an example. We build a model that predicts an order of the target dependency tree, which induces an order on the target sentence words. The dependency tree constrains the possible orders of the target sentence only to the ones that are projective with respect to the tree. An order of the sentence is projective with respect to the tree if each word and its descendants form a contiguous subsequence in the ordered sentence. Figure 1(b) shows several orders of the sentence which violate this constraint.[1]

Previous studies have shown that if both the source and target dependency trees represent linguistic constituency, the alignment between subtrees in the two languages is very complex (Wellington et al., 2006). Thus such parallel trees would be difficult for MT systems to construct in translation. In this work only the source dependency trees are linguistically motivated and constructed by a parser trained to determine linguistic structure. The target dependency trees are obtained through projection of the source dependency trees, using the word alignment (we use GIZA++ (Och and Ney, 2004)), ensuring better parallelism of the source and target structures.

2.1 Obtaining Target Dependency Trees Through Projection

Our algorithm for obtaining target dependency trees by projection of the source trees via the word alignment is the one used in the MT system of (Quirk et al., 2005). We describe the algorithm schematically using the example in Figure 1. Projection of the dependency tree through alignments is not at all straightforward. One of the reasons of difficulty is that the alignment does not represent an isomorphism between the sentences, i.e. it is very often not a one-to-one and onto mapping.[2] If the alignment were one-to-one we could define the parent of a word w_t in the target to be the target word aligned to the parent of the source word s_i aligned to w_t. An additional difficulty is that such a definition could result in a non-projective target dependency tree. The projection algorithm of (Quirk et al., 2005) defines heuristics for each of these problems. In case of one-to-many alignments, for example, the case of "constraints" aligning to the Japanese words for "restriction" and "condition", the algorithm creates a

[1] For example, in the first order shown, the descendants of word 6 are not contiguous and thus this order violates the constraint.

[2] In an onto mapping, every word on the target side is associated with some word on the source side.

subtree in the target rooted at the rightmost of these words and attaches the other word(s) to it. In case of non-projectivity, the dependency tree is modified by re-attaching nodes higher up in the tree. Such a step is necessary for our example sentence, because the translations of the words “all” and “constraints” are not contiguous in the target even though they form a constituent in the source.

An important characteristic of the projection algorithm is that all of its heuristics use the *correct* target word order.[3] Thus the target dependency trees encode more information than is present in the source dependency trees and alignment.

2.2 Task Setup for Reference Sentences vs MT Output

Our model uses input of the same form when trained/tested on reference sentences and when used in machine translation: a source sentence with a dependency tree, an unordered target sentence with and unordered target dependency tree, and word alignments.

We train our model on reference sentences. In this setting, the given target dependency tree contains the correct bag of target words according to a reference translation, and is projective with respect to the correct word order of the reference by construction. We also evaluate our model in this setting; such an evaluation is useful because we can isolate the contribution of an order model, and develop it independently of an MT system.

When translating new sentences it is not possible to derive target dependency trees by the projection algorithm described above. In this setting, we use target dependency trees constructed by our baseline MT system (described in detail in 6.1). The system constructs dependency trees of the form shown in Figure 1 for each translation hypothesis. In this case the target dependency trees very often do not contain the correct target words and/or are not projective with respect to the best possible order.

3 Language Model with Syntactic Constraints: A Pilot Study

In this section we report the results of a pilot study to evaluate the difficulty of ordering a target sentence if we are given a target dependency tree as the one in Figure 1, versus if we are just given an unordered bag of target language words.

The difference between those two settings is that when ordering a target dependency tree, many of the orders of the sentence are not allowed, because they would be non-projective with respect to the tree. Figure 1 (b) shows some orders which violate the projectivity constraint. If the given target dependency tree is projective with respect to the correct word order, constraining the possible orders to the ones consistent with the tree can only help performance. In our experiments on reference sentences, the target dependency trees are projective by construction. If, however, the target dependency tree provided is not necessarily projective with respect to the best word order, the constraint may or may not be useful. This could happen in our experiments on ordering MT output sentences.

Thus in this section we aim to evaluate the usefulness of the constraint in both settings: reference sentences with projective dependency trees, and MT output sentences with possibly non-projective dependency trees. We also seek to establish a baseline for our task. Our methodology is to test a simple and effective order model, which is used by all state of the art SMT systems – a trigram language model – in the two settings: ordering an unordered bag of words, and ordering a target dependency tree.

Our experimental design is as follows. Given an unordered sentence t and an unordered target dependency tree $tree(t)$, we define two spaces of target sentence orders. These are the unconstrained space of all permutations, denoted by $\mathsf{Permutations}(t)$ and the space of all orders of t which are projective with respect to the target dependency tree, denoted by $\mathsf{TargetProjective}(t, tree(t))$. For both spaces S, we apply a standard trigram target language model to select a most likely order from the space; i.e., we find a target order $order^*{}_S(t)$ such that: $order^*{}_S(t) = argmax_{order(t) \in S} Pr_{LM}(order(t))$. The operator which finds $order^*{}_S(t)$ is difficult to implement since the task is NP-hard in both set-

[3] For example, checking which word is the rightmost for the heuristic for one-to-many mappings and checking whether the constructed tree is projective requires knowledge of the correct word order of the target.

Reference Sentences		
Space	**BLEU**	**Avg. Size**
Permutations	58.8	2^{61}
TargetProjective	83.9	2^{29}
MT Output Sentences		
Space	**BLEU**	**Avg. Size**
Permutations	26.3	2^{56}
TargetProjective	31.7	2^{25}

Table 1: Performance of a tri-gram language model on ordering reference and MT output sentences: unconstrained or subject to target tree projectivity constraints.

tings, even for a bi-gram language model (Eisner and Tromble, 2006).[4] We implemented left-to-right beam A* search for the Permutations space, and a tree-based bottom up beam A* search for the TargetProjective space. To give an estimate of the search error in each case, we computed the number of times the correct order had a better language model score than the order returned by the search algorithm.[5] The lower bounds on search error were 4% for Permutations and 2% for TargetProjective, computed on reference sentences.

We compare the performance in BLEU of orders selected from both spaces. We evaluate the performance on reference sentences and on MT output sentences. Table 1 shows the results. In addition to BLEU scores, the table shows the median number of possible orders per sentence for the two spaces.

The highest achievable BLEU on reference sentences is 100, because we are given the correct bag of words. The highest achievable BLEU on MT output sentences is well below 100 (the BLEU score of the MT output sentences is 33). Table 3 describes the characteristics of the main data-sets used in the experiments in this paper; the test sets we use in the present pilot study are the reference test set (Reftest) of 1K sentences and the MT test set (MT-test) of 1,000 sentences.

The results from our experiment show that the target tree projectivity constraint is extremely powerful on reference sentences, where the tree given is indeed projective. (Recall that in order to obtain the target dependency tree in this setting we have used information from the true order, which explains in part the large performance gain.)

The gain in BLEU due to the constraint was not as large on MT output sentences, but was still considerable. The reduction in search space size due to the constraint is enormous. There are about 2^{30} times fewer orders to consider in the space of target projective orders, compared to the space of all permutations. From these experiments we conclude that the constraints imposed by a projective target dependency tree are extremely informative. We also conclude that the constraints imposed by the target dependency trees constructed by our baseline MT system are very informative as well, even though the trees are not necessarily projective with respect to the best order. Thus the projectivity constraint with respect to a reasonably good target dependency tree is useful for addressing the search and modeling problems for MT ordering.

4 A Global Order Model for Target Dependency Trees

In the rest of the paper we present our new word order model and evaluate it on reference sentences and in machine translation. In line with previous work on NLP tasks such as parsing and recent work on machine translation, we develop a discriminative order model. An advantage of such a model is that we can easily combine different kinds of features (such as syntax-based and surface-based), and that we can optimize the parameters of our model directly for the evaluation measures of interest.

Additionally, we develop a globally normalized model, which avoids the independence assumptions in locally normalized conditional models.[6] We train a global log-linear model with a rich set of syntactic and surface features. Because the space of possible orders of an unordered dependency tree is factorially large, we use simpler models to generate N-best orders, which we then re-rank with a global model.

4.1 Generating N-best Orders

The simpler models which we use to generate N-best orders of the unordered target dependency trees are the standard trigram language model used in Section 3, and another statistical model, which we call a Local Tree Order Model (LTOM). The LTOM model

[4] Even though the dependency tree constrains the space, the number of children of a node is not bounded by a constant.

[5] This is an underestimate of search error, because we don't know if there was another (non-reference) order which had a better score, but was not found.

[6] Those models often assume that current decisions are independent of future observations.

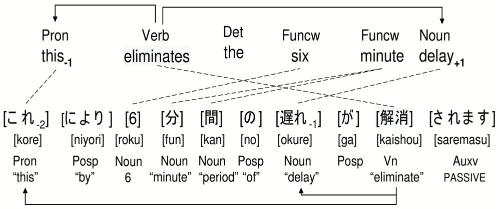


Figure 2: Dependency parse on the source (English) sentence, alignment and projected tree on the target (Japanese) sentence. Notice that the projected tree is only partial and is used to show the head-relative movement.

uses syntactic information from the source and target dependency trees, and orders each local tree of the target dependency tree independently. It follows the order model defined in (Quirk et al., 2005).

The model assigns a probability to the position of each target node (modifier) relative to its parent (head), based on information in both the source and target trees. The probability of an order of the complete target dependency tree decomposes into a product over probabilities of positions for each node in the tree as follows:

$$P(order(t)|s,t) = \prod_{n \in t} P(pos(n, parent(n))|s,t)$$

Here, position is modelled in terms of closeness to the head in the dependency tree. The closest pre-modifier of a given head has position -1; the closest post-modifier has a position 1. Figure 2 shows an example dependency tree pair annotated with head-relative positions. A small set of features is used to reflect local information in the dependency tree to model $P(pos(n, parent(n))|s,t)$: (*i*) lexical items of n and $parent(n)$, (*ii*) lexical items of the source nodes aligned to n and $parent(n)$, (*iii*) part-of-speech of the source nodes aligned to the node and its parent, and (*iv*) head-relative position of the source node aligned to the target node.

We train a log-linear model which uses these features on a training set of aligned sentences with source and target dependency trees in the form of Figure 2. The model is a local (non-sequence) classifier, because the decision on where to place each node does not depend on the placement of any other nodes.

Since the local tree order model learns to order whole subtrees of the target dependency tree, and since it uses syntactic information from the source, it provides an alternative view compared to the trigram language model. The example in Figure 2 shows that the head word "eliminates" takes a dependent "this" to the left (position -1), and on the Japanese side, the head word "kaishou" (corresponding to "eliminates") takes a dependent "kore" (corresponding to "this") to the left (position -2). The trigram language model would not capture the position of "kore" with respect to "kaishou", because the words are farther than three positions away.

We use the language model and the local tree order model to create N-best target dependency tree orders. In particular, we generate the N-best lists from a simple log-linear combination of the two models:

$$P(o(t)|s,t) \propto P_{LM}(o(t)|t)P_{LTOM}(o(t)|s,t)^{\lambda}$$

where $o(t)$ denotes an order of the target.[7] We used a bottom-up beam A* search to generate N-best orders. The performance of each of these two models and their combination, together with the 30-best oracle performance on reference sentences is shown in Table 2. As we can see, the 30-best oracle performance of the combined model (98.0) is much higher than the 1-best performance (92.6) and thus there is a lot of room for improvement.

4.2 Model

The log-linear reranking model is defined as follows. For each sentence pair sp_l ($l = 1, 2, ..., L$) in the training data, we have N candidate target word orders $o_{l,1}, o_{l,2}, ..., o_{l,N}$, which are the orders generated from the simpler models. Without loss of generality, we define $o_{l,1}$ to be the order with the highest BLEU score with respect to the correct order.[8]

We define a set of feature functions $f_m(o_{l,n}, sp_l)$ to describe a target word order $o_{l,n}$ of a given sentence pair sp_l. In the log-linear model, a corresponding weights vector λ is used to define the distribution over all possible candidate orders:

$$p(o_{l,n}|sp_l, \lambda) = \frac{e^{\lambda F(o_{l,n}, sp_l)}}{\sum_{n'} e^{\lambda F(o_{l,n'}, sp_l)}}$$

[7] We used the value $\lambda = .5$, which we selected on a development set to maximize BLEU.

[8] To avoid the problem that all orders could have a BLEU score of 0 if none of them contains a correct word four-gram, we define sentence-level k-gram BLEU, where k is the highest order, $k \leq 4$, for which there exists a correct k-gram in at least one of the N-Best orders.

We train the parameters λ by minimizing the negative log-likelihood of the training data plus a quadratic regularization term:

$$L(\lambda) = -\sum_l \log p(o_{l,1}|sp_i, \lambda) + \frac{1}{2\sigma^2}\sum_m {\lambda_m}^2$$

We also explored maximizing expected BLEU as our objective function, but since it is not convex, the performance was less stable and ultimately slightly worse, as compared to the log-likelihood objective.

4.3 Features

We design features to capture both the head-relative movement and the surface sequence movement of words in a sentence. We experiment with different combinations of features and show their contribution in Table 2 for reference sentences and Table 4 in machine translation. The notations used in the tables are defined as follows:

Baseline: **LTOM+LM** as described in Section 4.1

Word Bigram: Word bigrams of the target sentence. Examples from Figure 2: *"kore"*+*"niyori"*, *"niyori"*+*"roku"*.

DISP: Displacement feature. For each word position in the target sentence, we examine the alignment of the current word and the previous word, and categorize the possible patterns into 3 kinds: (a) parallel, (b) crossing, and (c) widening. Figure 3 shows how these three categories are defined.

Pharaoh DISP: Displacement as used in Pharaoh (Koehn, 2004). For each position in the sentence, the value of the feature is one less than the difference (absolute value) of the positions of the source words aligned to the current and the previous target word.

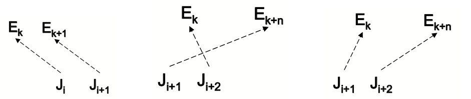
POSs and **POSt**: POS tags on the source and target sides. For Japanese, we have a set of 19 POS tags.

'+' means making conjunction of features and $prev()$ means using the information associated with the word from position -1.

In all explored models, we include the log-probability of an order according to the language model and the log-probability according to the local tree order model, the two features used by the baseline model.

5 Evaluation on Reference Sentences

Our experiments on ordering reference sentences use a set of 445K English sentences with their reference Japanese translations. This is a subset of the set MT-train in Table 3. The sentences were annotated with alignment (using GIZA++ (Och and Ney, 2004)) and syntactic dependency structures of the source and target, obtained as described in Section 2. Japanese POS tags were assigned by an automatic POS tagger, which is a local classifier not using tag sequence information.



(a) parallel (b) crossing (c) widening

Figure 3: Displacement feature: different alignment patterns of two contiguous words in the target sentence.

We used 400K sentence pairs from the complete set to train the first pass models: the language model was trained on 400K sentences, and the local tree order model was trained on 100K of them. We generated N-best target tree orders for the rest of the data (45K sentence pairs), and used it for training and evaluating the re-ranking model. The re-ranking model was trained on 44K sentence pairs. All models were evaluated on the remaining 1,000 sentence pairs set, which is the set Ref-test in Table 3.

The top part of Table 2 presents the 1-best BLEU scores (actual performance) and 30-best oracle BLEU scores of the first-pass models and their log-linear combination, described in Section 4. We can see that the combination of the language model and the local tree order model outperformed either model by a large margin. This indicates that combining syntactic (from the LTOM model) and surface-based (from the language model) information is very effective even at this stage of selecting N-best orders for re-ranking. According to the 30-best oracle performance of the combined model LTOM+LM, 98.0 BLEU is the upper bound on performance of our re-ranking approach.

The bottom part of the table shows the performance of the global log-linear model, when features in addition to the scores from the two first-pass models are added to the model. Adding word-bigram features increased performance by about 0.6 BLEU points, indicating that training language-model like features discriminatively to optimize ordering performance, is indeed worthwhile. Next we compare

First-pass models		
Model	BLEU	
	1 best	30 best
Lang Model (Permutations)	58.8	71.2
Lang Model (TargetProjective)	83.9	95.0
Local Tree Order Model	75.8	87.3
Local Tree Order Model + Lang Model	92.6	98.0
Re-ranking Models		
Features	BLEU	
Baseline	92.60	
Word Bigram	93.19	
Pharaoh DISP	92.94	
DISP	93.57	
DISP+POSs	94.04	
DISP+POSs+POSt	94.14	
DISP+POSs+POSt, prev(DISP)+POSs+POSt	94.34	
DISP+POSs+POSt, prev(DISP)+POSs+POSt, WB	94.50	

Table 2: Performance of the first-pass order models and 30-best oracle performance, followed by performance of re-ranking model for different feature sets. Results are on reference sentences.

the Pharaoh displacement feature to the displacement feature we illustrated in Figure 3. We can see that the Pharaoh displacement feature improves performance of the baseline by .34 points, whereas our displacement feature improves performance by nearly 1 BLEU point. Concatenating the DISP feature with the POS tag of the source word aligned to the current word improved performance slightly.

The results show that surface movement features (i.e. the DISP feature) improve the performance of a model using syntactic-movement features (i.e. the LTOM model). Additionally, adding part-of-speech information from both languages in combination with displacement, and using a higher order on the displacement features was useful. The performance of our best model, which included all information sources, is 94.5 BLEU points, which is a 35% improvement over the fist-pass models, relative to the upper bound.

6 Evaluation in Machine Translation

We apply our model to machine translation by reordering the translation produced by a baseline MT system. Our baseline MT system constructs, for each target translation hypothesis, a target dependency tree. Thus we can apply our model to MT output in exactly the same way as for reference sentences, but using much noisier input: a source sentence with a dependency tree, word alignment and an unordered target dependency tree as the example shown in Figure 2. The difference is that the target dependency tree will likely not contain the correct target words and/or will not be projective with respect to the best possible order.

data set	num sent.	English		Japanese	
		avg. len	vocab	avg. len	vocab
MT-train	500K	15.8	77K	18.7	79K
MT-test	1K	17.5	–	20.9	–
Ref-test	1K	17.5	–	21.2	–

Table 3: Main data sets used in experiments.

6.1 Baseline MT System

Our baseline SMT system is the system of Quirk et al. (2005). It translates by first deriving a dependency tree for the source sentence and then translating the source dependency tree to a target dependency tree, using a set of probabilistic models. The translation is based on treelet pairs. A treelet is a connected subgraph of the source or target dependency tree. A treelet translation pair is a pair of word-aligned source and target treelets.

The baseline SMT model combines this treelet translation model with other feature functions — a target language model, a tree order model, lexical weighting features to smooth the translation probabilities, word count feature, and treelet-pairs count feature. These models are combined as feature functions in a (log)linear model for predicting a target sentence given a source sentence, in the framework proposed by (Och and Ney, 2002). The weights of this model are trained to maximize BLEU (Och and Ney, 2004). The SMT system is trained using the same form of data as our order model: parallel source and target dependency trees as in Figure 2.

Of particular interest are the components in the baseline SMT system contributing most to word order decisions. The SMT system uses the same target language trigram model and local tree order model, as we are using for generating N-best orders for reranking. Thus the baseline system already uses our first-pass order models and only lacks the additional information provided by our re-ranking order model.

6.2 Data and Experimental Results

The baseline MT system was trained on the MT-train dataset described in Table 3. The test set for the MT experiment is a 1K sentences set from the same domain (shown as MT-test in the table). The weights in the linear model used by the baseline SMT system were tuned on a separate development set.

Table 4 shows the performance of the first-pass models in the top part, and the performance of our

First-pass models		
Model	BLEU	
	1 best	30 best
Baseline MT System	33.0	–
Lang Model (Permutations)	26.3	28.7
Lang Model (TargetCohesive)	31.7	35.0
Local Tree Order Model	27.2	31.5
Local Tree Order Model + Lang Model	33.6	36.0
Re-ranking Models		
Features	BLEU	
Baseline	33.56	
Word Bigram	34.11	
Pharaoh DISP	34.67	
DISP	34.90	
DISP+POSs	35.28	
DISP+POSs+POSt	35.22	
DISP+POSs+POSt, prev(DISP)+POSs+POSt	35.33	
DISP+POSs+POSt, prev(DISP)+POSs+POSt, WB	35.37	

Table 4: Performance of the first pass order models and 30-best oracle performance, followed by performance of re-ranking model for different feature sets. Results are in MT.

re-ranking model in the bottom part. The first row of the table shows the performance of the baseline MT system, which is a BLEU score of 33. Our first-pass and re-ranking models re-order the words of this 1-best output from the MT system. As for reference sentences, the combination of the two first-pass models outperforms the individual models. The 1-best performance of the combination is 33.6 and the 30-best oracle is 36.0. Thus the best we could do with our re-ranking model in this setting is 36 BLEU points.[9] Our best re-ranking model achieves 2.4 BLEU points improvement over the baseline MT system and 1.8 points improvement over the first-pass models, as shown in the table. The trends here are similar to the ones observed in our reference experiments, with the difference that target POS tags were less useful (perhaps due to ungrammatical candidates) and the displacement features were more useful. We can see that our re-ranking model almost reached the upper bound oracle performance, reducing the gap between the first-pass models performance (33.6) and the oracle (36.0) by 75%.

7 Conclusions and Future Work

We have presented a discriminative syntax-based order model for machine translation, trained to to select from the space of orders projective with respect to a target dependency tree. We investigated a combination of features modeling surface movement and syntactic movement phenomena and showed that these two information sources are complementary and their combination is powerful. Our results on ordering MT output and reference sentences were very encouraging. We obtained substantial improvement by the simple method of post-processing the 1-best MT output to re-order the proposed translation. In the future, we would like to explore tighter integration of our order model with the SMT system and to develop more accurate algorithms for constructing projective target dependency trees in translation.

[9] Notice that the combination of our two first-pass models outperforms the baseline MT system by half a point (33.6 versus 33.0). This is perhaps due to the fact that the MT system searches through a much larger space (possible word translations in addition to word orders), and thus could have a higher search error.

References

Y. Al-Onaizan and K. Papineni. 2006. Distortion models for statistical machine translation. In *ACL*.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL*.

M. Collins. 2000. Discriminative reranking for natural language parsing. In *ICML*, pages 175–182.

J Eisner and R. W. Tromble. 2006. Local search with very large-scale neighborhoods for optimal permutations in machine translation. In *HLT-NAACL Workshop*.

H. Fox. 2002. Phrasal cohesion and statistical machine translation. In *EMNLP*.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL*.

P. Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*.

R. Kuhn, D. Yuen, M. Simard, P. Paul, G. Foster, E. Joanis, and H. Johnson. 2006. Segment choice models: Feature-rich models for global distortion in statistical machine translation. In *HLT-NAACL*.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *ACL*.

B. Wellington, S. Waxmonsky, and I. Dan Melamed. 2006. Empirical lower bounds on the complexity of translational equivalence. In *ACL-COLING*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

D. Xiong, Q. Liu, and S. Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *ACL*.

K. Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *ACL*.