Proceedings of the 16th EAMT Conference, 28-30 May 2012, Trento, Italy

Discriminative Reordering Extensions for Hierarchical Phrase-Based Machine Translation

Matthias Huck and **Stephan Peitz** and **Markus Freitag** and **Hermann Ney**

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

In this paper, we propose novel extensions of hierarchical phrase-based systems with a discriminative lexicalized reordering model. We compare different feature sets for the discriminative reordering model and investigate combinations with three types of non-lexicalized reordering rules which are added to the hierarchical grammar in order to allow for more reordering flexibility during decoding. All extensions are evaluated in standard hierarchical setups as well as in setups where the hierarchical recursion depth is restricted. We achieve improvements of up to +1.2 %BLEU on a large-scale Chinese→English translation task.

1 Introduction

Lexicalized reordering models are a common component of standard phrase-based machine translation systems. In hierarchical phrase-based machine translation, reordering is modeled implicitely as part of the translation model. Hierarchical phrase-based decoders conduct phrase reorderings based on a one-to-one relation between the non-terminals on source and target side within hierarchical translation rules. Non-terminals on source and target side are linked if they result from the same valid phrase being cut out at their position during phrase extraction. Usually neither explicit lexicalized reordering models nor additional mechanisms to perform reorderings that do not result from the application of hierarchical rules are integrated into hierarchical decoders.

In this work, we augment the grammar with more flexible reordering mechanisms based on additional non-lexicalized reordering rules and integrate a discriminative lexicalized reordering model. This kind of model has been shown to perform well when being added to the log-linear model combination of standard phrase-based systems. We present an extension of a hierarchical decoder with the discriminative reordering model and evaluate it in setups with the usual hierarchical grammar as well as in setups with a shallow hierarchical grammar. The shallow grammar restricts the depth of the hierarchical recursion. Two different feature sets for the discriminative reordering model are examined. We report experimental results on the large-scale NIST Chinese→English translation task. The best translation quality is achieved with combinations of the extensions with additional reordering rules and with the discriminative reordering model. The overall improvement over the respective baseline system is +1.2 %BLEU / -0.6 %TER absolute in the standard setup and +1.2 %BLEU / -0.5 %TER absolute in the shallow setup.

2 Related Work

Hierarchical phrase-based translation was proposed by Chiang (2005). Iglesias et al. (2009) and in a later journal publication Gispert et al. (2010) present a way to limit the recursion depth for hierarchical rules by means of a modification to the hierarchical grammar. Their work is of interest to us as a limitation of the recursion depth affects the search space and in particular the reordering capabilities of the system. It is therefore basically antipodal to some of the techniques presented in this paper, which allow for even more flexibility during the search process by extending the grammar with specific non-lexicalized reordering rules. Combinations of both techniques are possible, though, and in fact Iglesias et al. (2009) also investigate a maximum phrase jump of 1 (MJ1) reordering model. In the MJ1 experiment, they include a swap rule, but simultaneously withdraw all hierarchical phrases.

Vilar et al. (2010) extend a hierarchical phrase-based system with non-lexicalized rules that permit jumps across whole blocks of symbols and report improvements on a German→English Europarl task. Their technique is inspired by conventional phrase-based IBM-style reordering (Zens et al., 2004). In an Arabic→English NIST setup, Huck et al. (2011) try a similar reordering extension, but conclude that it is less helpful for their task. Other groups attempt to attain superior modeling of reordering effects in their hierarchical systems by examining syntactic annotation, e.g. Gao et al. (2011).

He et al. (2010a) combine an additional BTG-style swap rule with a maximum entropy based lexicalized reordering model and achieve improvements on the Chinese→English NIST task. Their approach is comparable to ours, but their reordering model requires the training of different classifiers for different rule patterns (He et al., 2010b). Extracting training instances separately for several patterns of hierarchical rules yields a dependence on the phrase segmentation. In the more general approach we propose, the definition of the features is independent of the phrase boundaries on the source side.

In standard phrase-based systems, lexicalized reordering models are a commonly included component. A widely used variant is the orientation model as implemented in the Moses toolkit (Tillmann, 2004; Koehn et al., 2007) which distinguishes monotone, swap, and discontinuous phrase orientations. Galley and Manning (2008) suggest a refinement of the same model. A discriminatively trained lexicalized reordering model as the one employed by us has been examined in a standard phrase-based setting by Zens and Ney (2006).

3 Shallow-1 Grammar

Gispert et al. (2010) propose a limitation of the recursion depth for hierarchical rules with shallow-n grammars. The main benefit of the limitation is a gain in decoding efficiency. Moreover, the modification of the grammar to a shallow version restricts the search space of the decoder and may be convenient to prevent overgeneration. We will investigate reordering extensions to both standard hierarchical systems and systems with a shallow-1 grammar, i.e. a grammar which limits the depth of the hierarchical recursion to one. We refer to this kind of rule set and the parses produced with such a grammar as *shallow*, in contrast to the standard rule set and parses which we denote as *deep*.

In a shallow-1 grammar, the generic non-terminal X of the standard hierarchical approach is replaced by two distinct non-terminals XH and XP. By changing the left-hand sides of the rules, lexical phrases are allowed to be derived from XP only, hierarchical phrases from XH only. On all right-hand sides of hierarchical rules, the X is replaced by XP. Gaps within hierarchical phrases can thus be filled with contiguous lexical phrases only, not with hierarchical phrases. The initial rule is substituted with

$$\begin{aligned} S &\to \langle XP^{\sim 0}, XP^{\sim 0} \rangle \\ S &\to \langle XH^{\sim 0}, XH^{\sim 0} \rangle \,, \end{aligned} \tag{1}$$

and the glue rule is substituted with

$$\begin{aligned} S &\to \langle S^{\sim 0} XP^{\sim 1}, S^{\sim 0} XP^{\sim 1} \rangle \\ S &\to \langle S^{\sim 0} XH^{\sim 1}, S^{\sim 0} XH^{\sim 1} \rangle \,. \end{aligned} \tag{2}$$

4 Reordering Rules

In this section we describe three types of reordering extensions to the hierarchical grammar. All of them add specific non-lexicalized reordering rules which facilitate a more flexible arrangement of phrases in the hypotheses. We first present a simple swap rule extension (Section 4.1), then we suggest two different extensions with several additional rules that allow for more complex jumps (Section 4.2) or very constrained jumps (Section 4.3). Furthermore, variants for deep and shallow grammars are proposed.

4.1 Swap Rule

4.1.1 Swap Rule for Deep Grammars

In a deep grammar, we can bring in more reordering capabilities by adding a single swap rule

$$X \to \langle X^{\sim 0} X^{\sim 1}, X^{\sim 1} X^{\sim 0} \rangle \tag{3}$$

supplementary to the standard initial rule and glue rule. The swap rule allows adjacent phrases to be transposed.

An alternative with a comparable effect would be to remove the standard glue rule and to add two rules instead, one of them being as in Equation (3) and the other a monotonic concatenation rule for the non-terminal X which is symmetric to the swap rule. The latter rule acts as a replacement for the glue rule. This is the approach He et al. (2010a) take. Our approach to keep the standard glue rule has however one advantage: We are still able to apply a maximum length constraint to X. The maximum length constraint restricts the length of the yield of a non-terminal. The lexical span covered by X is typically restricted to 10 to make decoding less demanding in terms of computational resources. We would still be able to add a monotonic concatenation rule to our grammar in addition to the standard glue rule. Its benefit is that it entails more symmetry in the grammar. In our variant, sub-derivations which result from applications of the swap rule can fill the gap within hierarchical phrases, while no mechanism to carry out the same in a monotonic manner is available. In the deep grammar, we refrain from adding a monotonic concatenation rule as recursive embeddings are possible anyway. We nevertheless tried the variant with the additional monotonic concatenation rule in a supplementary experiment (cf. Section 6.2.2) to make sure that our assumption that this rule is dispensable is correct. We were not able to obtain improvements over the setup with the swap rule only.

4.1.2 Swap Rule for Shallow Grammars

In a shallow grammar, several directions of integrating swaps are possible. We decided to add a swap rule and a monotonic concatenation rule

$$\begin{aligned} XP &\rightarrow \langle XP^{\sim 0} XP^{\sim 1}, XP^{\sim 1} XP^{\sim 0} \rangle \\ XP &\rightarrow \langle XP^{\sim 0} XP^{\sim 1}, XP^{\sim 0} XP^{\sim 1} \rangle \end{aligned} \tag{4}$$

supplementary to the standard shallow initial rules and glue rules. The swap rule allows adjacent lexical phrases to be transposed, but not hierarchical phrases. Here, we could as well have used XH as the left-hand side of the rules. As we chose XP and thus allow for embedding of sub-derivations resulting from applications of the swap rule into hierarchical phrases, which is not possible with sub-derivations resulting from applications of hierarchical rules in a shallow grammar, we also include the monotonic concatenation rule for symmetry reasons. A constraint can again be applied to the number of terminals spanned by both XP and XH. With a length constraint, building sub-derivations of arbitrary length by applying the rules from Equation (4) is impossible.

4.2 Jump Rules, Variant 1

Instead of employing a swap rule that transposes adjacent phrases, we can adopt more complex extensions to the grammar that implement jumps across blocks of symbols. Our first jump rules variant is inspired by Vilar et al. (2010), but is a generalization that facilitates an arbitrary number of blocks per sentence to be jumped across.

4.2.1 Jump Rules for Deep Grammars

In a deep grammar, to enable block jumps, we include the rules

$$\begin{aligned} S &\rightarrow \langle B^{\sim 0} X^{\sim 1}, X^{\sim 1} B^{\sim 0} \rangle & \dagger \\ S &\rightarrow \langle S^{\sim 0} B^{\sim 1} X^{\sim 2}, S^{\sim 0} X^{\sim 2} B^{\sim 1} \rangle & \dagger \\ B &\rightarrow \langle X^{\sim 0}, X^{\sim 0} \rangle & \\ B &\rightarrow \langle B^{\sim 0} X^{\sim 1}, B^{\sim 0} X^{\sim 1} \rangle & \ddagger \end{aligned} \tag{5}$$

in addition to the standard initial rule and glue rule. The rules marked with † are jump rules that put jumps across blocks (B) on source side into effect. The rules with B on their left-hand side enable blocks that are skipped by the jump rules to be translated, but without further jumps. Reordering within these windows is just possible with hierarchical rules. Note that our rule set keeps the convenient property of the standard hierarchical grammar that the initial symbol S needs to be expanded in the leftmost cells of the CYK chart only.

A binary jump feature for the two jump rules (†) may be added to the log-linear model combination of the decoder, as well as a binary feature that fires for the rule that acts analogous to the glue rule, but within blocks that is being jumped across (‡). A maximum jump width can be established by applying a length constraint to the non-terminal B. A distance-based distortion model can also easily be implemented by computing the span width of the non-terminal B on the right-hand side of the jump rules at each application of one of them.

4.2.2 Jump Rules for Shallow Grammars

In a shallow grammar, block jumps are realized in the same way as in a deep one, but the number of rules that are required is doubled.

We include

$$
\begin{aligned}
&S \to \langle B^{\sim 0} X P^{\sim 1}, X P^{\sim 1} B^{\sim 0} \rangle && \dagger\\
&S \to \langle B^{\sim 0} X H^{\sim 1}, X H^{\sim 1} B^{\sim 0} \rangle && \dagger\\
&S \to \langle S^{\sim 0} B^{\sim 1} X P^{\sim 2}, S^{\sim 0} X P^{\sim 2} B^{\sim 1} \rangle && \dagger\\
&S \to \langle S^{\sim 0} B^{\sim 1} X H^{\sim 2}, S^{\sim 0} X H^{\sim 2} B^{\sim 1} \rangle && \dagger\\
&B \to \langle X P^{\sim 0}, X P^{\sim 0} \rangle &&\\
&B \to \langle X H^{\sim 0}, X H^{\sim 0} \rangle &&\\
&B \to \langle B^{\sim 0} X P^{\sim 1}, B^{\sim 0} X P^{\sim 1} \rangle && \ddagger\\
&B \to \langle B^{\sim 0} X H^{\sim 1}, B^{\sim 0} X H^{\sim 1} \rangle && \ddagger
\end{aligned}
\tag{6}
$$

in addition to the standard shallow initial rules and glue rules.

4.3 Jump Rules, Variant 2

As a second jump rules variant, we try an approach that follows (Huck et al., 2011) and allows for very constrained reorderings. At most one contiguous block per sentence can be jumped across in this variant.

In a deep grammar, to enable constrained block jumps with at most one jump per sentence, we replace the initial and glue rule by the rules given in Equation (7):

$$
\begin{aligned}
&S \to \langle M^{\sim 0}, M^{\sim 0} \rangle &&\\
&S \to \langle S^{\sim 0} M^{\sim 1}, S^{\sim 0} M^{\sim 1} \rangle && \ddagger\\
&S \to \langle B^{\sim 0} M^{\sim 1}, M^{\sim 1} B^{\sim 0} \rangle && \dagger\\
&M \to \langle X^{\sim 0}, X^{\sim 0} \rangle &&\\
&M \to \langle M^{\sim 0} X^{\sim 1}, M^{\sim 0} X^{\sim 1} \rangle && \ddagger\\
&B \to \langle X^{\sim 0}, X^{\sim 0} \rangle &&\\
&B \to \langle B^{\sim 0} X^{\sim 1}, B^{\sim 0} X^{\sim 1} \rangle && \ddagger
\end{aligned}
\tag{7}
$$

In these rules, the M non-terminal represents a block that will be translated in a monotonic way, and the B is a "back jump". We omit the exposition for shallow grammars as deducing the shallow from the deep version of the rules is straightforward from our previous explanations.

We add a binary feature that fires for the rules that act analogous to the glue rule (‡). We further conform to the approach of Huck et al. (2011) by additionally including a distance-based distortion model (*dist. feature*) that is computed during decoding whenever the back jump rule (†) is applied.

5 Discriminative Reordering Model

Our discriminative reordering extensions for hierarchical phrase-based machine translation systems integrate a discriminative reordering model that tries to predict the orientation of neighboring blocks. We use two orientation classes *left* and *right*, in the same manner as described by Zens and Ney (2006). The reordering model is applied at the phrase boundaries only, where words which are adjacent to gaps within hierarchical phrases are defined as boundary words as well. The orientation probability is modeled in a maximum entropy framework. We investigate two models that differ in the set of feature functions:

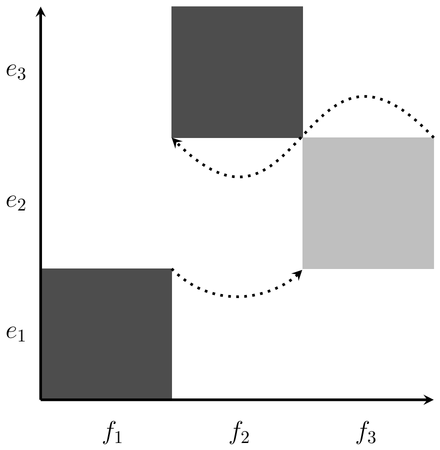

Figure 1: Illustration of an embedding of a lexical phrase (light) in a hierarchical phrase (dark), with orientations scored with the neighboring blocks.

discrim. RO (src word) The feature set of this model consists of binary features based on the source word at the current source position.

discrim. RO (src+tgt word+class) The feature set of this model consists of binary features based on the source word and word class at the current source position and the target word and word class at the current target position.

Using features that depend on word classes provides generalization capabilities. We employ 100 automatically learned word classes which are obtained with the `mkcls` tool on both source and target side.[1] The reordering model is trained with the Generalized Iterative Scaling (GIS) algorithm with the maximum class posterior probability as training criterion, and it is smoothed with a gaussian prior.

For each rule application during hierarchical decoding, we apply the reordering model at all

[1] `mkcls` is distributed along with the GIZA++ package: `http://code.google.com/p/giza-pp/`

boundaries where lexical blocks are placed side by side within the partial hypothesis. For this purpose, we need to access neighboring boundary words and their aligned source words and source positions. Note that, as hierarchical phrases are involved, several block joinings may take place at once during a single rule application. Figure 1 gives an illustration with an embedding of a lexical phrase (light) in a hierarchical phrase (dark). The gap in the hierarchical phrase $\langle f_1 f_2 X^{\sim 0}, e_1 X^{\sim 0} e_3 \rangle$ is filled with the lexical phrase $\langle f_3, e_2 \rangle$. The discriminative reordering model scores the orientation of the lexical phrase with regard to the neighboring block of the hierarchical phrase which precedes it within the target sequence (here: right orientation), and the block of the hierarchical phrase which succeeds the lexical phrase with regard to the latter (here: left orientation).

The way we interpret reordering in hierarchical phrase-based translation keeps our model simple. We are basically able to treat the orientation of contiguous lexical blocks in almost exactly the same way as the orientation of phrases in standard phrase-based translation. We avoid the usage of multiple reordering models for different source and target patterns of rules that is done by He et al. (2010b).

6 Experiments

We present empirical results obtained with the additional swap rule, the jump rules and the discriminative reordering model on the Chinese→English 2008 NIST task.[2]

6.1 Experimental Setup

We employ the freely available hierarchical translation toolkit Jane (Vilar et al., 2010) to set up our systems. In our experiments, we use the cube pruning algorithm (Huang and Chiang, 2007) to carry out the search. A maximum length constraint of 10 is applied to all non-terminals but the initial symbol S. We work with a parallel training corpus of 3.0M Chinese-English sentence pairs (77.5M Chinese / 81.0M English running words). Word alignments are created by aligning the data in both directions with GIZA++ and symmetrizing the two trained alignments (Och and Ney, 2003). The language model is a 4-gram with modified Kneser-Ney smoothing which was trained with the SRILM toolkit (Stolcke, 2002).

Model weights are optimized against BLEU with Minimum Error Rate Training on 100-best lists. We employ MT06 as development set to tune the model weights, MT08 is used as unseen test set. The performance of the systems is evaluated using the two metrics BLEU and TER. The results on the test set are checked for statistical significance over the baseline. Confidence intervals have been computed using bootstrapping for BLEU and Cochran's approximate ratio variance for TER (Leusch and Ney, 2009).

6.2 Experimental Results

The empirical evaluation of our reordering extensions is presented in Table 1. We report translation results on both the development and the test corpus. The figures with deep and with shallow rules are set side by side in separate columns to facilitate a direct comparison between them. All the setups given in separate rows exist in a deep and a shallow variant.

The shallow baseline is a bit worse than the deep baseline. Adding discriminative reordering models to the baselines without additional reordering rules results in an improvement of up to +0.6 %BLEU / -0.6 %TER (in the deep setup). The *src+tgt word+class* feature set for the discriminative reordering model altogether seems to perform slightly better than the *src word* feature set. Adding reordering rules in isolation can also improve the systems, in particular in the deep setup with the swap rule or the second jump rules variant. However, extensions with both reordering rules and discriminative lexicalized reordering model provide the best results, e.g. +1.0 %BLEU / -0.5 %TER with the system with deep grammar, swap rule, binary swap feature and discrim. RO (src+tgt word+class) and +1.2 %BLEU / -0.5 %TER with the system with shallow grammar, swap rule, binary swap feature and discrim. RO (src+tgt word+class). The second jump rules variant performs particularly well in combination with a deep grammar and the discrim. RO (src+tgt word+class) model, with an improvement of +1.2 %BLEU / -0.6 %TER absolute over the deep baseline. This system provides the best translation quality of all the setups investigated in this paper. With a shallow grammar, the combinations of the discrim. RO with the swap rule outperforms both

[2] http://www.itl.nist.gov/iad/mig/tests/mt/2008/

	MT06 (Dev)				MT08 (Test)			
	deep		shallow		deep		shallow	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
Baseline	32.6	61.2	31.4	61.8	25.2	66.6	24.9	66.6
+ discrim. RO (src word)	32.9	61.3	31.6	61.8	25.4	66.3	25.2	66.6
+ discrim. RO (src+tgt word+class)	33.0	61.3	31.6	61.6	25.8	66.0	25.1	66.3
+ swap rule	32.8	61.7	31.8	62.1	25.8	66.6	25.0	67.0
+ discrim. RO (src word)	33.0	61.2	32.5	61.4	25.8	66.1	**26.0**	66.2
+ discrim. RO (src+tgt word+class)	33.1	61.2	32.6	61.4	**26.0**	66.1	**26.1**	66.3
+ binary swap feature	33.2	61.0	32.1	61.8	25.9	66.2	**25.7**	66.5
+ discrim. RO (src word)	33.1	61.3	32.4	61.4	**26.0**	66.1	**26.1**	66.3
+ discrim. RO (src+tgt word+class)	33.2	61.3	32.9	61.0	**26.2**	66.1	**26.1**	66.1
+ jump rules, variant 1	32.9	61.3	32.1	62.4	25.6	66.4	25.1	67.5
+ discrim. RO (src word)	32.9	61.1	31.9	62.0	25.8	66.0	25.1	66.9
+ discrim. RO (src+tgt word+class)	33.2	61.0	32.1	62.0	25.9	66.1	25.6	66.5
+ binary jump feature	32.8	61.3	31.9	61.7	25.7	66.3	25.2	66.7
+ discrim. RO (src word)	32.8	61.3	32.2	61.9	25.8	66.1	25.2	66.7
+ discrim. RO (src+tgt word+class)	33.1	61.2	32.3	62.0	**26.0**	66.1	25.5	66.7
+ jump rules, variant 2 + dist. feature	33.0	61.5	31.5	62.0	25.8	66.5	25.3	66.3
+ discrim. RO (src word)	33.2	60.8	31.6	61.9	**26.2**	65.8	25.2	66.4
+ discrim. RO (src+tgt word+class)	33.2	61.0	31.7	62.1	**26.4**	66.0	25.5	66.3

Table 1: Experimental results for the NIST Chinese→English translation task (truecase). On the test set, bold font indicates results that are significantly better than the baseline ($p < .1$).

jump rules variants.

We proceed with discussing some supplementary results obtained with the deep grammar that are not included in Table 1. The results for Sections 6.2.2 through 6.2.4 can be found in Table 2.

6.2.1 Dropping Length Constraints

In order to find out if we lose performance by applying the maximum length constraint of 10 to all non-terminals but the initial symbol S during decoding, we optimized systems with no length constraints. When we drop the length constraint in the baseline setup, we observe no improvement on the dev set and +0.3 %BLEU improvement on the test set. Dropping the length constraint in the system with deep grammar, swap rule, discrim. RO (src+tgt word+class) and binary jump feature results in +0.2 %BLEU / -0.2 %TER on the dev set, but no improvement on the test set.

6.2.2 Monotonic Concatenation Rule

In this experiment, we add a monotonic concatenation rule

$$X \to \langle X^{\sim 0} X^{\sim 1}, X^{\sim 0} X^{\sim 1} \rangle \qquad (8)$$

as discussed in Section 4.1.1 to the system with deep grammar, swap rule, binary swap feature and discrim. RO (src+tgt word+class). As we presumed, the monotonic concatenation rule does not improve the performance of our system.

6.2.3 Distance-Based Distortion Feature

Our second jump rules variant includes a distance-based distortion feature (*dist. feature*). To make sure that the good performance of the jump rules variant 2 extension compared to jump rules variant 1 is not simply due to this feature, we also tested it in the best setup with our first jump rules variant. Adding the distance-based distortion feature does not yield an improvement over that setup. We tried such a feature with the swap rule as well by just computing the length of the yield of the left-hand side non-terminal at each swap rule application. Here again, adding the distance-based distortion feature does not yield an improvement.

6.2.4 Discriminative Reordering for Reordering Rules Only

Instead of applying the discriminative reordering model at all rule applications, the model can as well be used to score the orientation of blocks only if they are placed side by side within the target sequence by selected rules. We conducted ex-

	deep			
	MT06 (Dev)		MT08 (Test)	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]
Baseline	32.6	61.2	25.2	66.6
+ no length contraints	32.6	61.5	25.5	66.6
+ swap rule + bin. swap feat. + discrim. RO (src+tgt word+class)	33.2	61.3	**26.2**	66.1
+ no length contraints	33.4	61.1	**26.2**	66.3
+ monotonic concatenation rule	33.2	61.6	**26.0**	66.4
+ dist. feature	33.4	61.4	**26.2**	66.2
+ discrim. RO scoring restricted to swap rule	33.1	61.4	**26.0**	66.4
+ jump rules 1 + bin. jump feat. + discrim. RO (src+tgt word+class)	33.1	61.2	**26.0**	66.1
+ dist. feature	33.2	61.1	25.9	66.1
+ discrim. RO scoring restricted to jump rules	32.8	61.3	25.9	66.3

Table 2: Supplementary experimental results with the deep grammar (truecase).

	deep		shallow	
	Baseline	Best Swap System	Baseline	Best Swap System
used hierarchical phrases	25.8%	32.0%	17.8%	24.0%
used lexical phrases	45.8%	40.0%	47.6%	44.7%
used initial and glue rules	28.4%	26.8%	34.6%	29.5%
used swap rules	-	1.2%	-	1.8%
applied swap rule in sentences	-	295 (22%)	-	446 (33%)

Table 3: Statistics on the rule usage for the single best translation of the test set (MT08).

periments in which the discriminative reordering scoring is restricted to the swap rule or the explicit jump rules (marked as † in Eq. 5), respectively. The result is in both setups slightly worse than the result with the discriminative reordering model applied to all rules.

6.3 Investigation of the Rule Usage

To figure out the influence of the swap rule on the usage of different types of rules in the translation process, we compare in Table 3 the baseline systems (deep and shallow) with the systems using the swap rule, binary swap feature and discrim. RO (denoted as *Best Swap System* in the table). As expected, the deep systems use in general more hierarchical phrases compared to the shallow setups. However, adding the swap rule causes an increased usage of hierarchical phrases and less applications of the glue rule. The swap rule by itself makes up the smallest part, but is employed in 22% (deep) and 33% (shallow) respectively of the 1357 test sentences.

6.4 Translation Examples

Figure 2 depicts a translation example along with its decoding tree from our system with deep grammar, swap rule, binary swap feature and discrim. RO (src+tgt word+class). The example is taken from the MT08 set, with the four reference translations *"But it is actually very hard to do that."*, *"However, it is indeed very difficult to achieve."*, *"But to achieve this point is actually very difficult."* and *"But to be truly frank is, in fact, very difficult."*. The hypothesis does not match any of the references, but still is a fully convincing English translation. Note how the application of the swap rule affects the translation. Our baseline system with deep grammar translates the sentence as *"but to do this , it is in fact very difficult ."*.

7 Conclusion

We presented novel extensions of hierarchical phrase-based systems with a discriminative lexicalized reordering model. We investigated combinations with three variants of additional non-lexicalized reordering rules. Our approach shows significant improvements (up to +1.2 %BLEU) over the respective baselines with both a deep and a shallow-1 hierarchical grammar on a large-scale Chinese→English translation task.

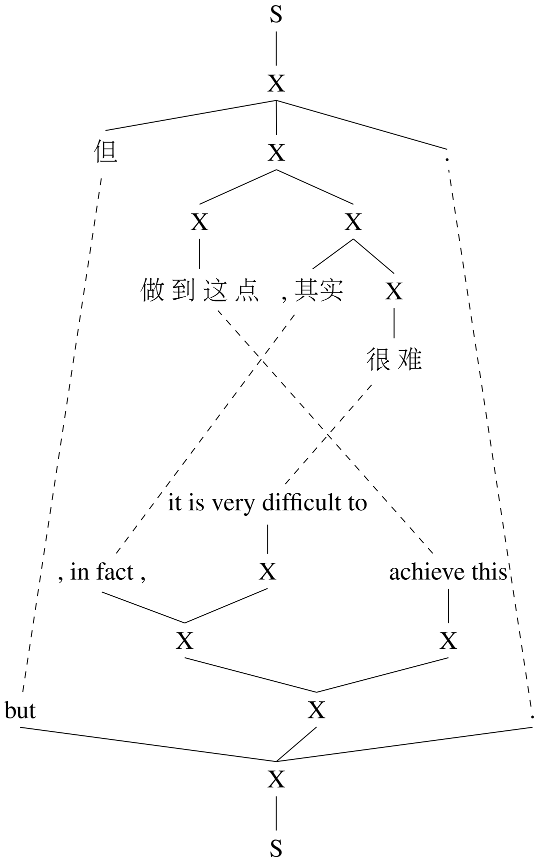

Figure 2: Translation example from the system with deep grammar, swap rule, binary swap feature and discrim. RO (src+tgt word+class).

Acknowledgments

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and partly funded by the European Union under the FP7 project T4ME Net, Contract No. 249119.

References

Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proc. of the ACL*, pages 263–270, Ann Arbor, MI, June.

Galley, M. and C. D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proc. of the EMNLP*, pages 847–855, Honolulu, Hawaii, October.

Gao, Y., P. Koehn, and A. Birch. 2011. Soft Dependency Constraints for Reordering in Hierarchical Phrase-Based Translation. In *Proc. of the EMNLP*, pages 857–868, Edinburgh, Scotland, UK, July.

Gispert, A. de, G. Iglesias, G. Blackwood, E. R. Banga, and W. Byrne. 2010. Hierarchical Phrase-Based Translation with Weighted Finite-State Transducers and Shallow-n Grammars. *Computational Linguistics*, 36(3):505–533.

He, Z., Y. Meng, and H. Yu. 2010a. Extending the Hierarchical Phrase Based Model with Maximum Entropy Based BTG. In *Proc. of the AMTA*, Denver, CO, October/November.

He, Z., Y. Meng, and H. Yu. 2010b. Maximum Entropy Based Phrase Reordering for Hierarchical Phrase-based Translation. In *Proc. of the EMNLP*, pages 555–563, October.

Huang, L. and D. Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proc. of the ACL*, pages 144–151, Prague, Czech Republic, June.

Huck, M., D. Vilar, D. Stein, and H. Ney. 2011. Advancements in Arabic-to-English Hierarchical Machine Translation. In *Proc. of the EAMT*, pages 273–280, Leuven, Belgium, May.

Iglesias, G., A. de Gispert, E. R. Banga, and W. Byrne. 2009. Rule Filtering by Pattern for Efficient Hierarchical Translation. In *Proc. of the EACL*, pages 380–388, Athens, Greece, March.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of the ACL*, pages 177–180, Prague, Czech Republic, June.

Leusch, G. and H. Ney. 2009. Edit distances with block movements and error rate confidence estimates. *Machine Translation*, December.

Och, F. J. and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Stolcke, A. 2002. SRILM – an Extensible Language Modeling Toolkit. In *Proc. of the ICSLP*, Denver, CO, September.

Tillmann, C. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *Proc. of the HLT-NAACL: Short Papers*, pages 101–104.

Vilar, D., D. Stein, M. Huck, and H. Ney. 2010. Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *Proc. of the ACL/WMT*, pages 262–270, Uppsala, Sweden, July.

Zens, R. and H. Ney. 2006. Discriminative Reordering Models for Statistical Machine Translation. In *Proc. of the HLT-NAACL*, pages 55–63, New York City, June.

Zens, R., H. Ney, T. Watanabe, and E. Sumita. 2004. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 205–211, Geneva, Switzerland, August.