A POS-Based Model for Long-Range Reorderings in SMT

Jan Niehues and Muntsin Kolss
Universität Karlsruhe
Karlsruhe, Germany
{jniehues,kolss}@ira.uka.de

Abstract

In this paper we describe a new approach to model long-range word reorderings in statistical machine translation (SMT). Until now, most SMT approaches are only able to model local reorderings. But even the word order of related languages like German and English can be very different. In recent years approaches that reorder the source sentence in a preprocessing step to better match target sentences according to POS(Part-of-Speech)-based rules have been applied successfully. We enhance this approach to model long-range reorderings by introducing discontinuous rules.

We tested this new approach on a German-English translation task and could significantly improve the translation quality, by up to 0.8 BLEU points, compared to a system which already uses continuous POS-based rules to model short-range reorderings.

1 Introduction

Statistical machine translation (SMT) is currently the most promising approach to machine translation of large vocabulary tasks. The approach was first presented by Brown et al. (1993) and has since been used in many translation systems (Wang and Waibel, 1998), (Och and Ney, 2000), (Yamada and Knight, 2000), (Vogel et al., 2003). State-of-the-art SMT systems often use translation models based on phrases to describe translation correspondences and word reordering between two languages. The reordering of words is one of the main difficulties in machine translation.

Phrase-based translation models by themselves have only limited capability to model different word orders in the source and target language, by capturing local reorderings within phrase pairs. In addition, the decoder can reorder phrases, subject to constraints such as confining reorderings to a relatively small window. In combination with a distance-based distortion model, some short-range reorderings can be handled. But for many language pairs this is not sufficient, and several authors have proposed additional reordering models as described in Section 2. In this work we present a new method that explicitly handles long-range word reorderings by applying discontinuous, POS-based reordering rules.

The paper is structured as follows: In the next section we present related work that was carried out in this area. Afterwards, we describe the problem of long-range reordering. In Section 4 the existing framework for reordering will be introduced. Section 5 describes the extraction of rules modeling long-range reorderings, and in the following section the integration into the framework will be explained. Finally, the model will be evaluated in Section 7, and a conclusion is given in Section 8.

2 Related Work

Several approaches have been proposed to address the problem of word reordering in SMT. Wu (1996) and Berger et al. (1996), for example, restrict the possible reorderings either during decoding time or during the alignment, but do not use any additional linguistic knowledge. A comparison of both methods can be found in Zens and Ney (2003).

Furthermore, techniques to use additional linguistic knowledge to improve the word order have been developed. Shen et al. (2004) and Och et al. (2004) presented approaches to re-rank the output of the decoder using syntactic information. Furthermore, lexical block-oriented reordering models have been developed in Tillmann and Zhang (2005) and Koehn et al. (2005). These models decide during decoding time for a given phrase, if

Proceedings of the 4th EACL Workshop on Statistical Machine Translation , pages 206–214,
Athens, Greece, 30 March – 31 March 2009.

the next phrase should be aligned to the left or to the right.

In recent years several approaches using reordering rules on the source side have been applied successfully in different systems. These rules can be used in rescoring as in Chen et al. (2006) or can be used in a preprocessing step. The aim of this step is to monotonize the source and target sentence. In Collins et al. (2005) and Popović and Ney (2006) hand-made rules were used to reorder the source side depending on information from a syntax tree or based on POS information. These rules had to be created manually, but only a few rules were needed and they were able to model long-range reorderings. Consequently, for every language pair these rules have to be created anew.

In contrast, other authors propose data-driven methods. In Costa-jussà and Fonollosa (2006) the source sentence is first translated into an auxiliary sentence, whose word order is similar to the one of the target sentences. Thereby statistical word classes were used. Rottmann and Vogel (2007),Zhang et al. (2007) and Crego and Habash (2008) used rules to reorder the source side and store different possible reorderings in a word lattice. They use POS tags and in the latter two cases also chunk tags to generalize the rules. The different reorderings are assigned weights depending on their relative frequencies (Rottmann and Vogel, 2007) or depending on a source side language model (Zhang et al., 2007).

In the presented work we will use discontinuous rules in addition to the rules used in Rottmann and Vogel (2007). This enables us to model long-range reorderings although we only need POS information and no chunk tags.

3 Long-Range Reorderings

One of the main problems when translating from German to English is the different word order in both languages. Although both languages are closely related, the word order is very different in some cases. Especially when translating the verb long-range reorderings have to be performed, since the position of the German verb is different from the one in the English sentence in many cases.

The finite verbs in the English language are always located at the second position, in the main clauses as well as in subordinate clauses. In German this is only true for the main clause. In contrast to that, in German subordinate clauses the verb (*glauben*) is at the final position as shown in Example 1.

Example 1: *..., die an den Markt und an die Gleichbehandlung aller glauben.*

... who believe in markets and equal treatment for all.

Example 2: *Das wird mit derart unterschiedlichen Mitgliedern unmöglich sein .*

That will be impossible with such disparate members.

A second difference in both languages is the position of the infinitive verb (*sein/be*) as shown in Example 2. In contrast to the English language, where it directly follows the finite verb, it is at the final position of the sentence in the German language.

The two examples show that in order to be able to handle the reorderings between German and English, the model has to allow some words to be shifted across the whole sentence. If this is not handled correctly, phrase-based systems sometimes generate translations that omit words, as will be shown in Section 7. This is especially problematic in the German-English case because the verb may be omitted, which carries the most important information of the sentence.

4 POS-Based Reordering

We will first briefly introduce the framework presented in Rottmann and Vogel (2007) since we extended it to also use discontinuous rules.

In this framework, the first step is to extract reordering rules. Therefore, an aligned parallel corpus and the POS tags of the source side are needed. For every sequence of source words where the target words are in a different order, a rule is extracted that describes how the source side has to be reordered to match the target side. A rule may for example look like this: *VVIMP VMFIN PPER → PPER VMFIN VVIMP*. The framework can handle rules that only depend on POS tags as well as rules that depend on POS tags and words. We will refer to these rules as short-range reordering rules.

The next step is to calculate the relative frequencies which are used as a score in the word lattice. The relative frequencies are calculated as the number of times the source side is reordered this way divided by the number of times the source side occurred in the corpus.

In a preprocessing step to the actual decoding, different reorderings of the source sentences are encoded in a word lattice. For all reordering rules that can be applied to the sentence, the resulting edge is added to the lattice if the score is better than a given threshold. If a reordering is generated by different rules, only the path of the reordering with the highest score is added to the lattice. Then, decoding is performed on the resulting word lattice.

5 Rule Extraction

To be able to handle long-range reorderings, we extract discontinuous reordering rules in addition to the continuous ones. The extracted rules should look, for example, like this: *VAFIN * VVPP* → *VAFIN VVPP **, where the placeholder "*" represents one or more arbitrary POS tags.

Compared to the continuous, short-range reordering rules described in the previous section, extracting such discontinuous rules presents an additional difficulty. Not only do we need to find reorderings and extract the corresponding rules, but we also have to decide which parts of the rule should be replaced by the placeholder. Since it is not always clear what is the best part to be replaced, we extract four different types of discontinuous rules. Then we decide during decoding which type of rules to use.

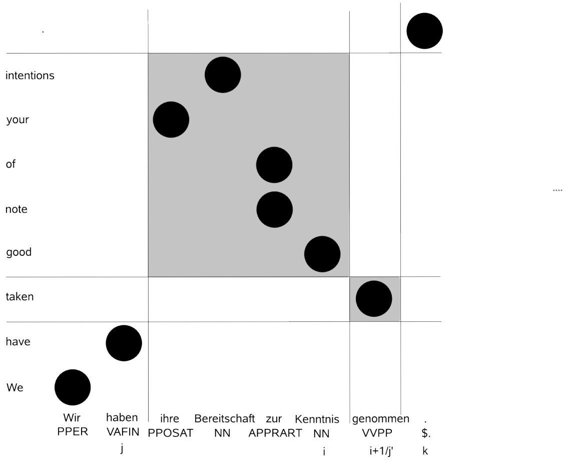
In a first step the reordering rule has to be found. Since this is done in a different way than for the continuous one, we will first describe it in detail. Like the continuous rules, the discontinuous ones are extracted from a word aligned corpus, whose source side is annotated with POS tags. Then the source side is scanned for reorderings. This is done by comparing the alignment points a_i and a_{i+1} of two consecutive words. We found a reordering if the target words aligned to f_i and f_{i+1} are in a different order than the source words. In our case the target word $e_{a_{i+1}}$ has to precede the target word e_{a_i}. More formally said, we check the following condition:

$$a_i > a_{i+1} \tag{1}$$

In Figure 1 an example with an automatically generated alignment is given. There, for example, a reordering can be found at the position of the word "*Kenntnis*".

Since we only check the links of consecutive words, we may miss some reorderings where there is an unaligned word between the words with a crossing link. However, in this case it is not clear where to place the unaligned word, so we do not extract rules from such a reordering.

Figure 1: Example training sentence used to extract reordering rules



So now we have found a reordering and also the border between the left and right part of the reordering. To be able to extract a rule for this reordering we need to find the beginning of the left and the end of the right part. This is done by searching for the last word before and the first word after the reordering. In the given example, the left part is "*ihre Bereitschaft zur Kenntnis*" and the right part would be "*genommen*". As shown in the figure, the words of the first part have to be aligned to target words that follow the target word aligned to the first word of the right part. Otherwise, they would not be part of the reordering. Consequently, to find the first word that is not part of the reordering, we search for the first word before the word f_{i+1} that is aligned to the word $e_{a_{i+1}}$ or to a target word before this word. More formally, we search for the word f_j that satisfies the following condition:

$$j = argmax_{l<i}\ a_l \leq a_{i+1} \tag{2}$$

The first word after the reordering is found in the same way. Formally, we search for the word f_k satisfying the condition:

$$k = argmax_{l>i+1}\ a_l \geq a_i \tag{3}$$

In our example, we now can extract the following reordering rule: *ihre Bereitschaft zur Kenntnis genommen* → *genommen ihre Bereitschaft zur Kenntnis*. In general, we will extract the rule: $f_{j+1} \dots f_i f_{i+1} \dots f_{k-1} \rightarrow f_{i+1} \dots f_{k-1} f_{j+1} \dots f_i$

An additional problem are unaligned words after f_j and before f_k. For these words it is not clear if they are part of the reordering or not. Therefore, we will include or exclude them depending on the type of rule we extract. To be able to write the rules in a easier way let $f_{j'}$ be the first word following f_j that is aligned and $f_{k'}$ the last word before f_k.

After extracting the reordering rule, we need to replace some parts of the rule by a placeholder to obtain more general rules. As described before, it is not directly clear which part of the rule should be replaced and therefore, we extract four different types of rules.

In the reordering, there is always a left part, in our example *ihre Bereitschaft zur Kenntnis*, and a right part (*genommen*). So we can either replace the left or the right part of the reordering by a placeholder. One could argue that always the longer sequence should be replaced, since that is more intuitive, but to lose no information we just extract both types of rules. Later we will see that depending on the language pair, one or the other type will generalize better. In the evaluation part the different types will be referred to as *Left* and *Right* rules.

Furthermore, not the whole part has to be replaced. It can be argued that the first or last word of the part is important to characterize the reordering and should therefore not be replaced. For each of the types described before, we extract two different sub-types of rules, which leads altogether to four different types of rules.

Let us first have a look at the types where we replace the left part. If we replace the whole part, in the example we would get the following rule: * *VVPP* → *VVPP* *. This would lead to problems during rule application. Since the rule begins with a placeholder, it is not clear where the matching should start. Therefore, we also include the last word before the reordering into the rule and can now extract the following rule from the sentence: *VAFIN* * *VVPP* → *VAFIN VVPP* *. In general, we extract the following rule to which we will refer as *Left All*:

$$f_j * f_{i+1} \dots f_{k'} \quad \rightarrow \quad f_j f_{i+1} \dots f_{k'} *$$

As mentioned in the beginning, we extracted a second sub-type of rule. This time, the first word of the left part is not replaced. The reason can be seen by looking at the reordered sequence. There, the second part of the reordering is moved between the last word before the reordering (f_j) and the first word of the first part (f_{j+1}). In our example this results in the following rule: *VAFIN PPOSAT* * *VVPP* → *VAFIN VVPP PPOSAT* * and in general, we extract the rule (*Left Part*):

$$f_j f_{j+1} * f_{i+1} \dots f_{k'} \quad \rightarrow \quad f_j f_{i+1} \dots f_{k'} f_{j+1} *$$

If we replace the right part by a star, we similarly get the following rule (*Right All*): *PPOSAT NN APPART NN* * → * *PPOSAT NN APPART NN*. The other rule (*Right Part*) can not be extracted from this example, since the right part has length one. But in general we get the two rules:

$$\begin{aligned} f_{j'} \dots f_i * f_{k-1} f_k \quad &\rightarrow \quad * f_{k-1} f_{j+1} \dots f_i f_k \\ f_{j'} \dots f_i * f_k \quad &\rightarrow \quad * f_{j'} \dots f_i f_k \end{aligned}$$

Here we already see that the rules where the first part is replaced result in typical reordering between the German and English language. The second part of the verb is at the end of the sentence in German, but in an English sentence it directly follows the first part.

6 Rule Application

During the training of the system all reordering rules are extracted from the parallel corpus in the way described in the last section. The rules are only used if they occur more often than a given threshold value. In the experiments a threshold of 5 is used.

The rules are scored in the same way as the continuous rules were. The relative frequencies are calculated as the number of times the rule was extracted divided by the number of times both parts occur in one sentence.

Then, in the preprocessing step, continuous rules as described in Section 4 and discontinuous rules are applied to the source sentence. As in the framework presented before, the rules are applied only to the source sentence and not to the lattice. Thus the rules cannot be applied recursively. For the discontinuous rules the "*" could match any sequence of POS tags, but it has to consist of at least one tag. If more than one rule can be applied to a sequence of POS tags and they generate different output, all edges are added to the lattice. If they generate the same sequence, only the rule with the highest probability is applied.

In initial experiments we observed that some rules can be applied very often to a sentence and therefore the lattice gets quite big. Therefore, we first check how often a rule can be applied to a sentence. If this exceeds a given threshold, we do not use this rule for this sentence. In these cases, the rule will most likely not find a good reordering, but randomly shuffle the words. In the experiments we use 5 as threshold, since this reduces the lattices to a decent size.

These restrictions limit the number of reorderings that have to be tested during decoding. But if all reorderings that can be generated by the remaining rules would be inserted into the lattice, the size of the lattice would still be too big to be able to do efficient decoding. Therefore, only rules with a probability greater than a given threshold are used to reorder the source sentence. Since the probabilities of the long-range reorderings are quite small compared to those of the short-range reorderings, we used two different thresholds.

7 Evaluation

We performed the experiments on the translation task of the WMT'08 evaluation. Most of the experiments were done on the German-English task, but in the end also some results on German-French and English-German are shown. The systems were trained on the European Parliament Proceedings (EPPS) and the News Commentary corpus. For the German-French task we used the intersection of the parallel corpora from the German-English and English-French task. The data was preprocessed and we applied compound splitting to the German corpus for the tasks translating from German. Afterwards, the word alignment was generated with the GIZA++-Toolkit and the alignments of the two directions were combined using the *grow-diag-final-and* heuristic. Then the phrase tables were created where we performed additional smoothing of the relative frequencies (Foster et al., 2006). Furthermore, the phrase table applied in the news task was adapted to this domain. In addition, a 4-gram language model was trained on both corpora. The rules were extracted using the POS tags generated by the TreeTagger (Schmid, 1994). In the end a beam-search decoder as described in Vogel (2003) was used to optimize the weights using the MER-training on the development sets provided for the different task by the workshop. The systems were tested on the test2007 set for the EPPS task and on the nc-test2007 testset for the news task. For test set translations the statistical significance of the results was tested using the bootstrap technique as described in Zhang and Vogel (2004).

Table 1: Evaluation of different Lattice sizes generated by changing the short-range threshold θ_{short} and long-range threshold θ_{long}

θ_{short}	θ_{long}	#Edges	Dev	Test
0.2	1	112K	24.57	27.25
0.1	1	203K	24.71	27.48
0.2	0.2	113K	24.70	27.51
0.2	0.1	121K	24.97	27.56
0.2	0.05	152K	25.28	27.80
0.1	0.1	212K	24.97	27.49
0.1	0.05	243K	25.12	27.81

7.1 Lattice Creation

In a first group of experiments we analyzed the influence of the two thresholds that determine the minimal probability of a rule that is used to insert the reordering into the lattice. The experiments were performed on the news task and used only the long-range rules generated by the *Part All* rules. The results are shown in Table 1 where θ_{short} is the threshold for the short-range reorderings and θ_{long} for the long-range reorderings. Consequently, only paths were added that are generated by a short-range reordering rule that has a probability greater than θ_{short} or paths generated by a long-range reordering rule with a minimum probability of θ_{long}. We used different thresholds for both groups of rules since the probabilities of long-range reorderings are in general lower.

The first two systems use no long-range reorderings. Adding the long-range reorderings does improve the translation quality and it makes sense to add even all edges generated by rules with a probability of at least 0.05. Using this system, less short-range reorderings are needed. The system using the thresholds of 0.2 and 0.05 has a performance nearly as good as the one using the thresholds 0.1 and 0.05, but it needs fewer edges. If long-range reordering is applied, fewer edges are needed than in the case of using only short-range reordering even though the translation quality is better. Therefore, we used the thresholds 0.2 and 0.05 in the following experiments.

Figure 2: Most common long-range reordering rules of type *Left Part*

NN ADV * VAFIN	→	NN VAFIN ADV *
VAFIN ART * VVPP	→	VAFIN VVPP ART *
ˆ ADV * PPER	→	ˆ PPER ADV *
$, ART * VVINF PTKZU	→	$, VVINF PTKZU ART *
PRELS ART * VVFIN	→	PRELS VVFIN ART *

Figure 3: Most common long-range reordering rules of type *Left All*

PRELS * VAFIN	→	PRELS VAFIN *
PRELS * VAFIN VVPP	→	PRELS VAFIN VVPP *
PPER * VMFIN	→	PPER VMFIN *
PRELS * VMFIN	→	PRELS VMFIN *
VMFIN * VAINF	→	VMFIN VAINF *

Table 2: Number of long-range reordering rules of different types used to create the lattices

Type	Left	Right
Part	8079	1127
All	2470	509
Both	9223	1405

Table 3: Translation results for the German-English task using different rule types (BLEU)

Type	EPPS		NEWS	
	Dev	Test	Dev	Test
Left Part	26.99	29.16	25.12	27.88
Right Part	26.69	28.73	24.76	27.28
Right/Left Part	26.99	28.96	25.06	27.69
Left All	26.77	28.76	24.37	26.56
Left Part/All	26.99	29.32	25.38	27.86
All	27.02	29.14	25.20	27.63

7.2 Rule Usage

We analyzed which long-range reordering rules were used to build the lattices. First, we compared the usage of the different types of rules. Therefore, we counted the number of rules that were applied to the development set of 2000 sentences if the thresholds 0.2 and 0.05 were used. The resulting numbers are shown in Table 2.

As it can be seen, the *Left* rules are more often used than the *Right* ones. This is what we expected, since when translating from German to English, the most important rules move the verb to the left. And these rules should be more general and therefore have a higher probability than the rules that move the words preceding the verb to the end of the sentence.

Next we analyzed which rules of the *Left Part* ones are used most frequently. The five most frequent rules are shown in Figure 2. The first, fourth and fifth rule moves the verb more to the front, as is often needed in English subordinate clauses. The second one moves both parts of the verb together.

The third most frequent rule moves personal pronouns to the front. In the English language the subject has to be always at the front. In contrast, in German the word order is not that strict and the subject can appear later.

We have done the same for the *Left All* rules. The rules are shown in Figure 3. In this type of rule the five most frequent rules all try to move the verb more to the front of the sentence. In the last case both parts of the verb are put together.

7.3 Rule Types

In a next group of experiments we evaluated the performance of the different rule types. In Table 3 the translation performance of systems using different rule types is shown. The experiments were carried out on the EPPS task as well as on the NEWS task.

First it can be seen that the *Left* rules perform better than the *Right* rules. This is not surprising, since they better describe how to reorder from German to English and because they are more often used in the lattice. If both types are used this

Table 4: Summary of translation results for the German-English tasks (BLEU)

System	EPPS		NEWS	
	Dev	Test	Dev	Test
Baseline	25.47	27.24	23.40	25.90
Short	26.77	28.54	24.73	27.48
Long	26.99	29.32	25.38	27.86

Table 5: Translation results for the German-French translation task (BLEU)

System	EPPS		NEWS	
	Dev	Test	Dev	Test
Baseline	25.86	27.05	17.90	18.52
Short	27.02	28.06	18.59	19.99
Long	27.27	28.61	19.10	20.11

lowers the performance a little. So if it is clear which type explains the reordering better, only this type should be used, but if that is not possible using both types can still help.

If both types of rules are compared, it can be seen that *Part* rules seem to have a more positive influence than *All* ones. The reason for this may be that the *Part* rules can also be applied more often than the rules of the other type. Using the combination of both types of rules, the performance is better on one task and equally good on the other task. Consequently, we used the combination of both types in the remaining experiments.

7.4 German-English

The results on the German-English task are summarized in Table 4. The long-range reorderings could improve the performance by 0.8 and 0.4 BLEU points on the different tasks compared to a system applying only short-range reorderings. These improvements are significant at a level of 5%.

We also analyzed the influence of tagging errors. Therefore, we tagged every word of the test sentence with the tag that this word is mostly assigned to in the training corpus. If the word does not occur in the training corpus, it was tagged as a noun. This results in different tags for 5% of the words and a BLEU score of 27.68 on the NEWS test set using long-range reorderings. So the translation quality drops by about 0.2 BLEU points, but it is still better than the system using only short-range reorderings.

In Figure 4 example translations of the baseline system, the system modeling only short-range reorderings and the system using also long-range reorderings rules are shown. The part of the sentences that needs long-range reorderings is always underlined.

In the first two examples the verbal phrase consists of two parts and the German one is splitted. In these cases, it was impossible for the short-range reordering model to move the second part of the verb to the front so that it could be translated correctly. In one case this leads to a selection of a phrase pair that removes the verb from the translation. Thus it is hard to understand the meaning of the sentence.

In the other two examples the verb of the subordinate clause has to be moved from the last position in the German sentence to the second position in the English one. This is again only possible using the long-range reordering rules. Furthermore, if these rules are not used, it is possible that the verb will be not translated at all as in the last example.

7.5 German-French

We also performed similar experiments on the German-French task. Since the type of reordering needed for this language pair is similar to the one used in the German-English task, we used also the *Left* rules in the long-range reorderings. As it can be seen in Table 5, the long-range reordering rules could also help to improve the translation performance for this language pair. The improvement on the EPPS task is significant at a level of 5%.

7.6 English-German

In a last group of experiments we applied the same approach also to the English-German translation task. In this case the verb has to be moved to the right, so that we used the *Right* rules for the long-range reorderings. Looking at the rule usage of the different type of rules, the picture was quite promising. This time the *Right* rules could be applied more often and the *Left* ones only a few times. But if we look at the results as shown in Table 6, the long-range reorderings do not improve the performance. We will investigate the reasons for this in future work.

Figure 4: Example translation from German to English using different type of rules

Source:	Diese Maßnahmen werden als eine Art Wiedergutmachung für früher begangenes Unrecht angesehen .
Baseline:	these measures will as a kind of compensation for once injustice done .
Short:	these measures will as a kind of compensation for once injustice done .
Long:	these measures will be seen as a kind of compensation for once injustice done .
Source:	Das wird mit derart unterschiedlichen Mitgliedern unmöglich sein .
Baseline:	this will with such different impossible .
Short:	this will with such different impossible .
Long:	this will be impossible to such different members .
Source:	Er braucht die Unterstützung derer , die an den Markt und an die Gleichbehandlung aller glauben .
Baseline:	he needs the support of those who market and the equal treatment of all believe .
Short:	it needs the support of those who in the market and the equal treatment of all believe .
Long:	it needs the support of those who believe in the market and the equal treatment of all .
Source:	.., daß sie das Einwanderungsproblem als politischen Hebel benutzen .
Baseline:	.. that they the immigration problem as a political lever .
Short:	.. that the problem of immigration as a political lever .
Long:	.. that they use the immigration problem as a political lever .

Table 6: Translation results for the English-German translation task (BLEU)

System	EPPS		NEWS	
	Dev	Test	Dev	Test
Baseline	18.93	2072	16.31	17.91
Short	19.49	21.56	17.13	18.31
Long	19.56	21.33	16.93	18.15

8 Conclusion

We have presented a new method to model long-range reorderings in statistical machine translation. This method extends a framework based on extracting POS-based reordering rules from an aligned parallel corpus by adding discontinuous reordering rules. Allowing rules with gaps captures very long-range reorderings while avoiding the data sparseness problem of very long continuous reordering rules.

The extracted rules are used to generate a word lattice with different possible reorderings of the source sentence in a preprocessing step prior to decoding. Placing various restrictions on the application of the rules keeps the lattice small enough for efficient decoding. Compared to a baseline system that only uses continuous reordering rules, applying additional discontinuous rules improved the translation performance on a German-English translation task significantly by up to 0.8 BLEU points.

In contrast to approaches like Collins et al. (2005) and Popović and Ney (2006), the rules are created in a data-driven way and not manually. It was therefore easily possible to transfer this approach to the German-French translation task, and we showed that we could improve the translation quality for this language pair as well. Furthermore, this approach needs only the POS information and no syntax tree. Thus, if we use the approximation for the tags as described before, the approach could also easily be integrated into a real-time translation system.

An unsolved problem is still why this approach does not improve the results of the English-German translation task. An explanation might be that here the reordering problem is even more difficult, since the German word order is very free.

Acknowledgments

This work was partly supported by Quaero Programme, funded by OSEO, French State agency for innovation.

References

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Boxing Chen, Mauro Cettolo, and Marcello Federico. 2006. Reordering Rules for Phrase-based Statistical Machine Translation. In *International Workshop on Spoken Language Translation (IWSLT 2006)*, Kyoto, Japan.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause Restructuring for Statistical Machine Translation. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 531–540.

Marta R. Costa-jussà and José A. R. Fonollosa. 2006. Statistical Machine Reordering. In *Conference on Empirical Methods on Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Nizar Crego and Nizar Habash. 2008. Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, Columbus, Ohio, USA.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable Smoothing for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Philipp Koehn, Amittai Axelrod, Alexandra B. Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *IWSLT*, Pittsburgh, PA, USA.

Franz Josef Och and Herman Ney. 2000. Improved Statistical Alignment Models. In *38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, Hong Kong.

Franz J. Och, Daniel Gildea, Sanjeev P. Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David A. Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir R. Radev. 2004. A Smorgasboard of Features for Statistical Machine Translation. In *Human Language Technology Conference and the 5th Meeting of the North American Association for Computational Linguistics (HLT-NAACL 2004)*, Boston, USA.

Maja Popović and Hermann Ney. 2006. POS-based Word Reorderings for Statistical Machine Translation. In *International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Kay Rottmann and Stephan Vogel. 2007. Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model. In *TMI*, Skövde, Sweden.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

Libin Shen, Anoop Sarkar, and Franz Och. 2004. Discriminative Reranking for Machine Translation. In *Human Language Technology Conference and the 5th Meeting of the North American Association for Computational Linguistics (HLT-NAACL 2004)*, Boston, USA.

Christoph Tillmann and Tong Zhang. 2005. A Localized Prediction Model for Statistical Machine Translation. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan, USA.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU Statistical Translation System. In *MT Summit IX*, New Orleans, LA, USA.

Stephan Vogel. 2003. SMT Decoder Dissected: Word Reordering. In *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China.

Yeyi Wang and Alex Waibel. 1998. Fast Decoding for Statistical Machine Translation. In *ICSLP'98*, Sydney, Australia.

Dekai Wu. 1996. A Polynomial-time Algorithm for Statistical Machine Translation. In *ACL-96: 34th Annual Meeting of the Assoc. for Computational Linguistics*, Santa Cruz, CA, USA, June.

Kenji Yamada and Kevin Knight. 2000. A Syntax-based Statistical Translation Model. In *38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, Hong Kong.

Richard Zens and Hermann Ney. 2003. A Comparative Study on Reordering Constraints in Statistical Machine Translation. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 192–202, Sapporo, Japan.

Ying Zhang and Stephan Vogel. 2004. Measuring Confidence Intervals for mt Evaluation Metrics. In *TMI 2004*, Baltimore, MD, USA.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *HLT-NAACL Workshop on Syntax and Structure in Statistical Translation*, Rochester, NY, USA.