

Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation

Teresa Herrmann, Jan Niehues, Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
Karlsruhe, Germany

{teresa.herrmann, jan.niehues, alexander.waibel}@kit.edu

Abstract

We describe a novel approach to combining lexicalized, POS-based and syntactic tree-based word reordering in a phrase-based machine translation system. Our results show that each of the presented reordering methods leads to improved translation quality on its own. The strengths however can be combined to achieve further improvements. We present experiments on German-English and German-French translation. We report improvements of 0.7 BLEU points by adding tree-based and lexicalized reordering. Up to 1.1 BLEU points can be gained by POS and tree-based reordering over a baseline with lexicalized reordering. A human analysis, comparing subjective translation quality as well as a detailed error analysis show the impact of our presented tree-based rules in terms of improved sentence quality and reduction of errors related to missing verbs and verb positions.

1 Introduction

One of the main difficulties in statistical machine translation (SMT) is presented by the different word orders between languages. Most state-of-the-art phrase-based SMT systems handle it within phrase pairs or during decoding by allowing words to be swapped while translation hypotheses are generated. An additional reordering model might be included in the log-linear model of translation. However, these methods can cover reorderings only over a very limited distance. Recently, reordering as preprocessing has drawn much attention. The idea is to detach the reordering problem from the decoding process and

to apply a reordering model prior to translation in order to facilitate a monotone translation.

Encouraged by the improvements that can be achieved with part-of-speech (POS) reordering rules (Niehues and Kolss, 2009; Rottmann and Vogel, 2007), we apply such rules on a different linguistic level. We abstract from the words in the sentence and learn reordering rules based on syntactic constituents in the source language sentence. Syntactic parse trees represent the sentence structure and show the relations between constituents in the sentence. Relying on syntactic constituents instead of POS tags should help to model the reordering task more reliably, since sentence constituents are moved as whole blocks of words, thus keeping the sentence structure intact.

In addition, we combine the POS-based and syntactic tree-based reordering models and also add a lexicalized reordering model, which is used in many state-of-the-art phrase-based SMT systems nowadays.

2 Related Work

The problem of word reordering has been addressed by several approaches over the last years.

In a phrase-based SMT system reordering can be achieved during decoding by allowing swaps of words within a defined window. Lexicalized reordering models (Koehn et al., 2005; Tillmann, 2004) include information about the orientation of adjacent phrases that is learned during phrase extraction. This reordering method, which affects the scoring of translation hypotheses but does not generate new reorderings, is used e.g. in the open source ma-

chine translation system Moses (Koehn et al., 2007).

Syntax-based (Yamada and Knight, 2001) or syntax-augmented (Zollmann and Venugopal, 2006) MT systems address the reordering problem by embedding syntactic analysis in the decoding process. Hierarchical MT systems (Chiang, 2005) construct a syntactic hierarchy during decoding, which is independent of linguistic categories.

To our best knowledge Xia and McCord (2004) were the first to model the word reordering problem as a preprocessing step. They automatically learn reordering rules for English-French translation from source and target language dependency trees. Afterwards, many followed these footsteps. Earlier approaches craft reordering rules manually based on syntactic or dependency parse trees or POS tags designed for particular languages (Collins et al., 2005; Popović and Ney, 2006; Habash, 2007; Wang et al., 2007). Later there were more and more approaches using data-driven methods. Costa-jussà and Fonollosa (2006) frame the word reordering problem as a translation task and use word class information to translate the original source sentence into a reordered source sentence that can be translated more easily. A very popular approach is to automatically learn reordering rules based on POS tags or syntactic chunks (Popović and Ney, 2006; Rottmann and Vogel, 2007; Zhang et al., 2007; Crego and Habash, 2008). Khalilov et al. (2009) present reordering rules learned from source and target side syntax trees. More recently, Genzel (2010) proposed to automatically learn reordering rules from IBM1 alignments and source side dependency trees. In DeNero and Uszkoreit (2011) no parser is needed, but the sentence structure used for learning the reordering model is induced automatically from a parallel corpus. Among these approaches most are able to cover short-range reorderings and some store reordering variants in a word lattice leaving the selection of the path to the decoder. Long-range reorderings are addressed by manual rules (Collins et al., 2005) or using automatically learned rules (Niehues and Kolss, 2009).

Motivated by the POS-based reordering models in Niehues and Kolss (2009) and Rottmann and Vogel (2007), we present a reordering model based on the syntactic structure of the source sentence. We intend to cover both short-range and long-range re-

ordering more reliably by abstracting to constituents extracted from syntactic parse trees instead of working only with morphosyntactic information on the word level. Furthermore, we combine POS-based and tree-based models and additionally include a lexicalized reordering model. Altogether we apply word reordering on three different levels: lexicalized reordering model on the word level, POS-based reordering on the morphosyntactic level and syntax tree-based reordering on the constituent level. In contrast to previous work we use original syntactic parse trees instead of binarized parse trees or dependency trees. Furthermore, our goal is to address especially long-range reorderings involving verb constructions.

3 Motivation

When translating from German to English different word order is the most prominent problem. Especially the verb needs to be shifted over long distances in the sentence, since the position of the verb differs in German and English sentences. The finite verbs in the English language are generally located at the second position in the sentence. In German this is only the case in a main clause. In German subordinate clauses the verb is at the final position as shown in Example 1.

Example 1:

Source: ..., nachdem ich eine Weile im Internet gesucht habe.
Gloss: ... after I a while in-the internet searched have.
POS Reord.: ..., nachdem ich habe eine Weile im Internet gesucht.
POS Transl.: ... as I have for some time on the Internet.

The example shows first the source sentence and an English gloss. **POS Reord** presents the reordered source sentence as produced by POS rules. This should be the source sentence according to target language word order. **POS Transl** shows the translation of the reordered sequence. We can see that some cases remain unresolved. The POS rules succeed in putting the auxiliary *habe/have* to the right position in the sentence. But the participle, carrying the main meaning of the sentence, is not shifted together with the auxiliary. During translation it is

dropped from the sentence, rendering it unintelligible.

A reason why the POS rules do not shift both parts of the verb might be that the rules operate on the word level only and treat every POS tag independently of the others. A reordering model based on syntactic constituents can help with this. Additional information about the syntactic structure of the sentence allows to identify which words belong together and should not be separated, but shifted as a whole block. Abstracting from the word level to the constituent level also provides the advantage that even though reorderings are performed over long sentence spans, the rules consist of less reordering units (constituents which themselves consist of constituents or words) and can be learned more reliably.

4 Tree-based Reordering

In order to encourage linguistically meaningful reorderings we learn rules based on syntactic tree constituents. While the POS-based rules are flat and perform the reordering on a sequence of words, the tree-based rules operate on subtrees in the parse tree as shown in Figure 1.

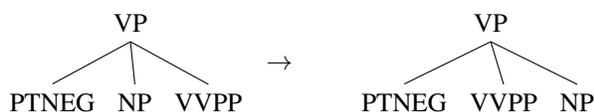


Figure 1: Example reordering rule based on subtrees

A syntactic parse tree contains both the word-level categories, i.e. parts-of-speech and higher order categories, i.e. constituents. In this way it provides information about the building blocks of a sentence that belong together and should not be taken apart by reordering. Consequently, the tree-based reordering operates both on the word level and on the constituent level to make use of all available information in the parse tree. It is able to handle long-range reorderings as well as short-range reorderings, depending on how many words the reordered constituents cover. The tree-based reordering rules should also be more stable and introduce less random word shuffling than the POS-based rules.

The reordering model consists of two stages. First the rule extraction, where the rules are learned by searching the training corpus for crossing alignments which indicate a reordering between source

and target language. The second is the application of the learned reordering rules to the input text prior to translation.

4.1 Rule Extraction

As shown in Figure 4 we learn rules like this:

$VP\ PTNEG\ NP\ VVPP \rightarrow VP\ PTNEG\ VVPP\ NP$

where the first item in the rule is the head node of the subtree and the rest represent the children. In the second part of the rule the children are indexed so that children of the same category cannot be confused. Figure 2 shows an example for rule extraction: a sentence in its syntactic parse tree representation, the sentence in the target language and an automatically generated alignment. A reordering occurs between the constituents *VVPP* and *NP*.

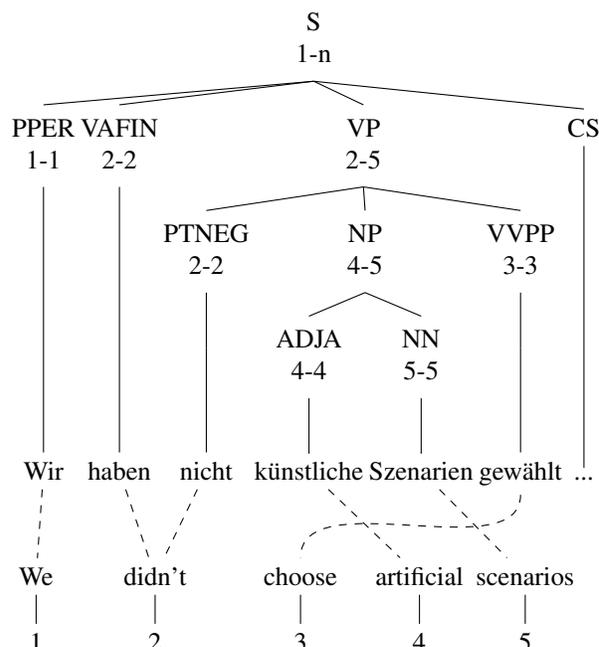


Figure 2: Example training sentence used to extract reordering rules

In a first step the reordering rule has to be found. We extract the rules from a word aligned corpus where a syntactic parse tree is provided for each source side sentence. We traverse the tree top down and scan each subtree for reorderings, i.e. crossings of alignment links between source and target sentence. If there is a reordering, we extract a rule that rearranges the source side constituents according to the order of the corresponding words on

the target side. Each constituent in a subtree comprises one or more words. We determine the lowest (*min*) and highest (*max*) alignment point for each constituent c_k and thus determine the range of the constituent on the target side. This can be formalized as $\min(c_k) = \min\{j | f_i \in c_k; a_i = j\}$ and $\max(c_k) = \max\{j | f_i \in c_k; a_i = j\}$. To illustrate the process, we have annotated the parse tree in Figure 2 with the alignment points (*min-max*) for each constituent.

After defining the range, we check for the following conditions in order to determine whether to extract a reordering rule.

1. all constituents have a non-empty range
2. source and target word order differ

First, for each subtree at least one word in each constituent needs to be aligned. Otherwise it is not possible to determine a conclusive order. Second, we check whether there is actually a reordering, i.e. the target language words are not in the same order as the constituents in the source language: $\min(c_k) > \min(c_{k+1})$ and $\max(c_k) > \max(c_{k+1})$.

Once we find a reordering rule to extract, we calculate the probability of this rule as the relative frequency with which such a reordering occurred in all subtrees of the training corpus divided by the number of total occurrences of this subtree in the corpus. We only store rules for reorderings that occur more than 5 times in the corpus.

4.1.1 Partial Rules

The syntactic parse trees of German sentences are quite flat, i.e. a subtree usually has many children. When a rule is extracted, it always consists of the head of the subtree and all its children. The application requires that the applicable rule matches the complete subtree: the head and all its children. However, most of the time only some of the children are actually involved in a reordering. There are also many different subtree variants that are quite similar. In verb phrases or noun phrases, for example, modifiers such as prepositional phrases or adverbial phrases can be added nearly arbitrarily. In order to generalize the tree-based reordering rules, we extend the rule extraction. We do not only extract the rules from the complete child sequence, but also from any continuous child sequence in a constituent.

This way, we extract generalized rules which can be applied more often. Formally, for each subtree $h \rightarrow c_1^n = c_1 c_2 \dots c_n$ that matches the constraints presented in Section 4.1, we modify the basic rule extraction to: $\forall i, j 1 \leq i < j \leq n : h \rightarrow c_i^j$. It could be argued that the partial rules might be not as reliable as the specific rules. In Section 6 we will show that such generalizations are meaningful and can have a positive effect on the translation quality.

4.2 Rule Application

During the training of the system all reordering rules are extracted from the parallel corpus. Prior to translation the rules are applied to the original source text. Each rule is applied independently producing a reordering variant of that sentence. The original sentence and all reordering variants are stored in a word lattice which is later used as input to the decoder. The rules may be applied recursively to already reordered paths. If more than one rule can be applied, all paths are added to the lattice unless the rules generate the same output. In this case only the rule with the highest probability is applied.

The edges in a word lattice for one sentence are assigned transition probabilities as follows. In the monotone path with original word order all transition probabilities are initially set to 1. In a reordered path the first branching transition is assigned the probability of the rule that generated the path. All other transition probabilities in this path are set to 1. Whenever a reordered path branches from the monotone path, the probability of the branching edge is subtracted from the probability of the monotone edge. However, a minimum probability of 0.05 is reserved for the monotone edge. The score of the complete path is computed as the product of the transition probabilities. During decoding the best path is searched for by including the score for the current path weighted by the weight for the reordering model in the log-linear model. In order to enable efficient decoding we limit the lattice size by only applying rules with a probability higher than a pre-defined threshold.

4.2.1 Recursive Rule Application

As mentioned above, the tree-based rules may be applied recursively. That means, after one rule is applied to the source sentence, a reordered path may

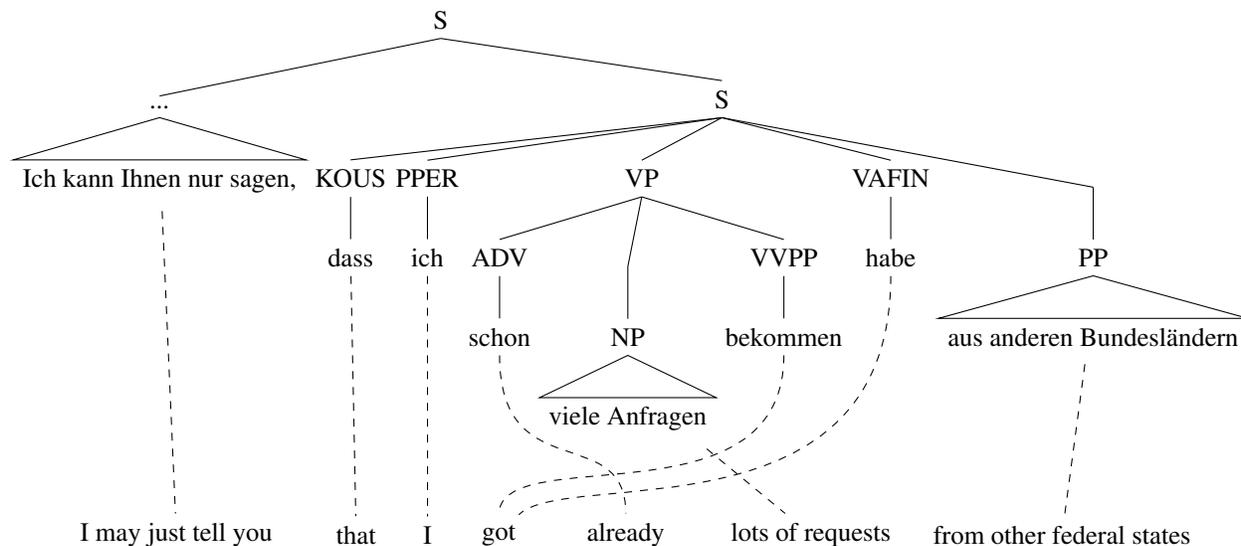


Figure 3: Example parse tree with separated verb particles

be reordered again. The reason is the structure of the syntactic parse trees. Verbs and their particles are typically not located within the same subtree. Hence, they cannot be covered by one reordering rule. A separate rule is extracted for each subtree. Figure 3 demonstrates this in an example. The two parts that belong to the verb in this German sentence, namely *bekommen* and *habe*, are not located within the same constituent. The finite verb *habe* forms a constituent of its own and the participle *bekommen* forms part of the VP constituent. In English the finite verb and the participle need to be placed next to each other. In order to rearrange the source language words according to the target language word order, the following two reordering movements need to be performed: the finite verb *habe* needs to be placed before the VP constituent and the participle *bekommen* needs to be moved within the VP constituent to the first position. Only if both movements are performed, the right word order can be generated.

However, the reordering model only considers one subtree at a time when extracting reordering rules. In this case two rules are learned, but if they are applied to the source sentence separately, they will end up in separate paths in the word lattice. The decoder then has to choose which path to translate: the one where the finite verb is placed before the VP constituent **or** the path where the participle is at the first position in the VP constituent.

To counter this drawback the rules may be applied

recursively to the new paths created by our reordering rules. We use the same rules, but newly created paths are fed back into the queue of sentences to be reordered. However, we only apply the rules to parts of the reordered sentence that are still in the original word order and restrict the recursion depth.

5 Combining reordering methods

In order to get a deeper insight into their individual strengths we compare the reordering methods on different linguistic levels and also combine them to investigate whether gains can be increased. We address the word level using the lexicalized reordering, the morphosyntactic level by POS-based reordering and the constituent level by tree-based reordering.

5.1 POS-based and tree-based rules

The training of the POS-based reordering is performed as described in (Rottmann and Vogel, 2007) for short-range reordering rules, such as $VVIMP\ VMFIN\ PPER \rightarrow PPER\ VMFIN\ VVIMP$. Long-range reordering rules trained according to (Niehues and Kolss, 2009) include gaps matching longer spans of arbitrary POS sequences ($VAFIN\ * \ VVPP \rightarrow VAFIN\ VVPP\ *$). The POS-based reordering used in our experiments always includes both short and long-range rules.

The tree-based rules are trained separately as described above. First the POS-based rules are applied to the monotone path of the source sentence and then

the tree-based rules are applied independently, producing separate paths.

5.2 Rule-based and lexicalized reordering

As described in Section 4.2 we create word lattices that encode the reordering variants. The lexicalized reordering model stores for each phrase pair the probabilities for possible reordering orientations at the incoming and outgoing phrase boundaries: monotone, swap and discontinuous. In order to apply the lexicalized reordering model on lattices the original position of each word is stored in the lattice. While the translation hypothesis is generated, the reordering orientation with respect to the original position of the words is checked at each phrase boundary. The probability for the respective orientation is included as an additional score.

6 Results

The tree-based models are applied for German-English and German-French translation. Results are measured in case-sensitive BLEU (Papineni et al., 2002).

6.1 General System Description

First we describe the general system architecture which underlies all the systems used later on. We use a phrase-based decoder (Vogel, 2003) that takes word lattices as input. Optimization is performed using MERT with respect to BLEU. All POS-based or tree-based systems apply monotone translation only. Baseline systems without reordering rules use a distance-based reordering model. In addition, a lexicalized reordering model as described in (Koehn et al., 2005) is applied where indicated. POS tags and parse trees are generated using the Tree Tagger (Schmid, 1994) and the Stanford Parser (Rafferty and Manning, 2008).

6.1.1 Data

The German-English system is trained on the provided data of the WMT 2012. news-test2010 and news-test2011 are used for development and testing. The type of data used for training, development and testing the German-French system is similar to WMT data, except that 2 references are available. The training corpus for the reordering models consist of the word-aligned Europarl and News Commentary corpora where POS tags and parse trees are

generated for the source side.

6.2 German-English

We built systems using POS-based and tree-based reordering and show the impact of the individual models as well as their combination on the translation quality. The results are presented in Table 1.

For each system, two different setups were evaluated. First, with a distance-based reordering model only (noLexRM) and with an additional lexicalized reordering model (LexRM). The baseline system which uses no reordering rules at all allows a reordering window of 5 in the decoder for both setups. For all systems where reordering rules are applied, monotone translation is performed. Since the rules take over the main reordering job, only monotone translation is necessary from the reordered word lattice input. In this experiment, we compare the tree-based rules with and without recursion, and the partial rules.

Rule Type \ System	noLexRM		LexRM	
	Dev	Test	Dev	Test
Baseline (no Rules)	22.82	21.06	23.54	21.61
POS	24.33	21.98	24.42	22.15
Tree	24.01	21.92	24.24	22.01
Tree rec.	24.37	21.97	24.53	22.19
Tree rec.+ par.	24.31	22.21	24.65	22.27
POS + Tree	24.57	22.21	24.91	22.47
POS + Tree rec.	24.61	22.39	24.81	22.45
POS + Tree rec.+ par.	24.80	22.45	24.78	22.70

Table 1: *German-English*

Compared to the baseline system using distance-based reordering only, 1.4 BLEU points can be gained by applying combined POS and tree-based reordering. The tree rules including partial rules and recursive application alone achieve already a better performance than the POS rules, but using them all in combination leads to an improvement of 0.4 BLEU points over the POS-based reordering alone. When lexicalized reordering is added, the relative improvements are similar: 1.1 BLEU points compared to the Baseline and 0.55 BLEU points over the POS-based reordering. We can therefore argue that the individual rule types as well as the lexicalized reordering model seem to address complementary reordering issues and can be combined successfully to

obtain an even better translation quality.

We applied only tree rules with a probability of 0.1 and higher. Partial rules require a threshold of 0.4 to be applied, since they are less reliable. In order to prevent the lattices from growing too large, the recursive rule application is restricted to a maximum recursion depth of 3. These values were set according to the results of preliminary experiments investigating the impact of the rule probabilities on the translation quality. Normal rules and partial rules are not mixed during recursive application.

With the best system we performed a final experiment on the official testset of the WMT 2012 and achieved a score of 23.73 which is 0.4 BLEU points better than the best constrained submission.

6.3 Translation Examples

Example 2 shows how the translation of the sentence presented above is improved by adding the tree-based rules. We can see that using tree constituents in the reordering model indeed addresses the problem of verb particles and especially missing verb parts in German.

Example 2:

Src: ..., *nachdem ich eine Weile im Internet gesucht habe.*

Gloss: ..., *after I a while in-the Internet searched have.*

POS: ... *as I have for some time on the Internet.*

+Tree: ... *after I have looked for a while on the Internet.*

Example 3 shows another aspect of how the tree-based rules work. With the help of the tree-based reordering rules, it is possible to relocate the separated prefix of German verbs and find the correct translation. The verb *vorschlagen* consist of the main verb (MV) *schlagen* (here conjugated as *schlägt*) and the prefix (PX) *vor*. Depending on the verb form and sentence type, the prefix must be separated from the main verb and is located in a different part of the sentence. The two parts of the verb can also have individual meanings. Although the translation of the verb stem were correct if it were the full verb, not recognizing the separated prefix and ignoring it in translation, corrupts the meaning of the sentence. With the help of the tree-based rules, the dependency

between the main verb and its prefix is resolved and the correct translation can be chosen.

6.4 German-French

The same experiments were tested on German-French translation. For this language pair, similar improvements could be achieved by combining POS and tree-based reordering rules and applying a lexicalized reordering model in addition. Table 2 shows the results. Up to 0.7 BLEU points could be gained by adding tree rules and another 0.1 by lexicalized reordering.

Rule Type \ System	noLexRM		LexRM	
	Dev	Test	Dev	Test
POS	41.29	38.07	42.04	38.55
POS + Tree	41.94	38.47	42.44	38.57
POS + Tree rec.	42.35	38.66	42.80	38.71
POS + Tree rec.+ par.	42.48	38.79	42.87	38.88

Table 2: *German-French*

6.5 Binarized Syntactic Trees

Even though related work using syntactic parse trees in SMT for reordering purposes (Jiang et al., 2010) have reported an advantage of binarized parse trees over standard parse trees, our model did not benefit from binarized parse trees. It seems that the flat hierarchical structure of standard parse trees enables our reordering model to learn the order of the constituents most efficiently.

7 Human Evaluation

7.1 Sentence-based comparison

In order to have an additional perspective of the impact of our tree-based reordering, we also provide a human evaluation of the translation output of the German-English system without the lexicalized reordering model. 250 translation hypotheses were selected to be annotated. For each input sentence two translations generated by different systems were presented, one applying POS-based reordering only and the other one applying both POS-based and tree-based reordering rules. The hypotheses were anonymized and presented in random order.

Table 3 shows the BLEU scores of the analyzed systems and the manual judgement of comparative, subjective translation quality. In 50.8% of the sen-

Example 3:

Src: *Die RPG Byty schlägt ihnen in den Schreiben eine Mieterhöhung von ca. 15 bis 38 Prozent vor.*

Gloss: *The RPG Byty proposes-MV them in the letters a rent increase of ca. 15 to 38 percent proposes-PX*

POS: *The RPG Byty beats them in the letter, a rental increase of around 15 to 38 percent.*

+Tree: *The RPG Byty proposes them in the letters a rental increase of around 15 to 38 percent.*

System	BLEU	wins	%
POS Rules	21.98	58	23.2
POS + Tree Rules rec. par.	22.45	127	50.8

Table 3: Human Evaluation of Translation quality

Type	all	exist	position	form
Improvements	48	22	21	5
Degradations	16	2	11	3

Table 4: Manual Analysis of verbs

tences, the translation generated by the system using tree-based rules was judged to be better, whereas in 23.2% of the cases the system without tree-based rules was rated better. For 26% of the sentences the translation quality was very similar. Consequently, in 76.8% of the cases the tree-based system produced a translation that is either better or of the same quality as the system using POS rules only.

7.2 Analysis of verbs

Since the verbs in German-to-English translation were one of the issues that the tree-based reordering model should address, a more detailed analysis was performed on the first 165 sentences. We especially investigated the changes regarding the verbs between the translations stemming from the system using the POS-based reordering only and the one using both the POS and the tree-based model. We examined three aspects of the verbs in the two translations. Each change introduced by the tree-based reordering model was first classified either as an improvement or a degradation of the translation quality. Secondly, it was assigned to one of the following categories: **exist**, **position** or **form**. In case of an improvement, **exist** means a verb existed in the translation due to the tree-based model, which did not exist before. A degradation in this category means that a verb was removed from the translation when including the tree-based reordering model. An improvement or degradation in the category **position** or **form** means that the verb position or verb form was improved or degraded, respectively.

Table 4 shows the results of this analysis. In total, 48 of the verb changes were identified as improvements, while only 16 were regarded as degradations of translation quality. Improvements mainly concern

improved verb position in the sentence and verbs that could be translated with the help of the tree-based rules that were not there before. Even though also degradations were introduced by the tree-based reordering model, the improvements outweigh them.

8 Conclusion

We have presented a reordering method based on syntactic tree constituents to model long-range reorderings in SMT more reliably. Furthermore, we combined the reordering methods addressing different linguistic abstraction levels. Experiments on German-English and German-French translation showed that the best translation quality could be achieved by combining POS-based and tree-based rules. Adding a lexicalized reordering model increased the translation quality even further. In total we could reach up to 0.7 BLEU points of improvement by adding tree-based and lexicalized reordering compared to only POS-based rules. Up to 1.1 BLEU points were gained over to a baseline system using a lexicalized reordering model.

A human evaluation showed a preference of the POS+Tree-based reordering method in most cases. A detailed analysis of the verbs in the translation outputs revealed that the tree-based reordering model indeed addresses verb constructions and improves the translation of German verbs.

Acknowledgments

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

References

- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA.
- Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of ACL 2005*, Ann Arbor, Michigan.
- 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.
- Josep M. Crego and Nizar Habash. 2008. Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT. In *ACL-HLT 2008*, Columbus, Ohio, USA.
- John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of EMNLP 2011*, pages 193–203, Edinburgh, Scotland, UK.
- Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of COLING 2010*, Beijing, China.
- Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. *Proceedings of the 11th MT Summit*.
- Jie Jiang, Jinhua Du, and Andy Way. 2010. Improved phrase-based smt with syntactic reordering patterns learned from lattice scoring. In *Proceedings of AMTA 2010*, Denver, CO, USA.
- M. Khalilov, J.A.R. Fonollosa, and M. Dras. 2009. A new subtree-transfer approach to syntax-based reordering for statistical machine translation. In *Proc. of the 13th Annual Conference of the European Association for Machine Translation (EAMT'09)*, pages 198–204, Barcelona, Spain.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual meeting-association for computational linguistics*, volume 45, page 2.
- Jan Niehues and Muntsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Division, T. J. Watson Research Center.
- 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.
- Anna N. Rafferty and Christopher D. Manning. 2008. Parsing three German treebanks: lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, Columbus, Ohio.
- 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.
- Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 101–104.
- Stephan Vogel. 2003. SMT Decoder Dissected: Word Reordering. In *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China.
- Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 737–745.
- Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of COLING 2004*, Geneva, Switzerland.
- Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, Stroudsburg, PA, 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.
- Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 138–141, Stroudsburg, PA, USA.