SMT: "Reordering Word Problem" and Solutions

Alok Parlikar

11-734: Advanced MT Seminar; Spring 2008

Abstract

Because different languages employ different word orders in their syntax, one requirement of an MT system is to get the target words in the right order. While phrase based MT systems do very well at reordering inside short windows of words, long-distance reordering seems to be a challenging task. One way of overcoming this challenge is to use linguistic information and reorder the input sentence so that the word order is consistent with what the target language might expect, and then decode this reordered input using a phrase-based MT system. In this paper, we would look at a few approaches that use this idea of fixing the word order as a 'preprocessing' step. Specifically, we will look at approaches that use Part-of-Speech information, and parse trees as the linguistic input. We will see that although at first glance this approach might sound counter-intuitive, it works very well. Towards the end, we will also briefly discuss some future ideas further in this direction.

Contents

PART 1

Malaise in Foreign Langue

The problem introduced. . .

> The greatest masterpiece in literature is only a dictionary that is out of order.
>
> JEAN COCTEAU

When a string of text has to be translated from one language into another, the implicit objective of the process is to capture the meaning of that string, and to rewrite the meaning in words that speakers of the target language would understand. Bilingual speakers who engage in translation tasks usually follow this understand-and-rewrite approach. However, because computers' skills at natural language understanding and generation are currently very limited, a somewhat different strategy has to be used for machine translation. One approach to machine translation, called 'Phrase-based Statistical Machine Translation' (SMT) deals with the translation problem by making three assumptions. First, SMT assumes that there exist independent units of meaning, either single words or phrases[1], and that these units line up together to express a composite meaning. The second assumption that SMT makes is that if we translate these individual units from one language into another, the translated units can be lined up together to express the same meaning as in the original text. Finally, SMT believes that translated units of meaning can not necessarily be put together in the same original order—they may need to be reshuffled. It should be noted that these assumptions are not infallible. The assumption that composite meaning can be built up from units of meanings tends to fall apart when text consists of long idioms and proverbs. Nonetheless, SMT systems perform very well under most circumstances, and hence these assumptions can be deemed reasonable.

SMT systems have achieved state-of-the-art performance. These systems process a lot of training data and build a table that provides a mapping between phrases in the source language to phrases in the target language, and scores of how likely the mapping would be. This table gives SMT systems a great advantage with respect to word choices. In fact, phrases inside the table also account for short-distance reshuffling of words between the source and target languages. For example, an entry in the table could map the words 'black_i cake_j' from English to 'gateux_j noir_i' in French, thereby taking into account a local reordering. However, SMT systems have potential limitations when it comes to reorderings of words that happen at longer distances. If the source and target languages differ significantly in how words

[1] Here, a phrase means a group of words, not necessarily syntactic constituents.

are to be sequenced in a sentence, the output of SMT systems looks quite garbled, and this calls for a better methodology for SMT systems to deal with the word order problem. In the following section, we shall look at how SMT systems typically deal with the issue of getting the word order in the target language right. We will then look at divergences between some language pairs, and review some options for handling them.

1.1 Reordering and Phrase-based SMT

One trivial solution to handle reordering in SMT is to allow for all possible reorderings among the translated phrases and then to choose the best one. However, if arbitrary reorderings are allowed, the search problem of the decoder has been classified NP-complete[Kni99]. If we constrain the search path to be monotone, the search is possible to be done in polynomial time. SMT systems typically use two types of constraints to narrow down the search space during reordering.

Many SMT systems use a distance-based reordering model. This model assigns a penalty to every reordering, and the penalty increases as the reordering distance increases. Thus, a longer reordering is preferred only if other models in the decoder give it a good score. Another model that SMT systems use is called a lexicalized reordering model. Here, reorderings are scored on the basis of distance of words being reordered, as seen in the training data.

The search space is still huge, and hence the systems usually specify a value of the maximum window size within which reordering could take place. That restricts the capabilities of SMT to get the word order right if the source and target languages are radically different in word order behavior.

1.2 Word Order Divergences Between Languages

One interesting ground for comparing languages is the word order that they use, and what the word order encodes. A very shallow analysis of word order involves figuring out where the subject, object and the verb occur in the sentences. Based on this, languages could be classified as SVO (English), SOV (Hindi)[DPB01], VSO (Arabic)[YK01], etc. Some languages, such as Russian allow a free word order[AO90]. This means that the word order doesnot convey information about subject and object, but instead conveys something different—possibly old and new information. As we look at the differences in word order in more details, things start to become more and more complex, and languages tend to stand out distinctly against one other. These deeper differences pose challenges to SMT because as sentences get longer in length, they are no longer simple enough to contain a subject, object and a verb, but are complex constructions made up of several sentential components. Getting the word order right thus gets increasingly difficult without linguistic information at hand—information that current SMT systems don't necessarily have.

In the papers that we shall discuss in this report, researchers have used a few language pairs for translation, and have tried to achieve the correct word order. Let us quickly take a look at divergences in word orders of these language pairs, so that we will be better equipped to discuss the performance of methods that the researchers have proposed.

French and Spanish differ from English in the orderings of adjectives and nouns. While in English adjectives precede nouns (such as *black* cake), most adjectives in French and Spanish succede the nouns that they qualify, (such as gateux *noir*.)

Long-distance reordering is observed between English and German. Infinitives and past participles occur at the end of clauses in German, while they usually occur towards the beginning of the clauses in English. Similarly, German has detatched verbal prefixes that are placed at end of clauses. For example, a sentence such as "I *will arrive* tomorrow afternoon..." would be translated into German as "Ich *werde* morgen nachmittag ... *ankommen*". Since the length of a clause could be, in theory, unbounded, the word reordering between English and German could have to be done at any arbitrarily long distance.

In Vietnamese, WH-word movement is signifi-

cantly different than in English. The interrogative word is not moved to the beginning of the sentence. Also, unlike in English, most Vietnamese yes-no questions end in an interrogative word. Another differentiating factor between English and Vietnamese is that Vietnamese phrases are head-initial. Thus, what in English would be "his friend's *book*," would be "*book* 's friend his" in Vietnamese.

1.3 Providing Linguistic Information to SMT

By looking at the divergences between different languages, it is evident that linguistic information is essential if we have to get the target words in the right order. However, using linguistic knowledge within an SMT decoder requires it to be significantly modified. Certainly, there are other MT approaches that are built around the core idea of using linguistic information1073134,974747,zollmann06syntax. However, in this report, we shall focus on how phrase based SMT can gain in performance by using linguistics.

One proposed solution is to use linguistic information to preprocess all input text to an SMT decoder. Specifically, the approaches that we shall look at in this report use the following method. Given an input sentence that we have to translate, let us first reorder the words of that sentence, so that the order would correspond to what the target language expects. For example, if we were to translate the sentence "He eats an apple" into Hindi, we would transform this string as "He an apple eats," because Hindi, as we saw, uses an SOV word order. This transformed input would then be given to an SMT decoder. The decoder may or may not be allowed to do any further reordering. Thus, the output text would have the required word order, and would express the right meaning if the SMT system is well trained.

This report will survey a handful of approches to do the outlined pre-processing of text in the source language. Although it might sound simple at first thought, designing a preprocessing-stage reordering model requires us to answer three critical questions:

- How do we model the reordering from the source language to the target language? Do we learn it automatically from data?
- How do we assign scores to different reorderings?
- How do we apply the reordering model at runtime?

In the course of this report, we will look at different proposed approaches and implicitly or explicitly answer these questions for each one of them.

PART 2

Part-of-Speech based Word-Reordering

'Shallower' solution to the problem...

> You can explain almost all grammar from just eight parts of speech
>
> JANE BELL KIESTER

Parts of speech (POS) provide a lot of information about word order. In English, for example, we know that a determiner must always precede a noun. In every language, parts of speech can be believed to follow a set of rules. In fact, these rules can be put together hierarchically to obtain a grammar for parsing sentences. As it turns out, assigning POS tags is much easier a problem than generating parses, hence a POS-based approach can be very useful. POS tags also tend to be much more accurate than parse structures, because there is lesser ambiguity to deal with. In this part of the report, we will look at how only POS information can be used to define rules that allow us to reorder source-language text and make it closely resemble the target-language word order.

2.1 Manually Written Rules

Earlier, we saw that French and Spanish differ from English in how the adjectives and nouns are relatively ordered. This information can be very easily transcribed into a rule that uses POS to reorder the noun and adjectives during translation. Popović and Ney[PN06] used exactly this rule in their experiments. The experiments actually involved three languages: English, German and Spanish. These languages allowed them to study both local and long-distance reorderings. They used the European Parliament corpus of about 700,000 sentences for the experiment. In order to investigate sparse training scenarios, they also performed experiments on 1% of the original corpus. English and Spanish were the source languages. All three languages were used as target languages. The rules used for performing the reordering were as follows:

- While translating from Spanish to English/German: Move adjectives before the noun group.
- While translating from English/German to Spanish: Move adjectives after the noun group.
- When translating from Spanish/English to German: Move the infinitive or past participle to the end of the clause, while keeping the auxiliary verb in the original position.

The experimental setup used the RWTH SMT decoder. Two versions of the decoder were used. In one case, the decoder was trained on original data, as obtained from the corpus. In the other case, the reordering rules were applied to the entire corpus, and the reordered corpus was used to train the decoder.

The reordering rules resulted in an increased translation accuracy. While translation from Spanish to English, the best results obtained were from the decoder trained on reordered corpus. An improvement of 2.3 points was seen on test data over a baseline BLEU score of 19.7 points. In this case, however, the training corpus was of a small size (7000 sentences). With the full sized training corpus, the improvements diminished to only 1.3 BLEU points. While translating in the reverse direction, from English to Spanish, the improvements were smaller. The improvement in BLEU score for a system trained on the full sized corpus was only 0.5 points. The explanation for this asymmetry in improvement is that in Spanish, most adjectives come after the noun, but some exceptional ones must come before it. In English, all adjectives must precede the noun that they qualify. Clearly, a reordering rule from Spanish to English produces accurate results all the time, whereas a reordering rule from English to Spanish tends to do the wrong thing every so often. While translating from English to German, it was observed that the decoder that was trained on reordered corpus performed 1.9 BLEU points worse that the system trained on original corpus. Unlike in the case of local reordering that we just saw, best improvements of 1.1 BLEU points were obtained when the full sized training corpus was used. Experiments for translating Spanish into German were interesting because both local and long-distance reordering rules could be applied at the same time in this case. The combination of the two reordering rules had an additive nature to the improvements in BLEU score for system trained on the full sized corpus. This improvement was much smaller as compared to results between English and German. Allowing both types of reordering rules to be applied gave an improvement of 0.3 BLEU points over a baseline of 21.2.

2.2 Automatically Extracted Rules

In the previous section, we saw how only a couple of reordering rules can help improve translation quality. In this section, we will look at two POS based approaches that tried to learn reordering rules from data.

Crego and Mariño[Co06] presented an approach to use word-to-word alignments of bilingual data to learn POS based reordering rules. The main proceduce consists of identifying all crossings produced in the word to word alignments. Once a crossing has been detected, the source-side POS tags and alignments are used to account for a new instance of reordering pattern. The target side of the pattern is computed using the original order of the target words to which the source words are aligned. Figure 2.1 shows a clarifying example of pattern extraction.

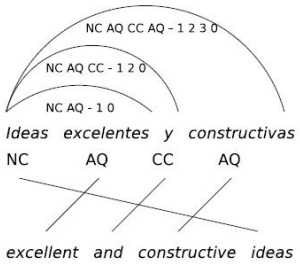

Figure 2.1: Extracting Reordering Patterns[Co06]

The monotone search path of the decoder is extended with reorderings following the patterns found in training. The procedure identifies the sequences of words in the input sentence that match any available pattern. Then, for each match, we add an arc into the search graph (encoding the reordering learnt in the pattern) unless a translation unit with the same source-side words is already available. Once the search graph is built, the decoder traverses it looking for the best translation. Hence, the winner hypothesis is computed using all the available information (all the SMT models).

The experimental setup used the language pairs English and Spanish. The Europarl corpus (1.28M sentences) was used for the training. An n-gram

based decoder called MARIE was used. Word alignments were obtained by running Giza++ upto IBM Model 4.

A huge number of reordering patterns were extracted using the method just outlined. It was seen that most patterns were erroneous, because of errors in word alignment. The patterns were filtered down. Maximum difference in number of words in the source and target side of a pattern was set to 4. Similarly, the source side could consist of at most 8 words. Patterns seen less than 1000 times were filtered out. Patterns with score less than 0.2, where score was computed as the number of occurrences of pattern divided by number of occurrences of pattern source words were filtered out. After patterns were filtered, only 29 of them remained. Unfortunately, it was discovered that some patterns were still erroneous.

The final set of reordering patterns was first used to reorder the entire training corpus and build a source-side language model. For each input sentence to be translated, it was POS-tagged. All applicable reordering patterns were considered one by one, and all valid search paths were condensed into a search graph. The BLEU score results of these experiments on the test data are shown in Table 2.1. The 'rgraph' system uses the reordering patterns only. The 'pos' system uses both the reordering patterns and the source-side language model. It can be seen that even if the extracted patterns had errors, the system showed good improvements.

System	Sp-En	En-Sp
base	52.9	48.1
rgraph	53.3	49.0
pos	53.9	49.1

Table 2.1: POS-based Reordering Results[Co06]

In order to better assess the rules extracted, human evaluation was performed. Reordering patterns were classified as good or bad, and the bad ones were filtered out. However, results of using these newly filtered rules have not been reported.

Although reordering rules work very well in general, on closer inspection, we see that rules are applied erroneously at many places. Probably, some rules can be applied only under some circumstances, or only under the presence or absence of some POS tags nearby where the rule is being applied. Context information has been shown to be useful during POS tagging[BM04], so we might expect it to also help us choose which reordering rules to apply and when.

Rottmann and Vogel studied the usefulness of context information[RV07]. Their approach was similar to Crego and Mariño. Word alignments and POS tags of a bilingual corpus were used to extract reordering rules. Unlike in the previous method, they stored context information along with the rule. Context included 1-2 tags on the left and right of the rule being extracted. A rule that was observed to occur within a longer rule was kept only if it occurred as the longest reordering in some other sentence pair. Rules that occurred less than 5 times in the corpus were filtered out, and remaining rules were scored using their relative frequency.

Before decoding, input sentences were processed very much the same way as in the previous approach. Sentences were POS tagged. For every applicable rule with matching context, a new search path was created, and later, all search paths were compressed into a search lattice. Here, edges of the lattices were also assigned probabilities, based on the rule scores, so that the decoder could prefer one search path over the other.

Rottmann and Vogel ran experiments with English, Spanish and German languages. The Europarl corpus (about 33M words) was used for training. Rules were extracted from the training data. They allowed long rules, of length upto 15, in order to account for very long distance reorderings.

The results of translation experiments are shown in Table 2.2. We observe that context information does help a lot while translating from English to Spanish, and from English to German, although not so much while translating from German to English. However, the absolute improvements in BLEU scores are rather low for English to Spanish translation. There are reportedly two reasons for these low improvements: Translation from English to Spanish is already very good, and because most reorderings are local, they are already captured in the phrase table.

More importantly, these experiments were performed with the decoder trained on original corpus, not a reordered one, and thus many potential phrases in the source sentence did not match to any entry in the phrase table.

System	en→ es	en → de	de → en
Baseline	48.51	17.69	23.70
No Context	49.52	17.78	24.79
All Rules	49.58	18.27	24.85

Table 2.2: Translation BLEU Scores[RV07]

In the next experiments, we shall see the effect of training the decoder based on the reordered corpus. There could be two approaches to reordering the training corpus. First, we could use the giza word alignments and simply use this information to get the source side of the corpus in monotone order. In the second approach, we could use the giza alignments and POS tags of the training corpus, and use the extracted rules to get the most probable reordering for the given source sentence. The results of these experiments are shown in Table 2.3. It turns out that using just the GIZA word alignments does not help a lot in all situations, probably because the phrases that the decoder is trained with do not match the situation at decoding time, since the input sentences are reordered using rules.

System	en→ es	en → de	de → en
Baseline	48.51	17.69	23.70
GIZA-RO	49.78	18.23	24.09
Rule-RO	49.75	18.42	25.06

Table 2.3: Translation BLEU Scores with decoder trained on reordered corpus[RV07]

We thus see that having context information can help improving translation accuracy, although the results are different for different language pairs. Further, system trained on a reordered corpus tends to perform better than otherwise, for reasons relating to phrase-matching during decoding. It would also be interesting to what happens if in addition to the reordering lattice as input, the decoder is allowed to perform short-distance reordering. We expect the decoding time to go up, and so also the translation accuracy, but these results are not available for the experiments that we saw in this part.

2.3 Summary

We saw three approaches to using POS information to do the reordering of input text before translation. Reordering rules could be written manually, or could be learned from data. Rules could also specify the context in which they should be applied. Multiple rules could apply for a given POS sequence. Building a search lattice allows us to have all possible reorderings searched by the decoder. The lattice itself could be annotated with edge scores, so that the decoder prefers some paths over others during the search. The decoder could be trained on either the original corpus, or a corpus reordered using the rules extracted from itself, and systems tend to perform better when the decoder is trained on reordered corpus. In all cases, the results of applying reordering as a preprocessing step are better that otherwise, although the magnitude and significance of improvements depends very much on the language pair itself. Improvements in translating between English and German assert that POS based reordering approach can handle long-distance reorderings well.

PART 3

Syntax based Word-Reordering

'Deeper' solution to the problem...

> The rules of syntax can look inside a sentence or phrase and cut and paste the smaller phrases inside it.
>
> Steven Arthur Pinker

Parts of speech do provide linguistic information about the words that we look at, but they hide away a lot of the big picture. Words in a sentence are connected along many dimensions, such as syntax, semantics and pragmatics. Rules of syntax can build tree structures starting with part of speech information. These trees can provide considerably deeper insight into how sentences are organized. Unlike parts of speech, grammar rules that build these trees can let us do long distance reorderings in a very efficient way. In the simplest example, changing from an SVO word order to SOV word order involves learning only one rule, that reorders the verb and Noun Phrase that make up the predicate of the sentence. In terms of parts of speech, this one rule could be seen in data as multiple different rules, thereby increasing the variance in training data significantly. Clearly, we would expect that if source-language text is reordered using rules of syntax, we could achieve better results. In this part of the report, we will look at a handful of approaches that use grammar rules and parse trees to generate reordered source text that can then be translated using a phrase based SMT decoder.

3.1 Sequence of Reordering Rules

Let us consider that someone gave us a context-free representation of language grammar. We would have productions such as $A \rightarrow BC$ in the grammar. Writing a reordering rule would mean, that we change the grammar to accommodate for reorderings. Thus, our rule might change to $A \rightarrow CB$ when reordering is to take place. There rules are called rewrite patterns, or simply reordering rules. Rules are sometimes lexicalized—instead of having all non-terminal symbols in the right hand side of a rule, some of them can be replaced by the ultimate terminal symbol, or word, that they derive. Doing this helps us capture some context information. Rules can also be annotated, and they could tell where the syntactic head lies on the right hand side. Sometimes this information can be helpful.

One approach for learning and applying rules has been proposed by Xia and McCord[XM04]. They used Slot grammar parsers to parse the English-French Canadian Hansard Corpus (90M words). They then used word alignment information to align the phrases between parallel parse trees. If S is a source phrase, and T is a target phrase, then for ev-

ery S, T they assign a score, as given by:

$$V(S,T) = \frac{\text{links}(S,T)}{\text{Span}(S) + \text{Span}(T)}.$$

Finally, for every source phrase S, the target phrase that gave maximum score was selected to be aligned to. Once the aligned phrases and thus aligned nodes in the tree are known, the approach suggests finding out if children of those nodes align to each other, and specifically if the head word on source side aligns to head word on target side. If two nodes align to each other, and also if their children align among themselves, a rewrite pattern can be extracted based on the relative ordering of children. One constraint that this approach put up was that nodes that have more than 5 children will not be included as candidates for rewrite patterns. Using this approach, Xia and McCord extracted 56,000 rewrite patterns. When now given input text, they first parsed it, then traversed the parse tree and applied the most specific pattern applicable at each node. At the end of the traversal, they had the tree reordered, and by extracting the leaf nodes, they obtained the reordered sentence to be fed to the decoder. They did two versions of the experiment. In one case, the decoder was restricted to a monotone search. In the other case, local reordering was enabled inside the decoder. The results of their experiments in both these cases is shown in Table 3.1.

System	Monotone	Non-monotone
No RO	19.6	18.7
RO	21.5	18.5

Table 3.1: Translation BLEU scores[XM04]

Similar experiment was done by Collins and others for German-English language pair[CKK05]. In their system however, rules were not learned automatically from training data, but instead they were manually crafted. The rule set consisted of 6 transformations that handled phenomena with respect to German-English verb reordering. The corpus that they used to train their SMT system was the Europarl corpus (750,000 sentences). Using the reordering rules that they had written gave an improvement in BLEU scores from 25.2 to 26.8. To see how this improvement really correlates to improvement in translation accuracy as judged by humans, they did a subjective evaluation. The results showed that after using the reordering rules, 33 sentences out of randomly chosen 100 sentences showed an improvement in quality, whereas 13 sentences became worse.

3.2 More than one Reorderings

In the previous section, we saw that when when reordering rules were applied to a tree, only the final sentential reordering was considered as an input to the decoder. We know that reorderings rules may not always do the right thing. Sometimes, the target language does not expect words to be reordered where the rule might do so. Thus, we need an approach that takes into account several possible reorderings. Such an approach was proposed by Nguyen and Shimazu[NS06]. In their approach, rules were learned from corpora, and assigned scores. Based on these scores, application of rules to input text before decoding was also done statistically.

Given a CFG rule, there could be multiple ways to reorder it. Lexicalization of rules can help us decide which reordering should be applied. For example, Vietnamese has different word orders for 'a nice weather' and 'this nice weather', and which reordering to apply can be determined by having a lexical context in the rule. However, lexicalization can lead to too many rules, and score estimation would become a problem. Nguyen and Shimazu proposed using a Lexicalized Probabilistic Context Free Grammar (LPCFG) to assign rule scores.

In this approach, they first start with parsing the training text, and getting word alignments on it using GIZA. Using a strategy similar to Xia, McCord as we saw earlier, they align the source-side phrases. In case words have one-to-many alignments, one link is chosen based on these heuristics: (1) If source span is one word, choose the best link based on intersection of bidirectional alignments and lexical scores. (2) For each word outside source phrase, there should be no link to any word outside the target phrase, and vice versa.

For each node in the source tree, and based on the target phrase position of its children, a reordering rule is learned. Finally, all the rules are scored as follows:

$$p(LHS \rightarrow RHS|LHS \rightarrow RHS') = \frac{n(LHS \rightarrow RHS|LHS \rightarrow RHS')}{n(LHS \rightarrow RHS')}.$$

After the rules and their scores were learned from the corpus, Nguyen, Shimazu applied them to input text. They parsed the source sentence. This parse tree was then lexicalized by propogating all head-words bottom up. At each node now, several different reorderings can be applied. The way to choose the final reordering is as follows:

$$Q^* = \{RS^* : RS_i^* = \text{argmax}\,[P(L_i \rightarrow R_i|L_i \rightarrow R_i') * P(L_i \rightarrow R_i')]\}$$

where RS denotes a sequerce of Rules. L and R denote the LHS and RHS of a grammar rule, and R′ denotes the reordered right hand side. After choosing this rule set, one ultimate reordering is generated, and this is reflected in the surface string extracted from the parse tree.

In their experiment, Nguyen and Shimazu worked with English, French and Vietnamese languages. English and French allowed studying effects of local reordering, whereas Vietnamese allowed for studying longer distance reorderings. For extracting reordering rules, they used 40,000 parsed sentences. The English to Vietnamese translation was done on two corpora, called 'Computer' and 'Conversation', while the English-French experiments were done on the Europarl corpus. The results that they obtained are shown in Table 3.2.

In their experiments, they used the Pharaoh decoder. While training the SMT system, they tried various values for length of the longest phrases that the decoder would store. We mentioned before, that SMT relies on longer phrases to get some reorderings. The results were consistent with this idea. Although both the baseline Pharaoh system and the system that was using reordering had performance improvements as the pharaoh phrase-length increased, the improvements of the reordered system over the baseline decreased.

Language	Baseline	RO
En-Vi (1)	45.12	47.62
En-Vi (2)	33.85	36.26
En-Fr	26.41	28.02

Table 3.2: Translation BLEU scores with two corpora for English to Vietnamese and one corpus for English to French.[NS06]

One interesting factor that their experiments showed was that using reordering in SMT does not have the property of 'vanishing improvement'. That is to say, even if we train the SMT system on more and more data, using reordering rules consistently gives improvements over the baseline. In their experiment, they tried training sets of sizes ranging from 10K to 94K, and although the baseline score increased with increasing training data, an improvement of more than 2 BLEU points was obtained in each case.

One issue that remains to be solved in this approach is that although all possible reorderings are considering while making the decision to choose one, the decoder is still fed with one reordered input per sentence. It is quite possible that the best reordering that grammar rules would suggest may not be the one that models inside the decoder pick. Since translation quality is of ultimate importance here, we need an approach that would allow us to submit all possible reorderings to the decoder, and ask the decoder to choose the best one. Li and others have recently reported their work in this direction[LLZ+07]. What they propose is that given a sentence, we can obtain an n-best list of the reordered sentence. Each of these reorderings can be translated by a decoder, and then the best translation can be chosen. They also proposed giving up using rewrite patterns and instead learning reordering knowledge.

Consider that $A \rightarrow BC$ is a node in the source tree. Using word alignment information, we can determine the target positions that B occupies, and the target positions that C occupies. Ideally, these spans of B and C on the target side should not overlap. If they

do overlap, the proposal is to remove weak scoring word alignments until the spans no longer overlap. A similar strategy can be used when the grammar production is not binary. Once these spans are known, we can easily decide whether B and C should be reordered or not. We could now score the reorderings using relative frequency. Now, the proposal is to use Maximum entropy modelling to learn the reordering patterns. In the case of binary rules, the Maxent model does binary classification of whether the children of a node must be reordered or not. The suggested features for training the Maxent model are: Leftmost and rightmost word of a phrase and their POS, Head word of a phrase and its POS, Context words of the phrase (1 to the left and right) and their POS.

Once the Maxent model is trained with the reordering patterns, we can apply the reordering knowledge to new input text. For this, the group has proposed using a bottom up approach. If the node that we are at is a unary production, a score of 1 is assigned to it. Otherwise, the maxent model would be used to determine which reordering rule would be applicable at the node, and its rule-score would be obtained. The value of the current node would be set as product of this rule score and the values of the node's children. While traversing the tree, the idea is to keep track of the N-highest probabilities of nodes that we have seen. These probabilities correspond to the reorderings that must be used in generating the N-best list of reordered sentences.

Instead of operating at a sentence level, the group split sentences at clause boundaries in their experiments. They used the parse tree to identify the nodes that correspond to inflectional phrases, and used them to define clauses. For every clause that they thus split from the sentence, they generate an N-best list of reordered clauses. Each of these reorderings is translated to generate an N-best list of translation hypothesis per clause. While doing this, the decoder is provided with additional feature, which is the probability of the reordering, as estimated before. Also, the decoding is allowed to be non-monotone because local reorderings could have been pruned out in generating the N-best list. The best translation for each clause, as scored by the decoder, was used to reconstruct a single sentence back. This was returned as the final output for the sentence.

In their experimental setup, Li et al. used Pharaoh-like decoder that was trained on the Gigaword Corpus. They used the MT-05 data for testing. First, they demonstrated that clause splitting does not degrade the translation performance. When a standard phrase based SMT would give a BLEU score of 29.22, using clause splitting gave a score of 29.13. They also showed that using maxent models outperform simple rule-based models. Their results are shown in Table 3.3. From these results, we can see that more lexical features help improving scores, but combining phrasal and lexical features hurts performance.

	Setting	BLEU
1	rule	29.77
2	ME (phrase label)	29.93
3	ME (left, right)	30.10
4	ME (3 + head)	30.24
5	ME (3 + phrase label)	30.12
6	ME (4 + context)	30.24

Table 3.3: Translation Results using N-best RO approach[LLZ+07]

3.3 Summary

In this part of the report, we saw that we can use parse trees on the source side to extract reordering rules. These rules can take care of reorderings much effectively than what POS based rules can. We saw that using N-best list instead of single best reordering helps, and that splitting sentences into clauses at reasonable boundaries (such as IP nodes) does not degrade performance. We also saw that performance does not have 'vanishing effect' if SMT system is trained on large data. We also saw that we could use the parse trees to carry out long distance reorderings and allow the decoder to efficiently take care of local reorderings.

PART 4

Conclusions

From the present 'tense', to the future 'perfect'.

> A conclusion is the place where one gets tired of writing
>
> Based on a quote by Arthur Bloch

In this report, we saw that phrase-based SMT has challenges to get the word order on the target side correct. The approaches that this report surveyed gave strong evidence that this 'out-of-order' problem can be remedied by fixing it in the pre-processing stage of translation. Source text can be reordered to fit the target word order, and this could be done by using linguistic information obtained from POS tags, or parse trees on the source side. Given that this approach works well, let us briefly see what could be some steps to improve upon the techniques presented.

Although this report focussed on POS tags and phrase structure trees as carriers of linguistic information, there are other ways of representing the same. For example, we could use dependency trees instead of phrase structures. Even with phrase structure trees, we could investigate using shallow or deep parsers.

The rules that were learned in most of the approaches that we saw were essentially hard patterns. It would be interesting to use wild-cards in rules. For example, when dealing with POS tags, a rule could say: AUX-V-* becomes AUX-*-V. This would allow several reordering phenomena to be modelled using small set of simple rules. Even CFG reordering rules could contain wildcards in them. In fact, the author's recent experiments with manually writing few reordering rules for Arabic to English translation with the help of a linguist revealed that if t-grep style search expressions are used, representing the reordering rules would become very simple. Of course, there is a huge trade off here. Automatically learning rules that contain wildcards is a challenging problem of machine learning. Automatically learning complex search expressions that may not necessarily be context-free is a very difficult problem. However, there may be some reasonable way out of the trade off where we could get reasonable improvements.

If reordering rules are learned from data, there are several unsealed entry doors for errors to come in. Word alignments, parse trees may have errors, POS tagging may not be errorfree. Clearly, while designing a reordering system based on automatic data processing tools, it has to be made robust to these errors.

Overall, the precis of the report is this: "Works great the reordering of sentences source. The problem but is solved not. Reordering creative can indeed boost translation scores."

Bibliography

[AO90] Tania Avgustinova and Karel Oliva. Syntactic description of free word order languages. In *Proceedings of the 13th conference on Computational linguistics*, pages 311–313, Morristown, NJ, USA, 1990. Association for Computational Linguistics.

[BM04] Michele Banko and Robert C. Moore. Part of speech tagging in context. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 556, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[CKK05] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Co06] Josep Maria Crego and José Bernardo Mariño. Integration of postag-based source reordering into smt decoding by an extended search graph. In *The 7th Biennial Conference of the Association for Machine Translation in the Americas*, Boston, MA, USA, 2006.

[DPB01] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. Interlingua-based englishhindi machine translation and language divergence. *Machine Translation*, 16(4):251–304, 2001.

[Kni99] Kevin Knight. Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4):607–615, 1999.

[LLZ+07] Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 720–727, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[LVL+03] Alon Lavie, Stephan Vogel, Lori Levin, Erik Peterson, Katharina Probst, Ariadna Font Llitjós, Rachel Reynolds, Jaime Carbonell, and Richard Cohen. Experiments with a hindi-to-english transfer-based mt system under a miserly data scenario. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(2):143–163, 2003.

[NS06] Thai Phuong Nguyen and Akira Shimazu. *A Syntactic Transformation Model for Statistical Machine Translation*, volume Volume 4285/2006 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006.

[PN06] Maja Popović and Hermann Ney. Pos-based word reorderings for statistical machine translation. In *International Conference on Language Resources and Evaluation*, pages 1278–1283, Genoa, Italy, May 2006.

[RV07] Kay Rottmann and Stephan Vogel. Word reordering in statistical machine translation with a pos-based distortion model. In *TMI '07: Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, 2007.

[XM04] Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 508, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[YK01] Kenji Yamada and Kevin Knight. A decoder for syntax-based statistical mt. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 303–310, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[ZV06] A. Zollmann and A. Venugopal. Syntax augmented machine translation via chart parsing, 2006.