Syntax-enhanced N-gram-based SMT

Josep M. Crego and José B. Mariño

TALP Research Center
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya, Barcelona
{jmcrego,canton}@gps.tsc.upc.edu

Abstract

This paper addresses the problem of word reordering in statistical machine translation. We follow a word order monotonization strategy making use of syntax information (dependency parse tree) of the source language to build a set of automatically extracted reordering rules. The input sentence is extended to a graph built with reordering hypotheses, hence, allowing for a constrained search on the syntactically motivated reorderings.

Experiments are reported on the BTEC corpus (Chinese to English task) Results are presented regarding translation accuracy and computational efficiency, showing significant improvements in translation quality at a reasonable computational cost.

1 Introduction

In the statistical machine translation (SMT) community, it is widely accepted the need for structural information to account for mappings between the different language pairs. These mappings offer a greater potential to learn generalizations about relationships between languages than flat-structured models, such as the word-based IBM models of the early 1990s (Brown et al., 1990) or the more recent phrase-based models (Zens et al., 2002; Marcu and Wong, 2002; Koehn et al., 2003), which to date remain widely used. The need for structural information is specially relevant when handling language pairs with very different word order (such as Chinese-English), because the flat-structured models fail to derive generalizations from the training corpus.

Several alternatives have been proposed in the recent years to boost the use of syntactic information in SMT systems. They range from those aiming at monotonizing the word order of the considered language pairs by means of a set of linguistically-based reordering patterns (though, reducing the reordering needs in the overall search), to others considering translation as a synchronous parsing process, where reorderings introduced in the overall search are syntactically motivated.

Among the first group (**word order monotonization**) we can find (Xia and McCord, 2004), (Collins et al., 2005), (Costa-jussà and Fonollosa, 2006) or (Popovic and Ney, 2006). They modify the source language word order before decoding in order to acquire the word order of the target language. Then, the reordered source sentence is sent to a standard phrase-based decoder to be translated under monotonic conditions. In (Crego and Mariño, 2006) the same idea is enhanced by coupling reordering and decoding (using an N-gram-based system), what allows to further improve translation accuracy by avoiding some of the errors performed in the monotonization preprocessing step. (Collins et al., 2005) proposes a set of hand-crafted reordering rules for a German-English translation task.

The second group (**syntax-directed**) has gained many adepts in the last few years because of the significant improvements made by exploiting the power of synchronous rewriting systems. These systems employ source and/or target dependency (Quirk et al., 2005; Langlais and Gotti, 2006) or constituent trees, which can be formally syntax-based (Chiang, 2005; Watanabe et al., 2006) or linguistically syntax-based (Yamada and Knight, 2002; Wu, 1997; Marcu et al., 2006).

A main criticism to the first group is that it has shown a relatively good performance when tackling language pairs with reduced reordering needs (such as Spanish-English or French-English). On the other hand, syntax-directed systems show a main weakness on their poor efficiency results, recently overrided by the apparition of new decoders, which show significant improvements when handling with syntactically divergent language pairs, under large-scale data translation tasks. An example of such a system can be found in (Marcu et al., 2006).

Similar to (Crego and Mariño, 2006), in this work we follow a word order monotonization strategy applied on an N-gram-based SMT system. We introduce syntax information (in the form of **dependency parse trees**) to the problem of foreseeing which reorderings must be applied in a Chinese-English translation task.

The rest of this paper is organized as follows: in Section 2 we briefly review the N-gram-based approach to SMT. Section 3 outlines the reordering framework employed in this work and describes the syntactically motivated reordering rules. Experiments are reported in section 4. Finally, we draw conclusions and detail further work in section 5.

2 N-gram-based SMT System

Our SMT system follows a maximum entropy framework, where we can define a translation hypothesis t given a source sentence s, as the target sentence maximizing a log-linear combination of several feature functions:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \qquad (1)$$

where λ_m corresponds to the weighting coefficients of the log-linear combination, and the feature functions $h_m(s,t)$ to a logarithmic scaling of the probabilities of each model.

Following this approach, the **baseline** translation system described in this paper implements a log-linear combination of one translation model and **four** additional feature functions (models):

- a *target language model*,
- a *word bonus model*,
- a *source-to-target lexicon model* and
- a *target-to-source lexicon model.*

Given the combination of models presented above, we used **MARIE**[1], a freely available N-gram-based decoder implementing a beam search strategy with distortion (or reordering) capabilities (Crego et al., 2005a)(Crego and Mariño, 2007).

In contrast with standard phrase-based approaches, our translation model is expressed in *tuples* as bilingual units and estimated as an N-gram language model (Mariño et al., 2006). The next equation describes the particular N-gram language model:

$$\arg\max_{t_1^I}\{\prod_{i=1}^{K} p((s,t)_i|(s,t)_{i-N+1},...,(s,t)_{i-1})\} \qquad (2)$$

where $(s,t)_i$ refers to the i^{th} tuple of a given bilingual sentence pair which is segmented into K units.

2.1 Unfolded translation units

Additionally, translation units are extracted with reordered source words, following the **unfold** method described in (Crego et al., 2005b).

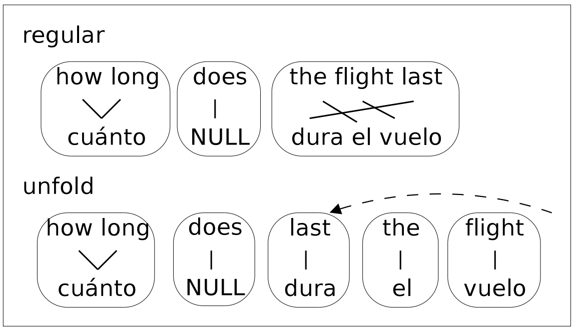

Figure 1: *Regular Vs. Unfold translation units.*

Figure 1 shows an example of translation units extracted with the original source word order (**regular**) and after reordering the source words (**unfold**).

In general, the unfold extraction method outperforms the regular method because it produces smaller units (specially relevant for languages with very different word order). Which means, less sparse and more reusable units. On the other hand, the unfold method needs the input source words be reordered during decoding similarly to how source words were reordered in training. If monotonic decoding were used with unfolded units, translation hypotheses would be formed following the source language word order.

3 Reordering Framework

The reordering framework employed in this work can be seen as a double-sided process. In training time, following the word-to-word alignments, a set of reordering rules are automatically learned, which are later used in decoding

[1] http://gps-tsc.upc.es/veu/soft/soft/marie/

time to provide the decoder with a set of reordering hypotheses in the form of a reordering input graph.

Rules extraction

Source side reorderings are introduced into the training corpus in order to harmonize the word order of the source and target sentences.

For each reordering produced in this step a record is taken in the form of a reordering rule. A reordering rule has the form of:

$$t_1, ..., t_n \mapsto i_1, ..., i_n \qquad (3)$$

where $t_1, ..., t_n$ is a sequence of tags (related to a sequence of source words), and indices $i_1, ..., i_n$ consist of a sequence of positions into which the source tags are to be reordered. In(Crego and Mariño, 2006) are shown reordering rules based on Part-Of-Speech tags used in a Spanish-English translation task.

In this work we extend this approach by allowing for syntax-based rules. Here the source side of the reordering rules consist of syntactic tags (dependency relations) which may be referred to one or a consecutive set of source words (all rules are unlexicalized). Further details in section 3.1.

Search graph extension

In decoding, the input sentence is handled as a word graph. A monotonic search graph contains a single path, composed of arcs covering the input words in the original word order. To allow for reordering, the graph is extended with new arcs, covering the source words in the desired word order (reorder).

The motivation of extending the input graph is double:

- First, the ability to produce reorderings (following the rules explained in the previous lines) aims at improving the translation quality, which is introduced at a reasonable efficiency cost. The size of the input graph can be easily limited in order to balance efficiency and accuracy by filtering out reordering rules. However, in the experiments of this work the entire set of reordering rules is always used due to the (limited) size of the translation task.
- Second, the reordering decision is more informed when tightly coupled with decoding. Some systems perform hard reordering decisions in preprocessing steps (previous to decoding), what may cause unrecoverable reordering errors. In our case the reordering decision is made in the overall search, when all the information (SMT models) are available.

Figure 2 shows an example of input search graph extension. The monotonic search graph (bold arcs) is expanded following two different POS-based reordering rules extracted from a Spanish-English translation task (*NC*, *AQ* and *CC* are POS tags standing respectively for *noun*, *adjective* and *conjunction*).

For a given test sentence, any sequence of the input tags fulfilling a source-side reordering rule implies the addition of a reordering path.

The rule '*NC AQ* → 1 0' indicates that the Spanish sequence '*noun adjective*' must be swapped (reordered) to acquire the right English word order (in English, adjectives are typically placed before the corresponding nouns while the opposite is expected in Spanish). Two new arcs are introduced extending the input graph following the previous rule (upper arcs).

The rule '*NC AQ CC AQ* → 1 2 3 0' illustrates the same situation (opposite word order of nouns and adjectives) but when the adjective role is played by an adjective phrase '*ambicioso y realista*'. Following this rule the graph is extended with four new arcs (lower arcs).

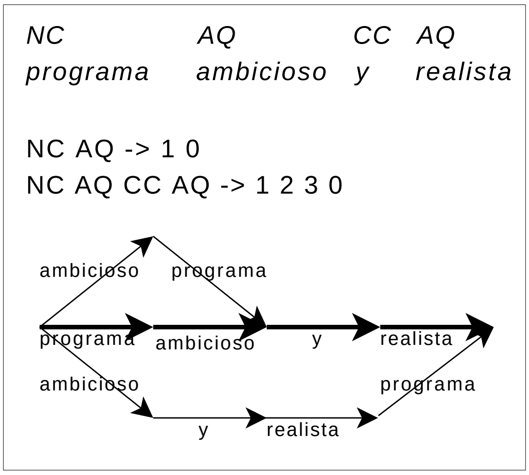

Figure 2: *Input graph extension.*

The previous example exhibits a main weakness of POS-based reordering rules:

While the underlying reordering rule can be simply stated as '*noun adjective* → 1 0' the system needs for a (potentially) infinite number of rules, sequences of POS tags indicating a noun phrase followed by an adjective phrase, to capture all the possible examples (recursive feature of natural languages).

In other words, the generalization power of POS-based reordering rules is somehow limited to short rules (less sparse) which fail to capture many real examples. That motivates the introduction of syntax-based reordering rules.

3.1 Syntax-based Reordering Rules

In this section we describe the use of structure information in the reordering framework described in the previous section.

Figure 3 illustrates the process of extracting syntax-based reordering rules. It basically employs the parse trees of the training source sentences (dependency trees) and their word-to-word alignments. *[syntax tree]*, *[zh]*, *[POS]*, *[align]* and *[en]* indicate respectively the Chinese sentence dependency tree, the Chinese words, the Chinese POS tags, the word-to-word alignment and finally the English corresponding sentence.

As introduced in the previous section, source words are reordered in order to mimic the word order of the target words. Therefore, in the example of figure 3, the reordered source words are indicated by the *[unfolding]* sequence, where the third source word is moved to the last position.

Once a source reordering is identified, a reordering rule is extracted relating the sequence of words implicated on it. In our example the sequence of words is *[3, 4, 5, 6, 7, 8, 9, 10]*. The procedure to extract a rule from the reordering sequence can be decomposed in two steps:

- First, a subtree of the whole parse tree is identified (see *[subtree]* in the example). The subtree must contain all the words of the reordering sequence *[3,..., 10]*, words must be consecutive and there must exist a single root node. In other words, there must exist a path (following the arcs of the tree) between any pair of nodes (source words) of the subtree (the nodes in the subtree must be connected).

  The latter constraint may expand the scope of the reordering sequence with new words. For instance, if the third source word was reordered after the sixth Chinese word (with corresponding reordering sequence *[3, 4, 5, 6]*), the resulting subtree would be the same that in our original example. The reason is that to connect the sixth with the first nodes (*3,4,5*) the subtree needs the introduction of the tenth node, and consequently the nodes *7,8,9* to make the sequence consecutive.

- Second, the previous subtree is pruned out. Any arc of the subtree can be pruned out (removing the arc as well as the node son) when all his brothers (sons of the same father node) maintain the same relative word order.

The pruning introduced in the second step is responsible of the generalization power (sparseness reduction) acquired by the syntax-based reordering rules in contrast to other methods, such as the POS-based method presented in (Crego and Mariño, 2006). In the example, *[rules]* show the unpruned rule (bold), and more generalized rules after the successive prunings (labeled *a) b)* and *c)*).

It is worth saying that the generalization power acquired by the pruning method introduces inaccuracy. Some generalized rules are too general and may only be valid in some cases (for some examples).

The fully-pruned rule (*c)*) is internally recorded using the following rule structure:

$$advmod\{0\}\{1\}\, asp\{2\}\{1\}\, dobj\{3\}\{1\} \rightarrow 1\,2\,3\,0$$

Where nodes ({*0*},{*1*},{*2*} and {*3*}) can designate either words or group of consecutive words, and relationships '$rel\{x\}\{y\}$' should be read as 'x' is a child of 'y' under the 'rel' dependency relationship.

The resulting set of syntax-based rules contains the fully-pruned (generalized with group of words) as well as the unpruned (fully-instantiated) rules. In the example, all the rules shown in the *[rules]* section.

The fully-pruned rules can capture a strict superset of the reorderings than can be captured by the fully-instantiated rules (at least the same), what in principle, makes it redundant to keep all of them. However, the generalization level of a rule can be used (among other information) to compute a confidence measure of the rule, which may help to keep or discard reordering hypotheses (hypothesis pruning used to further boost the efficiency of the system at no accuracy cost).

Nevertheless, in the experiments of this work, no confidence measure is computed.

In decoding, the monotonic search graph is expanded following the rules previously obtained. The procedure is very similar to the one

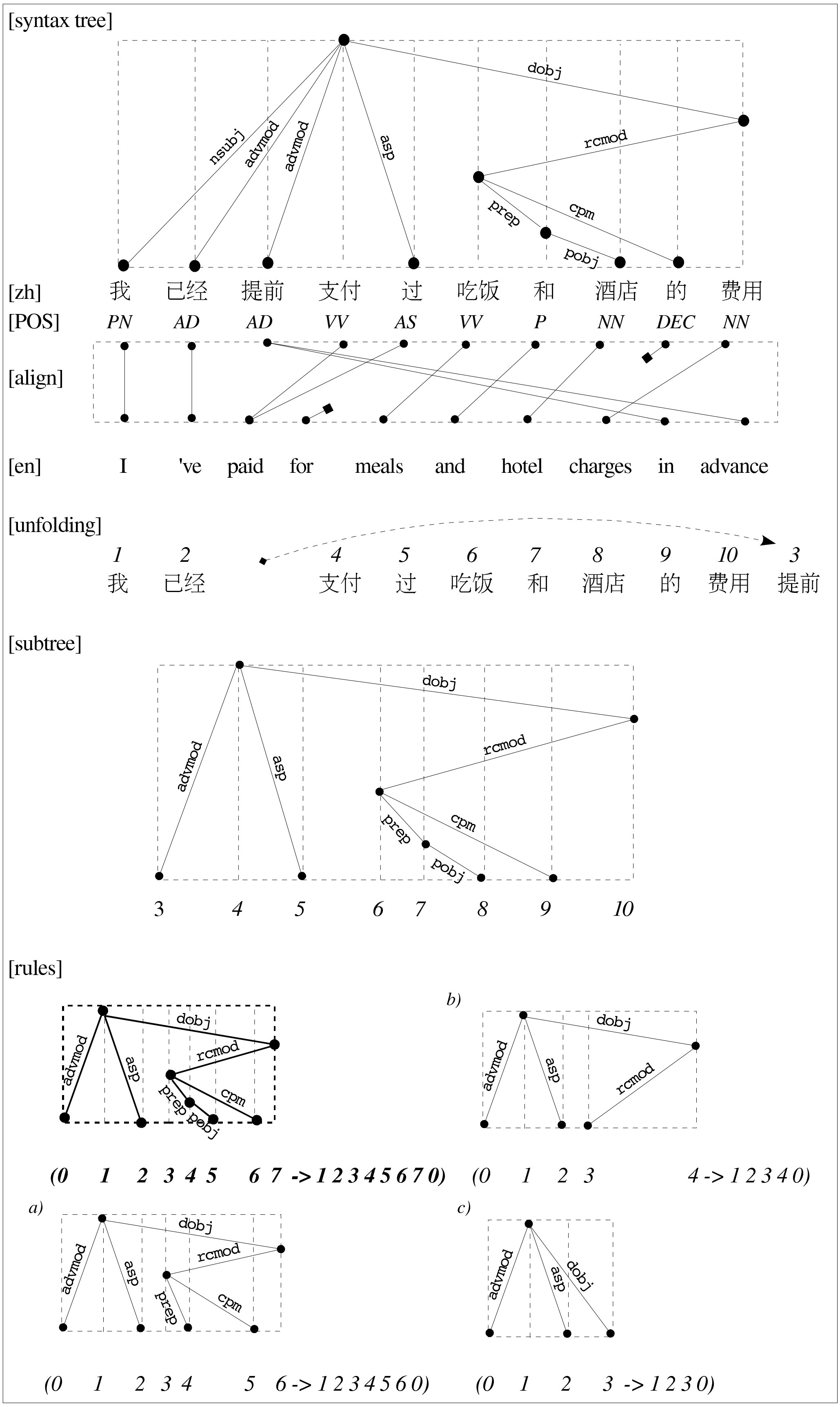

Figure 3: *Extraction of syntax-based reordering rules.*

detailed for the rule extraction. Each sequence of source words is hypothesized as reordering sequence. Hence, for each sequence, its parse subtree is identified and seeked in the training set of reordering rules (the successively pruned versions of the subtree are also searched). If the subtree (or pruned subtrees) exist in the set of rules, the input graph will be extended with a new reordering path, following the word order indicated in the reordering rule.

4 Experiments

In this section we report on the experimental work carried out in this paper.

4.1 Experimental framework

We have used the BTEC[2] corpus (Takezawa et al., 2002) from Chinese to English. It consists exactly of the corpus used in the IWSLT 2006 evaluation campaign as training and development sets. We have used two disjoint sets of the official Development set to build our Dev and Test sets.

Table 1 shows the main statistics of the used data, namely number of sentences, words, vocabulary, average sentence length and number of references for each language.

The training data was preprocessed by using standard tools for tokenizing and filtering.

Word-to-word alignments were computed using GIZA++[3]. The union of both alignment directions was used as starting point to extract translation units (tuples). The Chinese side of the corpus was parsed using the freely available Stanford parser[4].

Lng	Sent	Words	Voc	Avg	Refs
Train					
en	46k	326k	9,6k	7	1
zh		314k	9,7k	6.7	
Dev					
zh	489	5,478	1,096	11,2	7
Test					
zh	500	3,005	909	6	16

Table 1: *BTEC Corpus (Chinese-to-English).*

We used the SRI language modeling toolkit[5] to compute the N-gram language models, using respectively 4 and 5 as n-gram orders for the translation LM and target LM (values empirically determined).

Once the models were computed, optimal log-linear coefficients were estimated for each translation direction and system configuration using an in-house implementation of the widely used downhill simplex method. The BLEU score was used as maximization goal. In the overall search the decoder was always set to perform histogram pruning, keeping the best 50 hypotheses (in the optimization work, histogram pruning was set to keep the best 10 hypotheses).

4.2 Results

Table 2 shows the number of POS-based (**pos**) and syntax-based (**syn**) reordering rules that have appeared in the test set (used to extend the search graph) as a function of the number of tags per rule. Additionally, (**syn'**) shows the size measured in number of raw words (a syntax tag may concern several source words) per syntax-based rule. As a matter of example, the most generalized rule shown in the example of figure 3 (section *[rule]*) is composed of four tags and eight words.

#rules	[2,3]	[4,5]	[6,7]	[8,∞)
pos	2,028	818	100	4
syn	2,394	836	34	0
syn'	1,879	957	282	146

Table 2: *Reordering rules used for the test set.*

As it can be seen, the number of POS-based reordering rules according to its size (**pos**) present in the test set, is similar to the number of tokens of the syntax-based rules (**syn**). In both cases, rules longer than 6 words appear in a reduced number of cases due to its data sparseness problem. However, if number of raw words is taken into account for the syntax-based rules, longer rules appear in a not negligible number of cases.

#reorderings	[2,3]	[4,5]	[6,7]	[8,∞)
pos	156	50	7	0
syn'	198	104	20	17

Table 3: *Reorderings performed in the test set.*

Table 3 shows the number of reorderings performed in decoding for the 1-best translation option of the test set. Reorderings are grouped considering its size (measured in number of raw words). Results are indicated for both config-

[2] Basic Traveler's Expression Corpus

[3] http://www.fjoch.com/GIZA++.html

[4] http://nlp.stanford.edu/downloads/lex-parser.shtml

[5] http://www.speech.sri.com/projects/srilm/

urations: using POS-based (**pos**) and syntax-based (**syn'**) reordering rules.

As it can be seen, the **syn** configuration performs clearly a higher number of reorderings (339 > 213). It is also remarkable the introduction of the longest reorderings concerning the **syn** configuration, which appear hardly ever under the **syn** configuration. This confirms that longest reorderings are useful in such a task, and that they can appear when reducing the sparseness of the rules used to capture them.

In table 4 are shown the accuracy results, in terms of the widely used automatic measures *BLEU*, *NIST*and *mWER* .

The experiment is carried out over three different configurations, corresponding to:

- **mon**. Translation units are extracted with source side in the original word order (regular method). Decoding is carried under monotonic constraints.
- **pos**. Translation units are extracted with reordered source words (unfold method). In decoding, reordering is allowed by means of the POS-based reordering rules.
- **syn**. Translation units are extracted with reordered source words (unfold method). In decoding, syntax-based reordering rules are used.

Config	BLEU	NIST	mWER
mon	39.88	8.666	47.55
pos	42.47	8.875	45.18
syn	45.45	9.090	43.07

Table 4: *Translation accuracy.*

As it can be seen the accuracy is clearly higher when allowing for word reordering than under monotonic decoding conditions. Additionally, the use of syntax-based rules clearly outperforms the POS-based rules. The BLEU confidence interval for a 95% confidence level is ±3.55.

Config	#rules	words/sec.
mon	-	22.6
pos	2,950	4.83
syn	3,264	3.66

Table 5: *Translation efficiency.*

Finally, table 5 shows efficiency in terms of words/second. Figures are very similar for both reordered search conditions as the number of additional paths (reorderings) introduced in both cases are also very similar (see raws *pos* and *syn* in table 2).

Both reordering configurations are less efficient than the monotonic search. However, they clearly outperform a fully reordered search (constrained to a maximum distortion distance of 5 words and 3 reorderings per sentence), which obtains an efficiency score of 0.58 words/second and a less accurate translation.

5 Conclusions

We have tackled the reordering problem in statistical machine translation by means of a word order monotonization strategy, tightly coupled in decoding with the overall search. We have employed syntax information (dependency parse trees) in order to account for the differences in word order between the source and target languages.

Experiments are reported on a data limited Chinese-English translation task showing that accuracy results are improved at a reasonable efficiency cost. Furthermore, the use of syntax information allows to reduce the sparseness problem present in reordering rules built from other linguistic information, such as POS tags.

Current results should be confirmed when expanding the scope to larger data conditions. Further work is envisaged towards several directions:

- Use of weights in the reordering input graph. In order to perform a graph pruning, which may keep the best reorderings discarding the rest (trying to improve the efficiency at no accuracy cost). These weights could also be used as an additional information source in the overall search.
- Recursive reorderings are to be applied over input sentences. The generalization power acquired by the use of syntax information motivates the apparition of reorderings applied recursively (reorderings within reorderings).

6 Acknowledgments

This work has been partially funded by the European Union under the TC-STAR project (IST-2002-FP6-5067-38), the Spanish Government under the AVIVAVOZ project (TEC2006-13694-C03) and the Universitat Politècnica de Catalunya under UPC-RECERCA grant.

References

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J.D. Lafferty, R. Mercer, and P.S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. *43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, June.

M. Collins, Ph. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. *43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, June.

M.R. Costa-jussà and J.A.R. Fonollosa. 2006. Statistical machine reordering. *Proc. of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP'06*, July.

J.M. Crego and J.B. Mariño. 2006. Reordering experiments for n-gram-based smt. *1st IEEE/ACL Workshop on Spoken Language Technology*, December.

J.M. Crego and J.B. Mariño. 2007. Extending marie: an n-gram-based smt decoder. *45rd Annual Meeting of the Association for Computational Linguistics*, April.

J.M. Crego, J.B. Mariño, and A. de Gispert. 2005a. An ngram-based statistical machine translation decoder. *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 3193–3196, September.

J.M. Crego, J.B. Mariño, and A. de Gispert. 2005b. Reordered search and tuple unfolding for ngram-based smt. *Proc. of the MT Summit X*, pages 283–89, September.

Ph. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. *Proc. of the Human Language Technology Conference, HLT-NAACL'2003*, May.

P. Langlais and F. Gotti. 2006. Phrase-based smt with shallow tree-phrases. *Proceedings of the Workshop on Statistical Machine Translation*, pages 39–46, June.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. *Proc. of the Conf. on Empirical Methods in Natural Language Processing, EMNLP'02*, pages 133–139, July.

D. Marcu, Wong. W, A. Echihabi, and K. Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. In *Proc. of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP'06*, pages 44–52, Sydney, Australia, July.

J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M.R. Costa-jussà. 2006. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549.

M. Popovic and H. Ney. 2006. Pos-based word reorderings for statistical machine translation. *5th Int. Conf. on Language Resources and Evaluation, LREC'06*, pages 1278–1283, May.

Ch. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. *43rd Annual Meeting of the Association for Computational Linguistics*, pages 271–279, June.

T. Takezawa, E. Sumita, F. Sugaya, H Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual curpus for speech translation of travel conversations in the real world. *3rd Int. Conf. on Language Resources and Evaluation, LREC'02*, pages 147–152, May.

T. Watanabe, H. Tsukada, and H Isozaki. 2006. Left-to-right target generation for hierarchical phrase-based translation. *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, July.

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

F. Xia and M. McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. *Proc. of the 20th Int. Conf. on Computational Linguistics, COLING'04*, pages 508–514, August 22-29.

K. Yamada and K. Knight. 2002. A decoder for syntax-based statistical mt. *40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, July.

R. Zens, F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *KI - 2002: Advances in artificial intelligence*, volume LNAI 2479, pages 18–32. Springer Verlag, September.