Integration of POStag-based source reordering into SMT decoding by an extended search graph

Josep M. Crego
TALP Research Center (UPC)
Barcelona, 08034
jmcrego@gps.tsc.upc.edu

José B. Mariño
TALP Research Center (UPC)
Barcelona, 08034
canton@gps.tsc.upc.edu

Abstract

This paper presents a reordering framework for statistical machine translation (SMT) where source-side reorderings are integrated into SMT decoding, allowing for a highly constrained reordered search graph. The monotone search is extended by means of a set of reordering patterns (linguistically motivated rewrite patterns). Patterns are automatically learnt in training from word-to-word alignments and source-side Part-Of-Speech (POS) tags. Traversing the extended search graph, the decoder evaluates every hypothesis making use of a group of widely used SMT models and helped by an additional Ngram language model of source-side POS tags.

Experiments are reported on the Euparl task (Spanish-to-English and English-to-Spanish). Results are presented regarding translation accuracy (using human and automatic evaluations) and computational efficiency, showing significant improvements in translation quality for both translation directions at a very low computational cost.

1 Introduction

In statistical machine translation, the use of reordering strategies allows for an important improvement in translation accuracy, specially when translating between language pairs with high disparity in word order. On the other hand, when arbitrary word reorderings are permitted, the search problem is classified NP-complete (Knight, 1999), while polynomial time search algorithms can be obtained under monotone conditions.

The first SMT systems introducing reordering capabilities were founded on the brute force of computers, aiming at finding the best hypothesis through traversing a fully reordered graph (the whole permutations of source-side words are allowed in the search). This approach is computationally very expensive, even for very short input sentences. Therefore, different distance-based reordering constraints must be used to make the search feasible: **ITG** (Wu, 1996), **IBM** (Berger et al., 1996), **Local** (Kanthak et al., 2005), **MaxJumps** (Crego et al., 2005), etc. The use of these constraints implies a necessary balance between translation accuracy and efficiency.

Typically, a distance-based reordering model is used in the search to penalize longer reorderings, only allowed when well supported by the rest of models. Obviously, this model does not follow any property of language. Lexicalized reordering models, which use distance of words seen in train to score reorderings in search, (Koehn et al., 2005) , (Kumar and Byrne, 2005) have also been introduced.

A main criticism to this brute force approach is that it does not make use of any linguistic information, while in linguistic theory, reorderings between linguistic phrases in different language pairs are well described.

Lately, some SMT systems have introduced linguistic information in order to tackle the reordering problem:

- Reordering encoded within translation units in form of hierarchical units (Chiang, 2005), or phrases with gaps (Simard et al., 2005).
- Word order monotonization (in train and/or test). Consisting of learning reorderings into the source side to achieve a similar word order to that of the target side (Collins et al., 2005), (Xia and McCord, 2004).

We found specially interesting the work in (Xia and McCord, 2004), where reorderings are applied following a set of patterns which are automatically learned using lexical, syntactical and morphological information (words, parse trees, and POS tags). In test, a monotone search is applied after reordering the source words using the learnt patterns.

In this work we follow a similar strategy to learn reordering patterns but aiming at reducing the search graph. Our goal is double. On the one hand we add some linguistic information to the problem of guessing which reorderings must be applied (achieving generalization power through using POS tags instead of words). On the other hand, the final decision about reordering is taken in decoding time, when all the information is available (not just reordering patterns but the whole SMT models).

In (Matusov et al., 2005) a similar work can be found, where search graphs are restricted without linguistic motivation but using monotonic sequences seen in training.

The paper is organized as follows. In section 2 we review the translation system used in this work. Section 3 introduces the reordering framework proposed, giving details of the method used to extract reordering patterns, and how reorderings are supplied to the decoder in form of a reordering graph. Section 4 presents the experiments conducted to test the efectiveness of using the new reordering framework. Finally, Conclusion and further work are outlined in section 5.

2 Ngram-based SMT System

Our SMT system follows the maximum entropy framework (Berger et al., 1996), where we can define the translation hypothesis t given a source sentence s, as the target sentence maximizing a log-linear combination of feature functions, as described in the following equation:

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

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 **five** additional feature models. In contrast with standard phrase-based approaches, our translation model is expressed in *tuples* as bilingual units.

Given a word alignment, tuples define a unique and monotonic segmentation of each bilingual sentence , building up a much smaller set of units than with phrases and allowing N-gram estimation to account for the history of the translation process (Mariño et al., 2005).

The tuple N-gram translation model is a language model of a particular language composed by bilingual units which are referred to as tuples. This model approximates the joint probability between source and target languages by using N-grams as described by the following equation:

$$\hat{t}_1^I = \arg\max_{t_1^I} \{p(s_1^J, t_1^I)\} = \cdots = \qquad (2)$$

$$\arg\max_{t_1^I} \{\prod_{i=1}^{K} p((s,t)_i | (s,t)_{i-N+1}, ..., (s,t)_{i-1})\} \qquad (3)$$

where $(s,t)_i$ refers to the i^{th} tuple of a given bilingual sentence pair which is segmented into K units. It is important to notice that, since both languages are linked up in tuples, the context information provided by this translation model is bilingual.

- target language model
- word bonus model
- source-to-target lexicon model
- target-to-source lexicon model
- tagged target language model (using POS tags)

The first of these feature functions is a standard *target language model*, estimated as an N-gram over the target words, as expressed by this equation:

$$p_{LM}(t_k) \approx \prod_{n=1}^{k} p(w_n|w_{n-2}, w_{n-1}) \qquad (4)$$

where t_k refers to the partial translation hypothesis and w_n to the n^{th} word in it.

Usually, this feature function is accompanied by a *word bonus model*. This model introduces a sentence length bonus in order to compensate the system preference for short target sentences caused by the presence of the previous target language model. This bonus depends on the total number of words contained in the partial translation hypothesis, and it is computed as follows:

$$p_{WP}(t_k) = exp(\text{number of words in } t_k) \qquad (5)$$

where, again, t_k refers to the partial translation hypothesis.

The third and fourth feature functions correspond to source-to-target and target-to-source *lexicon models*. These models use IBM model 1 translation probabilities to compute a lexical weight for each tuple, which accounts for the statistical consistency of the pairs of words inside the tuple. These lexicon models are computed according to the following equations:

$$p_{IBM1}((t,s)_n) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(t_n^i|s_n^j) \quad (6)$$

$$p_{IBM1'}((t,s)_n) = \frac{1}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=0}^{J} p(t_n^j|s_n^i) \quad (7)$$

where s_n^j and t_n^i are the j^{th} and i^{th} words in the source and target sides of tuple $(t,s)_n$, being J and I the corresponding total number words in each side of it.

Finally, the fifth feature function consists of an N-gram language model estimated over the same target-side of the training corpus but using POS tags instead of raw words. The same equation 4 expresses the fifth feature function when replacing words (w_n) by POS tags (pos_n).

$$p_{LM}(t_k) \approx \prod_{n=1}^{k} p(pos_n|pos_{n-2}, pos_{n-1}) \qquad (8)$$

2.1 Ngram-based Decoder

Given the combination of models presented above, we used **MARIE**, a freely available decoder implementing a beam search strategy with distortion (or reordering) capabilities.

For efficient pruning of the search space, several pruning techniques are used, such as *threshold pruning*, *histogram pruning* and *hypothesis recombination*.

When allowing for reordering, the pruning strategies are not enough to reduce the combinatory explosion without an important loss in translation performance. With this purpose, two distance-based reordering strategies are used:

- A distortion limit (m): Any source word (phrase or tuple) is only allowed to be reordered if it does not exceed a distortion limit, measured in words.

- A reordering limit (j): Any translation path is only allowed to perform j reordering jumps.

3 Reordering Framework

In this section we outline the methods used to compute reordering patterns in training and to extend a monotone graph with additional arcs (reordering arcs) using the previous patterns.

3.1 Reordering Patterns using POS tags

To extract patterns, we use the word-to-word alignments (the union of both alignment directions) and source POS tags. The main procedure consists of identifying all crossings produced in the word-to-word alignments.

The next equation formalizes the set of crossings of a given pair of sentences word-to-word aligned:

$$\{(j_1, j_2) \,/\, (j_1 < j_2) \wedge (a[j_1] > a[j_2])\} \qquad (9)$$

where $a[j]$ accounts for the maximum target-side position to which the source word j is aligned to, and j_1, j_2 range over $[1, J]$.

Once a crossing has been detected, its source POS tags and alignments are used to account for a new instance of pattern. The target side of a pattern (source-side positions after reordering), is computed using the original order of the target words to which the source words are aligned. See figure 1 for a clarifying example of pattern extraction.

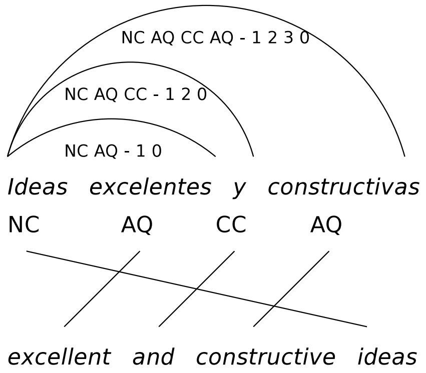

Figure 1: *Reordering patterns are extracted using word-to-word alignments (links between source and target words). Three instances of different patterns are extracted.(**NC AQ - 1 0**, **NC AQ CC - 1 2 0** and **NC AQ CC AQ - 1 2 3 0**).*

3.2 Extending a monotone search graph with additional arcs

The monotone search graph is extended with reorderings following the patterns found in training. The procedure identifies first 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. Figure 2 shows an example of the procedure.

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).

Additional details are given in section 4.2.

4 Experiments

4.1 Corpus

The EPPS data set corresponds to the parliamentary session transcriptions of the European Parliament and is currently available at the Parliament's website (`http://www.euro parl.eu.int/`). In the case of the results presented here, we have used the version of the EPPS data that was made available by RWTH Aachen University through the TC-STAR consortium[1].

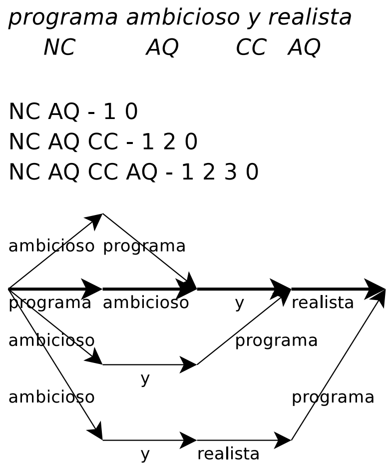

Figure 2: *Three additional arcs have been added to the original monotone graph (bold arcs) given the reordering patterns found matching any of the source POS tag sequences. Additional arcs are computed by using the reordering patterns (before decoding takes place).*

Table 1 presents some basic statistics of training, development and test sets, for each considered language: English and Spanish. More specifically, the statistics presented in Table 1 are the total number of sentences, the total number of words, the vocabulary size (or total number of distinct words) and the vocabulary of POS tags.

4.2 System details

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

Once the training data was preprocessed, word-to-word alignments were performed in both directions, source-to-target and target-to-source, by using GIZA++ (Och, 2003) and the intersection and union sets of both alignments were computed. To com-

[1]TC-STAR (Technology and Corpora for Speech to Speech Translation) is an European Community project funded by the *Sixth Framework Programme*. More information can be found at the consortium website: `http://www.tc-star.org/`

	sent	words	voc	POSvoc
Train set				
English	1.28 M	34.9 M	106 k	44
Spanish		36.6 M	153 k	328
Dev set				
English	735	18,764	3,193	41
Spanish	430	15,332	3,217	181
Test set				
English	1,094	26,917	3,958	42
Spanish	840	22,774	4,081	196

Table 1: *TC-Star English-Spanish Parallel corpus statistics.*

pute the alignments, five iterations for models IBM1 and HMM, and three iterations for models IBM3 and IBM4, were performed.

Then, a tuple set for each translation direction was extracted from the union set of alignments. The resulting tuple vocabularies were pruned considering the N best translations for each tuple source-side ($N = 30$ for the English-to-Spanish and $N = 20$ for the Spanish-to-English).

The English side of the training corpus was POS tagged using the freely available TNT tagger (Brants, 2000), for the Spanish side we used the freely available Freeling (Carreras et al., 2004). Only the first two characters of each tag were used for the Spanish side, aiming at achieving a higher level of generalization.

We used the SRI Language modelling toolkit (Stolcke, 2002) to compute the three Ngram language models, using respectively 4, 5 and 5 as ngram orders for the translation, target and tagged target models.

Once the models were computed, sets of 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 (Nelder and Mead, 1965).

The decoder was always set to perform histogram pruning, keeping the best $b = 25$ hypotheses.

4.3 Used patterns

A huge amount of patterns can be extracted using the method outlined in section 3, most of which only appear because of erroneous word-to-word alignments. In addition, the reordering needs of a Spanish-English translation task are very limited. Furthermore, working with a large list of patterns is not desirable because it slows down decoding.

For such reasons, we filtered out the list according to the following constraints:

- A maximum difference in number of words in both sides of an instance ($dif < 4$).
- A maximum number of source words for a pattern ($nwords < 8$).
- Only patterns with a minimum number of instances in train were kept ($N > 1000$).
- A score p was used to prune out a pattern. The score consists of the number of occurrences of a pattern divided by the number of occurrences of the pattern source words ($p > 0.2$).

After filtering, the list of patterns was reduced to those shown in table 3 (For the English-to-Spanish direction, 29 patterns were extracted). The table also shows the number of occurrences in training, development and test sets and an example of each pattern. Despite the filtering process, some patterns in the table are still erroneous. Some instances of patterns may occur due to wrong alignments, some others to the extraction method used in this work, where all crossings are taken into account. For instance, regarding the pattern **(NC RG - 1 0)**, it should be discarded as it consists of an inner crossing. That is, the right pattern usually contains inner crossings which may not be used as patterns. For this example, the right pattern would be like **(NC RG AQ - 1 2 0)**.

However, the framework proposed in this work does not aim at performing perfect reordering decisions before decoding (hard decisions) but only at reducing the number of reorderings that a fully reordered graph performs. Of course, the better the list of patterns provided to the decoder, the higher efficiency level it will achieve.

Even with (a priori) erroneous patterns, the current list is useful to test the ability of the decoder (the models used in decoding) to discard the wrong ones, performing only the reorderings supported by all the models.

The POS tagged source-side of the training was reordered following the previous filtered list of patterns. The resulting corpus is used to learn a POS tags N-gram language model (set to order 5). This model pretends to be used as a reordering model, supporting the decoder decisions of taking either monotonous or reordered (additional) paths. Of course this model is only helpful when reordering is applied. It is worth saying that only the training source POS tags were reordered in training (translation units were extracted using the source and target words in the original order).

Whenever more than one pattern can be used to reorder the same sequence of source words, the pattern with highest priority is always used. Patterns have been prioritized by its number of POS tag tokens (the longer the pattern, the higher the priority).

4.4 Results

In order to evaluate the different reordering approaches, three different system configurations have been considered:

- **baseline**: Using monotone search.
- **rgraph**: The same baseline system but allowing for reorderings using the reordering graphs built from the patterns learnt in training.
- **pos**: The baseline system along with the reordering graphs and an additional POS tagged reordered source-side language model.

The algorithms used for computing the evaluation measurements (BLEU, NIST, mWER and PER) were the official TC-STAR evaluation tools distributed by ELDA (http://www.elda.org/). Two reference translations were available for each language test set.

Table 2 shows the results obtained by each configuration, the second row shows the BLEU score obtained in the dev set after the optimization. The last four columns show the results on the test set.

As it can be seen in the Spanish-to-English task, development and test sets have a strong correlation when comparing the different configurations in terms of all measures except for PER. The reason is that PER does not account for reordering errors, which are the main differences between the system configurations. The rest of measures show the accuracy improvement of the system when using the *rgraph* configuration and also regarding the *pos* one.

About the English-to-Spanish task, when comparing the *base* and *rgraph* configurations, only the BLEU score shows a clear improvement, the rest of scores remain similar in both configurations. However, regarding the *pos* configuration, the improvement is clearly shown by all the scores. This situation is probably produced by a local maxima achieved in the development work when optimizing BLEU (which for the development set achieves almost the same score in both, *rgraph* and *pos*, configurations).

Conf	bleu'	bleu	nist	mwer	per
Spanish-to-English					
base	.529	.552	10.69	34.40	25.32
rgraph	.533	.556	10.70	34.23	25.50
pos	.539	.564	10.75	33.75	25.41
English-to-Spanish					
base	.481	.480	9.84	41.18	31.11
rgraph	.490	.485	9.81	41.15	31.87
pos	.491	.489	9.91	40.29	31.27

Table 2: *Translation results obtained by the baseline system, the baseline system using reordering graphs, and the baseline system using reordering graphs and helped by an additional POS tags Ngram model of the reordered source-side. The second row indicates the bleu score obtained in the dev set after the optimization. The last four columns show the results on the test set. Confidence intervals of BLEU are* ± 1.12 *and* ± 1.62 *(Spanish-to-English and English-to-Spanish respectively) for a* 95% *confidence level.*

In table 3 a human evaluation of the test sets (columns fifth and sixth) is also shown. The fifth column shows the number of reorderings performed by the decoder, while column six shows the number of subjective errors detected on the sequences of words the decoder was allowed to reorder.

Regarding the subjective evaluation, we focused on the sequences added to the graph as additional arcs (reorderings), and evaluated as erroneous both, bad reordering decisions and bad monotone decisions. By bad decisions we do not mean a bad translation (what is already done by automatic measures) but a bad word order in the target language. For instance, given the input sentence **'programa ambicioso y realista'** if the decoder decides to use the pattern **(NC AQ CC AQ - 1 2 3 0)** showing the translation **'ambitious and unrealistic programme'**, we account for a success, even if the translation is semantically wrong (the right order was achieved).

Pattern	train	dev	test	swap	error	Example
NC RG AQ CC AQ $\rightsquigarrow$ 1 2 3 4 0	1,406	1	1	1	0	ideas muy sencillas y elementales
NC AQ CC AQ $\rightsquigarrow$ 1 2 3 0	27,119	13	**23**	**17**	**2**	programa ambicioso y realista
NC AQ RG AQ $\rightsquigarrow$ 2 3 1 0	1,971	0	4	1	0	control fronterizo más estricto
NC CC NC AQ $\rightsquigarrow$ 3 0 1 2	3,355	6	**12**	**6**	**3**	mezquitas y centros islámicos
NC RG AQ CC $\rightsquigarrow$ 1 2 3 0	2,226	3	2	0	0	ideas muy sencillas y
AQ RG AQ $\rightsquigarrow$ 1 2 0	2,777	21	*7*	*2*	*1*	europea más sólida
NC AQ AQ $\rightsquigarrow$ 2 1 0	35,661	11	**24**	**18**	**3**	decisiones políticas delicadas
NC RG AQ $\rightsquigarrow$ 1 2 0	32,887	0	**35**	**26**	**1**	ideas muy sencillas
NC RG RG $\rightsquigarrow$ 1 2 0	1,473	0	3	3	2	texto mucho más
NC AQ $\rightsquigarrow$ 1 0	877,580	113	**142**	**110**	**16**	preguntas serias
NC RG $\rightsquigarrow$ 1 0	54,968	27	*47*	*7*	*7*	actividades aparentemente
AQ AQ $\rightsquigarrow$ 1 0	46,509	14	*40*	*4*	*2*	medioambientales europeas
RN VM $\rightsquigarrow$ 1 0	45,777	4	2	1	1	no promuevan
RG VA $\rightsquigarrow$ 1 0	9,824	0	2	1	0	ahora habíamos
AQ RG $\rightsquigarrow$ 1 0	8,701	11	*21*	*4*	*2*	suficiente todavía
RG VS $\rightsquigarrow$ 1 0	5,043	1	1	1	0	supuestamente somos
VM PP $\rightsquigarrow$ 1 0	4,769	6	**13**	**12**	**2**	estar ustedes
Total (17)	1,162,046	231	379	214	42	

Table 3: *List of patterns extracted from the training corpus for the Spanish-to-English translation direction. The first column shows each pattern, the next three columns show the occurrences of each pattern in train, test, and dev sets. Columns fifth and sixth show the results of the human evaluation. Finally, the last column shows an example of each pattern.*

Regarding the Spanish-to-English direction, the decoder finally decided to reorder 214 of the 379 additional arcs ($\sim 56\%$). Doing so, 42 decisions (either to reorder or to keep the monotone order) were wrong ($\sim 11\%$). Similar averages are achieved for the English-to-Spanish direction.

From the human evaluation, we can divide patterns in two main groups. First (in italic), those patterns with very few reorderings performed. Second (in bold), those patterns for which the decision to reorder (swap) was more often taken.

The first group identifies which patterns can be filtered out from the pattern list, as never showed a right reordering (almost no reordering has been performed). The second group shows that the decoder (SMT models) is able to decide whether a pattern (sometimes a very general rule) is suitable to be used for a given instance or not. Although the constraints used to filter out the reordering patterns were not deeply studied, it seems that the right (useful) set of rules does not differ very much from the set used in this work (due to the limited needs for reordering of the Spanish-English pair).

Finally, figure 3 shows the number of hypotheses expanded for different search conditions: a monotone search, a reordered search using reordering patterns, and a reordered search using distance-based constraints ($m = 3$ and $j = 3$). The English-to-Spanish test set was used in all cases.

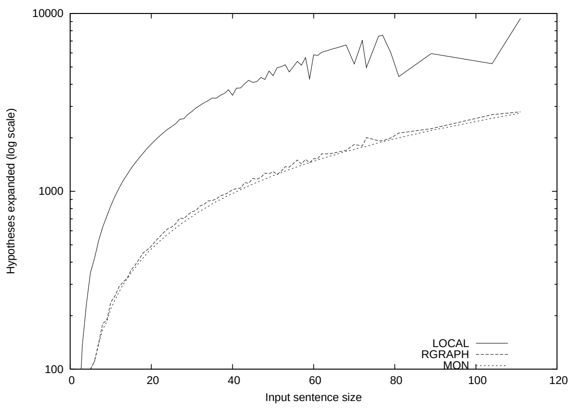

Figure 3: *Number of hypotheses expanded in the search when allowing for different reordering constraints. Monotone (MON), using the reordering patterns (RGRAPH) and using local (distance-based) constraints (LOCAL).*

It is remakable the fact that performing the search restricted with reordering patterns, achieves a similar level of efficiency than the monotone search.

5 Conclusions and Further work

This paper presents a reordering framework where linguistically motivated source-side reorderings are integrated into SMT decoding. Patterns are automatically learnt using word-to-word alignments and POS tags.

The framework has been tested on an Ngram-based SMT system, achieving accuracy improvements at a very low efficiency cost (very similar results regarding the monotone search). We have also performed a human evaluation that has revealed the robustness of the framework when dealing with less accurate reordering patterns.

Further work is envisaged to improve the identification of patterns (in terms of a better filtering, using additional information such as raw words, chunks and parse trees). Also work is being done in order deal with different language pairs.

6 Acknowledgements

This work has been partially funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation - (IST-2002-FP6-506738, http://www.tc-star.org), and the Universitat Politècnica de Catalunya under UPC-RECERCA grant.

References

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March.

T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proc. of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.

X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. Freeling: An open-source suite of language analyzers. *4th Int. Conf. on Language Resources and Evaluation, LREC'04*, May.

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.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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

S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney. 2005. Novel reordering approaches in phrase-based statistical machine translation. *Proceedings of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 167–174, June.

K. Knight. 1999. Decoding complexity in word replacement translation models. *Computational Linguistics*, 26(2):607–615.

P. Koehn, A. Axelrod, A. Birch, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. *Proc. of the Int. Workshop on Spoken Language Translation, IWSLT'05*, October.

S. Kumar and W. Byrne. 2005. Local phrase reordering models for statistical machine translation. *Proc. of the Human Language Technology Conference, HLT-EMNLP'2005*, October 6-8.

J.B. Mariño, R Banchs, J.M. Crego, A. de Gispert, P. Lambert, M. R. Costa-jussà, and J.A.R. Fonollosa. 2005. Bilingual n–gram statistical machine translation. *Proc. of the MT Summit X*, September.

E. Matusov, S. Kanthak, and H. Ney. 2005. Efficient statistical machine translation with constrained reordering. *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)*, pages 181–188, May.

J.A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7:308–313.

F.J. Och. 2003. Giza++ software. http://www-i6.informatik.rwth-aachen.de/˜och/ software/giza++.html. Technical report, RWTH Aachen University.

M. Simard, N. Cancedda, B. Cavestro, M. Dymetman, E. Gaussier, C. Goutte, K. Yamada, P. Langlais, and A. Mauser. 2005. Translating with non-contiguous phrases. *Proc. of the Human Language Technology Conference, HLT-EMNLP'2005*, page 8, October 6-8.

A. Stolcke. 2002. Srilm - an extensible language modeling toolkit. *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02*, September.

D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. *34th Annual Meeting of the Association for Computational Linguistics*, pages 152–158, June.

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*, August 22-29.