Decoding by Dynamic Chunking for Statistical Machine Translation

Sirvan Yahyaei
Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
sirvan@dcs.qmul.ac.uk

Christof Monz
ISLA, Informatics Institute
University of Amsterdam, Science Park 107
1098 XG Amsterdam, The Netherlands
c.monz@uva.nl

Abstract

In this paper we present an extension of a phrase-based decoder that dynamically chunks, reorders, and applies phrase translations in tandem. A maximum entropy classifier is trained based on the word alignments to find the best positions to chunk the source sentence. No language specific or syntactic information is used to build the chunking classifier. Words inside the chunks are moved together to enable the decoder to make long-distance re-orderings to capture the word order differences between languages with different sentence structures. To keep the search space manageable, phrases inside the chunks are monotonically translated, thus by eliminating the unnecessary local re-orderings, it is possible to perform long-distance re-orderings beyond the common fixed distortion limit. Experiments on German to English translation are reported.

1 Introduction

Despite the success of phrase-based statistical machine translation systems, fluency of the output, particularly for long sentences still remains one of the main challenges in current research on Machine Translation (MT). Most of the errors in the MT output are caused by word-order differences between the source and the target language. Compared to word-based Statistical Machine Translation (SMT) systems, phrase-based approaches perform very well in capturing local re-orderings. However, long distance re-orderings remain a serious challenge. As Knight (1999) showed, trying all the permutations is computationally intractable, and most phrase-based MT systems restrict the search space by limiting the set of re-orderings that are explored during decoding. Zens et al. (2004) examine the effect of different constraints on machine translation quality.

A constraint commonly used in phrase-based machine translation is the so-called distortion limit, which restricts the distance between the next phrase and the previously translated phrase. Most approaches described in the literature report a distortion limit ranging between 4 and 12 words. This limitation of course prohibits any word-reordering going beyond the set limit. This might not be a problem for language pairs with similar word order such as English-French or Dutch-German (Birch et al., 2008). A good language model or a lexicalized re-ordering model (Koehn et al., 2005) will be enough to capture the word order differences in these cases. However, when translating between languages with rather different word order, for example an SOV (subject-object-verb) language into an SVO (subject-verb-object) language, the distortion limit restriction can severely affect the decoder's ability to capture those word order differences correctly. When translating from German (an SOV language) into English (an SVO language), it is not unusual that more than 20 words on the source side need to be jumped over to translate the verb in the right position. While relaxing the distortion limit accordingly may seem a possible solution to this problem, it has two severe shortcomings: Firstly, decoding time rapidly increases with more relaxed distortion limits. Secondly, wider distortion limits also allow for

any re-ordering within the distortion limit which increases the level of noise and puts a higher burden on the language model to demote wrong re-orderings.

In this paper, we propose a method to enable the decoder to consider permutations which include long distance re-orderings. By grouping words and moving them together, we try to enable the decoder to consider long-distance re-orderings and avoid unnecessary short distance permutations. In addition, our method does not rely on language-dependent parsers or chunkers and uses the word alignment information to build the chunker.

The rest of the paper is organized as follows: Section 2 provides an overview of the related work addressing the issue of word re-ordering in statistical machine translation and the use of chunking in particular. Section 3 explains our proposed method. Section 4 discusses our experimental settings and results comparing the chunking method to a baseline. In Section 5 we draw some conclusions and discuss open issues.

2 Related Work

Several phrase-based SMT systems use a very simple distance-based re-ordering model (Koehn et al., 2003; Koehn et al., 2007). In such a distance-based model, monotone translation and short jumps are preferred over longer jumps. The cost in this model increases linearly by distance with a slight preference for jumps to the right:

$$d(i) = start_i - end_{i-1} - 1 \quad (1)$$

where $d(i)$ is the distortion cost of translating the ith phrase after the $(i-1)$th.

More recently, there have been efforts to incorporate syntax into statistical machine translation, particularly in order to address the issue of word re-ordering. A method to incorporate syntactic information is to apply syntactically motivated rules to render the word order of the source sentence similar to the target language. These transformation rules can be syntax-based or lexicalized rules. A syntax-based rule is a transformation rule that only contains syntactic tags (Collins et al., 2005; Wang et al., 2007), but a lexicalized rule contains at least one word as a constraint (Xia and McCord, 2004). Xia and McCord (2004) proposed a method to learn transformation rules, lexicalized and syntax-based (unlexicalized), from a parallel corpus. Their approach extracts re-write patterns, applies them to the source sentence after which the sentence is translated monotonically. To learn the rewrite patterns, the source side of the bitext is parsed, phrases are aligned and lexicalized, and unlexicalized patterns consisting of parent nodes with their children, plus their syntactic labels are extracted.

Collins et al. (2005) present an approach similar to (Xia and McCord, 2004), but with handcrafted, syntax-based rules to re-write source sentences. They argue that baseline phrase-based models are unable to perform the re-orderings found in translating between German and English. They show that many of the re-orderings require long distance jumps which are heavily penalized by a decoder applying a distance-based re-ordering strategy. Another benefit of source re-ordering is its ability to bring together source words that cannot be extracted as a phrase as they are non-contiguous in the original source sentence.

Chen et al. (2006) extract rules at the part-of-speech (POS) level from the word alignments and apply these rules to reorder the source sentences. Crego and Marino (2006) extract rewrite patterns at POS level as well, however, instead of re-ordering the source sentence, the re-ordering operations are integrated into the decoding process.

Zhang et al. (2007) developed a method similar to other source re-ordering methods, however their approach works on an intermediate level called 'syntactic chunks'. A syntactic chunk is a series of words that consist of a grammatical unit such as noun and verb. They use a maximum entropy tool to build the chunking model with training data provided by converting subtrees of Chinese treebank into chunks. A rule is composed of chunk and POS tags, where a chunk tag for each word determines the chunk type that the word belongs to and also whether the word is at the beginning of the chunk. Before extracting the rules POS tagging and chunking is applied. As several conflicting rules can match a given sentence, the different rule applications are passed to the decoder as a lattice.

For all of the the re-ordering approaches discussed above, a syntactic parser, chunker, or POS tagger of the foreign language is required. Unfortu-

nately, these resources (at sufficient levels of accuracy) tend to be scarce for many languages.

On the other hand, we believe re-ordering the source sentence makes hard decisions that cannot be undone. For example, Xia and McCord (2004) report a decrease in translation quality by allowing permutations after re-ordering the source sentence. Also, since all re-orderings are done beforehand, the impact of n-gram language models, which is quite crucial in other approaches, is eliminated. To take advantage of the language model feature, we prefer to make re-ordering decisions during decoding. In addition, since one of the strengths of phrase-based models is to learn many phrases which do not necessarily belong to any syntactic category (DeNeefe et al., 2007), we believe the syntactic chunks may diminish this feature. Therefore, we suggest to consider all possible chunks and identify the optimal chunk boundaries during decoding.

There are also a number of re-ordering approaches that fully integrate re-ordering into the decoding process, see, e.g., (Al-Onaizan and Papineni, 2006; Tillmann, 2004). These models typically predict the jump orientation (and sometimes distance) based on the previously translated phrase and the phrase that is to be translated next. A few simple syntactic features have been used in some of these models (Crego and Marino, 2006), however the fully lexicalized parameters remain the main source of evidence. Our method differs from lexicalized re-ordering models as it allows permutations beyond the fixed distortion limit and also removes the need for considering many unnecessary local re-orderings.

3 Integrating Chunking and Decoding

In this section we describe our approach which integrates chunking and decoding. While all of the previous chunk-based decoders first apply chunking, then reorder the chunks, and finally perform translation, our approach performs chunking and decoding at the same time. The advantage is that decisions at each level (chunking, chunk-based re-ordering, and translation) are not made independently of each other.

Penalizing the jumps according to the number of words in distance-based re-ordering severely discourages making long distance re-orderings and tends to bias the decoder to translate most of the sentences monotonically (Collins et al., 2005). Here, we group words together and penalize the jumps based on the number of skipped chunks. This enables the decoder to skip more than a fixed number of words and allows for long-distance re-orderings. On the other hand, we chunk the source sentence in a way that words inside a chunk can be translated monotonically in either direction: right to left or left to right. By eliminating local re-orderings (apart from the local re-orderings that are captured by the phrase translations themselves) within the chunks the size of the search size is kept manageable during decoding.

To accomplish this, we extended the standard phrase-based multi-stack decoding approach to simultaneously chunk and apply phrase applications. The approach consists of two components: Firstly, a chunk scoring component which is a binary classifier that gives each chunking candidate a score, and, secondly, an extension to the decoder that either expands the current chunking decision or applies a phrase translation inside an uncovered chunk.

3.1 Chunking Scorer

We define a chunk as a contiguous group of words that can be translated monotonically from left to right or right to left. Figure 1 shows an alignment matrix for a pair of sentences. Given a word alignment a_1^J between a source sentence $\mathbf{f} = f_1, ..., f_J$ and target sentence $\mathbf{e} = e_1, ..., e_I$. We define a *chunk boundary* between f_j and f_{j+1} if there is no source word aligned to $\{i | a_j < i < a_{j+1}\}$. For instance, in the example alignment, there is no *chunk boundary* between f_6 and f_7, because there is no i such as $a_6 < i < a_7$. Analogously for f_1 and f_2, as there is no source word aligned to e_2. According to this definition there is, for example, a *chunk boundary* between f_2 and f_3. The example in figure 1 contains three chunks. With this definition, a binary classifier will be learned to classify every point between two foreign words under two classes: 'chunk boundary' and 'no chunk boundary'

We use a maximum entropy classifier for this purpose and define a set of features based on the word alignments and above definition. Our set of feature functions include:

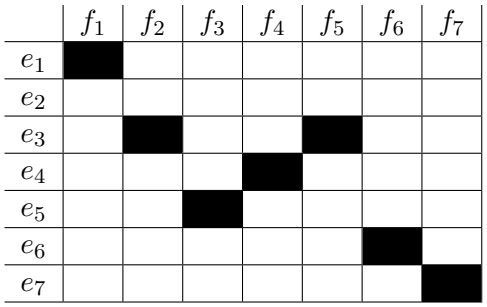

Figure 1: An example of chunks with left to right, (f_1, f_2), (f_6, f_7) and right to left (f_3, f_4, f_5) orientations.

- $h_1(\delta, f_j, f_{j+1})$, where $\delta \in \{1, 0\}$, $+$ indicates that the words are in different chunks, so the point between them is a chunk boundary. h_1 gives the probability of being a chunk boundary or not based on the collected frequencies. In the example of figure 1, we increment the $count(1|f_2, f_3)$, $count(1|f_5, f_6)$ and for all the other pairs $count(0|f_j, f_{j+1})$.

- $h_2(\delta, f_j)$, where $\delta \in \{1, 0\}$, 1 indicates the word is a left border of a chunk. In the example, f_1, f_3 and f_6.

- $h_3(\delta, f_j)$, where $\delta \in \{1, 0\}$, 1 indicates the word is a right border of a chunk. In the example, f_2, f_5 and f_7.

- $h_4(f_j, f_{j+1})$, which is a binary function indicating the significance of the pair in the data.

Given above feature functions, a first set of training sentences is used to collect the lexicalized frequencies and train the model, the second part is used to generate features for parameter estimation of the maximum entropy classifier. We use L-BFGS (Nocedal, 1980) implemented in (Le, 2004) to optimize the feature weights.

The chunking scorer is integrated into the baseline decoder as an additional feature. The feature function to integrate into Equation 2 is:

$$h_{chunk}(f_1^J, e_1^I, C, S) = \log \prod_1^J (C_j S(j) + (1 - C_j)(1 - S(j))) \quad (2)$$

where C is a function that maps each position on the foreign side to the set $\{1, 0\}$, indicating whether there is a chunk boundary after this word. S is the chunking scorer that assigns to each position the probability of being a chunk boundary.

3.2 Decoding by Chunking

The decoder is a multi-stack, multi-beam decoder that translates the sentence from left to right, which can skip multiple chunks and translate them later to perform any kind of re-ordering. For expanding each hypothesis either an uncovered chunk is picked and a phrase translation is applied or a new location is marked as a chunk boundary. As the chunking decisions affect the way phrase translations are applied, we insert hypotheses with the same covered words and the same last chunked position in the same stack. For expanding each hypothesis, the first step is to label more chunks from the last chunked position, which means expanding the current hypothesis by finding more chunks and assigning to them the chunking cost. In the next step, if the current position is inside an uncovered chunk, the decoder continues translating the chunk by applying new phrase translations. Otherwise, it picks a new chunk to translate and starts applying phrase translations within the chunk. No re-ordering inside the chunks is allowed.

Figure 2 shows an example of a chunk based derivation. In state 1 of this example, the decoder labels the position between German words 'muss' and 'die' as a chunk boundary. This is a chunking state (C), which finds the labels of the positions between the words and computes the chunking cost by the chunking scorer component. For the next state, the decoder either labels more positions to be chunked or applies phrase translations to uncovered words. The latter is done by translating the span 'man muss'. A translation state (P) can be reached by multiple phrase applications. In states 3 and 4, more positions are labeled as chunk boundaries (between 'wirkung', 'anerkennen' and 'anerkennen', '.'). In the next state, the decoder jumps over a chunk (9 words) to translate the verb. Grouping the words together makes it possible to do long-distance reordering such as this. The remainder of the decoding process is to translate the skipped chunk monotonically and finally chunk and translate the full stop.

1 C	[man muss] die schwierigkeiten bei der bestimmung von ursache und wirkung anerkennen .
2 P	[**man muss**] die schwierigkeiten bei der bestimmung von ursache und wirkung anerkennen . we must
3 C	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung] anerkennen . we must
4 C	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung][anerkennen] . we must
5 P	[man muss][die schwierigkeiten bei der bestimmung von ursache und wirkung][**anerkennen**] . we must recognise
6 P	[man muss][**die schwierigkeiten bei der bestimmung von ursache und wirkung**][anerkennen] . we must recognise the difficulties in the provision of cause and effect

Figure 2: An example of the decoding process by dynamic chunking. The *C* states are chunking states, which new chunking boundaries are detected and in *P* states, phrase translations are applied inside a chunk. The bold parts of the source sentence show the translated spans in that state. The rest of the decoding is chunking and translation the full stop.

With extra information in every hypothesis, the recombination criteria are redefined to consider the chunking status of a hypothesis. For two hypotheses to be recombinable (Koehn, 2004), they should have identical chunk boundaries for the uncovered positions. This is in addition to commonly used recombination criteria such as identical cover vectors, language model history, and last foreign position covered.

The chunking cost, estimated by the chunking scorer, is another feature along the baseline features. Also, the future cost computation component includes the future chunk distortion cost and future chunking cost together with the translation model and language model costs.

The following feature functions are defined to incorporate chunking costs and chunk re-orderings costs:

- Chunking cost feature function which assigns to each chunk a probability according to the classifier explained in the previous section.
- Chunking penalty which penalizes or rewards each chunking application based on the sign of its weight. The optimization algorithm, configures this feature in a way to encourage or discourage longer chunks.
- Chunk distortion model which penalizes jumps over chunks similar to distance-based reordering model, however instead of the number of words, it counts the number of chunks.

3.3 Parameters

To control the quality and the speed of the decoder for different language pairs, a few additional parameters are introduced. Since decoding inside the chunks is monotone, all baseline parameters[1] apart from the distortion limit are also needed here.

- chunk length limit: determines the maximum allowed length for each chunk. A large value, such as 100, lets the decoder try all available chunks. On the other hand, for languages with many local word re-orderings a smaller value can make the decoding process faster without hurting the performance (Default: 100).
- chunk number minimum and maximum: These values control the number of uncovered chunks before applying phrase applications. They can be used to control the amount of permutations during decoding (Default: 1 and unlimited).
- chunk distortion limit: similar to distortion limit in the baseline, but based on the chunks instead of words (Default: 6).

[1]This includes: stack limit, beam width, phrase length limit, and phrase table entries per source phrase.

		German	English
Train	Sentences	1.4M	
	Words	38M	40M
	Vocabulary	344K	113K
	Avg Sen. Length	26.17	27.51
Test(EP)	Sentences	2,000	
	Words	56K	60K
	Vocabulary	8844	6050
	Avg Sen. Length	28.31	30.09
Test(NC)	Sentences	2,028	
	Words	51K	49K
	Vocabulary	9849	7163
	Avg Sen. Length	25.31	24.63

Table 1: German to English corpus statistics. Europarl (EP) and News Commentary (NC) test sets of ACL WMT 2008.

4 Experiments

4.1 Experimental Setup

To examine the effects of dynamic chunking on translation quality, we have chosen German to English translation as it involves many long distance reorderings. The training and test data sets are taken from the ACL WMT evaluation (Koehn and Monz, 2006). The corpus statistics are shown in table 1.

The preprocessing stage includes tokenization and lower casing. There is only one reference translation for each sentence. The evaluation metrics used here are BLEU (Papineni et al., 2001), NIST (Doddington, 2002) and TER (Snover et al., 2006).

The baseline system is a common multi-beam, multi-stack phrase-based decoder, described in (Koehn et al., 2003) with following features:

- phrase translation probabilities and lexical probabilities for both directions
- a trigram language model
- phrase and word penalty
- distance-based re-ordering penalty

The weights for the features are optimized by MER training (Och, 2003) to maximize the BLEU (Papineni et al., 2001) score.

	Run	System	BLEU	NIST	1−TER
1	EP	Baseline	0.2687	7.0063	0.3374
2	EP	Chunk	0.2716	7.1084	0.3261
3	NC	Baseline	0.2454	7.1591	0.3476
4	NC	Chunk	0.2487	7.1798	0.3599

Table 2: Results on German to English task of ACL WMT 2008 translation task, Europarl (EP) and News Commentary (NC) test sets. Since TER is measuring the error, 1−TER is reported. Default values are used for parameters of the chunking decoder (see 3.3).

4.2 Results

The maximum entropy classifier is evaluated on the held-out data of the parallel corpus. The average accuracy[2] of 10-fold cross validation is 0.73, which means that around 25% of the chunk boundary decisions are incorrect. On the other hand, the classification decisions are not the only source of evidence that we use to choose the chunking boundaries. Both the language model and the translation models (phrases that cover the span) contribute to this decision. The probability of being a chunk boundary in the training data is 0.3, which is nearly identical to the probability of assigning a chunk boundary during the decoding. However, in 32% of the cases the chunking decision during decoding differs from the decision of the maximum entropy classifier. This means, even though the classifier classifies a point as a chunking boundary, the decoder decides not to use that chunking decision, mainly based on the translation and language model costs.

Table 2, shows the results of the chunking approach compared to the baseline. By looking at the translation outputs of the chunking system and comparing it to the baseline, we can observe that the chunking system generates very different translations to the baseline and not in all cases captures the proper order of the chunks to translate. In general, there are three main reasons for the chunking system to fail. Firstly, a wrong classification decision by the chunking scorer may lead the decoder to jump or monotonically translate in a wrong position. Secondly, although the classifier picks a proper chunking boundary, the other features force the decoder to

[2]The accuracy is computed based on how many of the boundary points are classified correctly. Note that, a sentence of length J, has $J-1$ boundary point.

apply the wrong re-ordering. Finally, even with accurate chunk boundaries, the decoder can still fail to apply the correct re-orderings.

5 Conclusion and Future Work

Inspired by previous work on integrating syntactic chunking into machine translation, a decoder that dynamically chunks and translates the source sentences is developed. The results show that the chunking system generates very different translations compared to the baseline and it is effective for a language pair such as German to English that needs long-distance re-orderings. Dealing with data sparseness and more accurate classification for detecting chunking boundaries seems very promising.

Although the current set of classification features is quite simple and it does not contain word classes or POS features, it performs well compared to the baseline. Incorporating more features and using word classes to deal with data sparseness could result in better classifier decisions and higher translation quality. It is not entirely surprising that the language model seems insufficient to accurately distinguish between correct and incorrect re-orderings of chunks in all cases. A lexicalized re-ordering model on the chunk-level could help to improve this aspect of our approach.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 529–536, Morristown, NJ, USA. Association for Computational Linguistics.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Honolulu, Hawaii, October. Association for Computational Linguistics.

Boxing Chen, Mauro Cettolo, and Marcello Federico. 2006. Reordering rules for phrase-based statistical machine translation. In *Proceeding of IWSLT 2006*, pages 53–58, Kyoto, Japan, November.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. 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. Association for Computational Linguistics.

Josep Maria Crego and Jose B. Marino. 2006. Improving statistical mt by coupling reordering and decoding. *Machine Translation*, 20(3):199–215, September.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 755–763.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session*, pages 177–180, Jun.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*, pages 115–124, Washington, District of Columbia.

Zhang Le. 2004. Maximum entropy modeling toolkit for python and C++. `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`.

Jorge Nocedal. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782.

CC	[sie kann nicht][als grundlage für die einführung einer europäischen verfassung][dienen][.]
CD	[1 sie kann nicht 1][3 als grundlage für die einführung einer europäischen verfassung 3] [2 dienen 2][4 . 4]
RE	it cannot serve as a basis for the establishment of a european constitution .
BL	it is not as a basis for the introduction of a european constitution .
CH	it cannot serve as a basis for the introduction of a european constitution .
CC	[ich weiß , dass es] [bezüglich des einen oder anderen änderungsantrags noch meinungsverschiedenheiten gibt][.]
CD	[1 ich weiß , dass es 1][4 bezüglich des einen oder anderen änderungsantrags 4] [3 noch meinungsverschiedenheiten 3][2 gibt 2][5 . 5]
RE	i know there are still differences of opinion on this or that amendment .
BL	i know that it is on the one or other amendment still differences of opinion .
CH	i know that there are still differences of opinion with regard to the one or other of the amendment .
CC	[ich möchte][frau gebhardt zu einer guten arbeit][gratulieren][.]
CD	[1 ich möchte 1][3 frau gebhardt zu einer guten arbeit 3][2 gratulieren 2][4 . 4]
RE	i would congratulate mrs gebhardt on a good piece of work .
BL	i would say to mrs gebhardt on a job well done .
CH	i would like to congratulate mrs gebhardt on a job well done .
CC	[die anstrengungen können nicht][von den erzeugern][allein][unternommen werden][.]
CD	[1 die anstrengungen können nicht 1][3 von den erzeugern allein 3][2 unternommen werden 2][4 . 4]
RE	efforts cannot be made by producers alone .
BL	the efforts made by producers alone cannot be done .
CH	the efforts cannot be done by producers alone .

Table 3: A few translation samples comparing the chunking-based decoder and the baseline. CC indicates the chunking decisions by the maximum entropy classifier. CD are the chunking boundaries picked by the decoder and their order of translation. RE is the English reference sentence. BL is the baseline output and CH is the chunking-based decoder output.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 737–745.

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

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 205, Morristown, NJ, USA.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting*, Rochester, NY, April.