Extending Memory-Based Machine Translation to Phrases

Maarten van Gompel
M.vanGompel@uvt.nl

Antal van den Bosch
Antal.vdnBosch@uvt.nl

Peter Berck
P.J.Berck@uvt.nl

ILK Research Group, Tilburg centre for Creative Computing
Tilburg University, Tilburg, The Netherlands

Abstract

We present a phrase-based extension to memory-based machine translation. This form of example-based machine translation employs lazy-learning classifiers to translate fragments of the source sentence to fragments of the target sentence. Source-side fragments consist of variable-length phrases in a local context of neighboring words, translated by the classifier to a target-language phrase. We compare three methods of phrase extraction, and present a new decoder that reassembles the translated fragments into one final translation. Results show that one of the proposed phrase-extraction methods—the one used in Moses—leads to a translation system that outperforms context-sensitive word-based approaches. The differences, however, are small, arguably because the word-based approaches already capture phrasal context implicitly due to their source-side and target-side context sensitivity.

1 Introduction

In characterising example-based machine translation, Somers [1] cites the common use of a collection of translations, and the process of matching new source-language sentences against stored source-language sentences in this collection. This matching, which may involve partial matching of fragments of sentences at the source side, leads to a selection of target-language fragments which are recombined and possibly post-processed to form the final translation.

Memory-based machine translation [2, 3, 4] (MBMT for short) is a form of example-based machine translation. A key characteristic of MBMT is the use of memory-based classifiers [5, 6] for the translation step. Memory-based classifiers do not only look up stored translation pairs, but are also able to generate translations when the input does not offer an exact match with a memorized translation pair. A parallel corpus serves as the main knowledge base. All sentences in this parallel corpus are tokenised and paired up with their counterparts, and between the words of each sentence pair, an alignment is computed. This alignment serves as the basis from which small fragments in their context can be extracted that are subsequently passed to a classifier for training. Whereas prior research in MBMT composed these fragments from single words in context [2, 4], the approach proposed in this study takes a phrase–one or more words—in context, as the focus element of each fragment.

We thus start from a mapping of fragments in the source language to fragments in the target language. Subsequently, memory-based learning is applied to convert these paired fragments into a memory-based classifier [5]. This classifier can then be used to translate new sentences. Given a sentence to translate, we segment this into various phrase-based fragments; for each fragment, a distribution of possible output fragment translations is predicted by the memory-based classifier. As a final step, all translations of these fragments are recombined by a new decoder that searches for a globally optimal translation of the given sentence.

The study builds upon previous research on MBMT [2, 3, 4]. The question addressed here is whether a phrase-based approach improves MBMT. An extension to phrases introduces non-trivial issues; one is how to detect phrases in a parallel training corpus. In the study described in this paper, three methods

of phrase extraction are tested and compared. Second, there is the issue of choosing a representation of variable-length phrases in a fixed-length feature vector used for memory-based classification.

In Section 2 we present our approaches to phrase-based MBMT in detail. In Section 3 the results of a comparative series of experiments are presented and discussed. We formulate our conclusions and starting points for future research in Section 4. Note that the system presented in this paper is available as open-source from `http://ilk.uvt.nl/mbmt/pbmbmt` .

2 Phrase-based memory based machine translation

An MBMT system divides into a training subsystem, producing a translation model, and a translation subsystem. A parallel corpus is used for phrase extraction and example generation, i.e. the generation of translations of source fragments to target fragments. These fragments, with as its main constituent an aligned pair of phrases, are compressed, rather than merely stored, in the training phase. The aim of compression into a tree structure is primarily to offer fast retrieval, but as a side effect memory needs are minimized as well. In testing, unseen source-language sentences in a test corpus are also transformed into fragments, which the memory-based classifier maps onto a distribution of target-language fragment translations. A decoder then reassembles all translated fragments together into one sentence, searching through and choosing between alternative solutions when more than a single target sentence can be built out of the predicted fragments.

2.1 Example generation

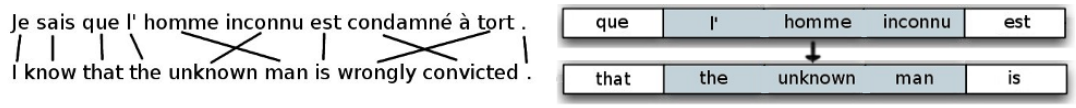

Figure 1: *Left:* A word alignment between a French and English sentence, *Right:* A phrase-based training example in context

We assume a word-alignment between all sentence pairs in the parallel corpus. Figure 1 (left) illustrates such a word-aligned sentence pair, serving as an example throughout this section. On the basis of this, we create example fragment translations that serve as training examples. On the input side, an example consists of a feature vector representing a source-language fragment; on the output side, the example is labeled with a class, representing a fragment of the target sentence aligning to the source fragment. In prior research [2, 4], the feature vector consisted of one focus word, one context word to the left, and one context word to the right; the class was composed of the target-language word aligned to the focus word, and again one context word to the left, and one to the right. Suppose we translate French to English and look at the word est in Figure 1 (left), then the feature vector would be (inconnu, est, condamné), and the class would be (man,is,wrongly). Note that the class is considered by the classifier as an atomic symbol, but it is decomposed later into its constituents by the decoder. By moving a sliding window over the source sentence, fragments can be generated for all words save for zero-fertility words.

The phrase-based approach we present here is similar. Examples are composed as follows: The feature vector consists of a phrase from the source sentence, with one context word on the left side, and one context word on the right side. The class consists of the target-language phrase that aligns to the source-language phrase, and can optionally also take left- and right context words. In this research however we found that taking no target-side context produced significantly better results. In representing the source-language side of the example as a feature vector, the focus can be coded into multiple features,

such as one per position relative to the focus. Since phrases can be of arbitrary length and the classifier expects a feature vector of fixed size, this poses a problem. Section 2.3 will address this issue further.

Suppose that the alignment links the source-language phrase *"l'homme inconnu"* to the target-language phrase *"the unknown man"* in the target language. Figure 1 (right) illustrates the phrase-based training example generated for this aligned pair of phrases.

2.2 Phrase extraction

A first task is how to determine phrases in the source- and target-language sentences in the parallel corpus available for training. One solution is to employ the same type of method as used in phrase-based statistical machine translation, making use of a phrase-translation table [7]. Such a table lists aligned phrases in both source and target language, assigning conditional probabilities for each. These aligned phrase pairs are computed statistically over the entire parallel corpus by taking the intersection of source-to-target and target-to-source word alignments [7], and can be extended by incrementally adding points from the union of the two alignments [8]. We use the implementation in Moses [9] to this end.

In addition to this first method, we include two other approaches to phrase extraction for comparison. The second method of phrase extraction, henceforth named the *phrase-list approach*, is a straightforward method that extracts frequent n-grams only from the source-language side of the training corpus, and stores this in what we call a *phrase list*. The approach needs a frequency threshold above which an n-gram is included in the phrase list, which after exploratory experimentation was fixed at 25. Unlike in the phrase-table method, the aligned counterpart of a source-side phrase is computed on the fly. Each source sentence is matched against the phrase list, and whenever a phrase is found, we follow the word-alignments from the phrase and assume that the sequence of words it points to is the aligned target phrase, possibly with intervening fertility words.

Using phrases from either a phrase-translation table or phrase-list, we can never expect to obtain full coverage of test sentences. To decrease problems of low coverage and data sparsity, we defined a phrase to consist of *one or more* words. In addition to phrase extraction, we always generate word-based fragments using the same word-oriented approach as used in prior research. This makes the phrase-based approach an extension of the word-based approach. Given the same parallel corpus and input sentences, the training and test examples in phrase-based MBMT are a superset of those in word-based MBMT.

Due to the phrase-based character of our approach, a word in the source sentence can be part of the focus of a feature vector multiple times. A single word always generates its word-based example, but there may also be one or more extracted phrases that the word is a part of. We may thus generate multiple examples that all contain the same word or words as part of their focus. Phrase overlap occurs in both training and test examples. The latter has an important side-effect that will have an impact on the decoding process we describe later on. If there are multiple examples covering the same words, then there will be multiple possible fragmentations of the input sense (see also Figure 2).

2.2.1 Marker-based chunking

The third phrase-extraction method is *marker-based chunking*, which segments a sentence into non-overlapping chunks, splitting whenever so-called marker words occur. Marker words are typically defined as closed-class function words, and overlap significantly with the top-most frequent words in most corpora. A chunk must contain at least one non-marker word at its end. The idea behind marker-based chunking is rooted in the Marker Hypothesis [10], an idea from psycho-linguistics that posits that all languages are marked for surface syntax by a specific closed set of lexemes or morphemes [3].

Marker-based chunking is a phrase extraction strategy that differs from the previous two in the sense that it does not use data statistics on n-grams. It has already been employed in a previous study of MBMT

[3], inspired in turn by its earlier application in EBMT [11, 12]. In this study we also use the method and compare it to the previous two methods.

2.3 Representing phrase-based examples

If we encode the focus phrase of the feature vector in terms of its encoding words and their position in the phrase, we end up with feature vectors of different sizes. However, the memory-based classifier demands a fixed number of features in order to compute its similarity function. There are at least three ways to resolve this problem. First, we can consider the phrase as one atomic feature. Second, we can reserve a fixed number n of features, and fill those with position-specific words (such as the final word, the prefinal word, etc.), assigning dummy values to unused feature slots. Third, we can assign separate classifiers to different phrase lengths, assigning examples with a particular phrase-length to a separate classifier trained only on examples of this length. In this setup a master process assigns examples to different classifiers and reassembles their output again for the decoder.

2.4 Decoding

Due to the overlapping nature of extractable phrases, and the fact that we may end up with multiple examples covering the same words in the source sentence, we can speak of various possible *fragmentations* of the source sentence S. We define a fragmentation to be a chain in which the focus parts are non-overlapping; each fragment covers a certain range of consecutive words of arbitrary length n in the source sentence S, where $1 \leq n \leq |S|$. In addition each fragment is associated with a left context and right context of a length predetermined during example generation. Figure 2 shows three example fragmentations.

Each test example is mapped by the classifier to a distribution of classes, which are the various target-side translations for the fragment and an associated probability score. From the perspective of the decoder these are referred to as *hypothesis fragments*. Thus, each fragment will be associated with a collection of one or more hypothesis fragments; the associated scores denote the translation probability for the particular fragment being translated to the particular hypothesis fragment. Figure 2 illustrates the relation between the fragmentation of a sample sentence, the source-side fragments that are extracted from it, and the target-side hypothesis fragments generated from the source-side fragments.

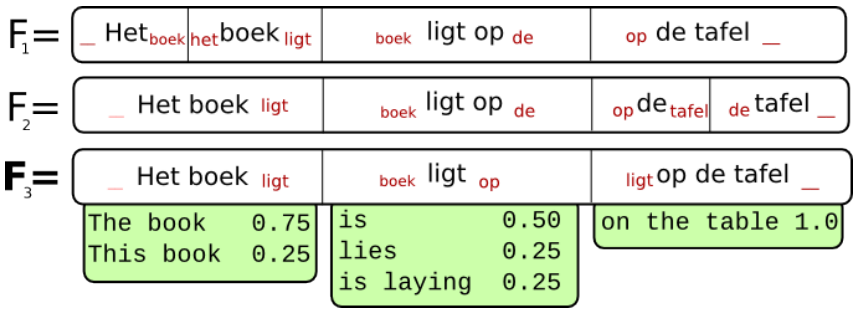

Figure 2: Fragmentations of the sample Dutch input sentence *"Het boek ligt op de tafel"* (The book is on the table). The third fragmentation is expanded to list the target-side hypothesis fragments associated with each of the three source-side fragments the fragmentation is composed of. Context information is printed in small text.

Having gathered all matching fragments for a given source sentence, the task is to search for "good" fragmentations, leading to the most likely translation. The number of fragmentations tends to increase exponentially in the length of the source sentence. Therefore it proved to be an infeasible approach to

generate all fragmentations in an exhaustive fashion. Instead, we attempt to select a number of good fragmentations in a local beam search, and decode only on the basis of the hypothesis fragments generated on the basis of this selection of source-side fragmentations.

For each fragmentation returned by the local beam search, the decoding procedure is started, which itself employs another local beam search. It should be noted that this makes the phrase-based approach computationally more expensive compared to the word-based approach, as the latter by definition only has one fragmentation of the source sentence. The decoder returns a list of the highest-scoring translation hypotheses for each fragmentation, limited in number by the beam size of the decoder.

The decoding procedure starts by generating an initial hypothesis: a translation hypothesis in which we simply select for each fragment in the fragmentation the hypothesis fragment with the highest translation probability. We order the hypothesis fragments for the initial hypothesis in the order we find the fragments in the source-sentence fragmentation. The initial hypothesis in Figure 2 thus is *"The book is on the table"*. In this example, the initial hypothesis already happens to generate the best translation, but in most cases there is more searching to do. A hypothesis can be modified in two main ways: (1) the order in which the hypothesis fragments are assembled can be changed, and (2) the choice of hypothesis fragments can be changed, i.e. other hypothesis fragments with an equal or lower translation probability could be tried. To this end, the decoder applies two operations to the initial hypothesis. Each yields new hypotheses and the best few, limited by the beamsize, are selected. To these hypotheses the operators are applied again. This procedure repeats itself until no better scoring hypotheses can be generated. The first operator is **substitution**. It generates new hypotheses in which a hypothesis fragment of a particular fragment is substituted by another hypothesis fragment from the list. This is done exhaustively. For each fragment, substitutions are made using all hypothesis fragments that have not undergone a substitution operation in a previous decoding round. The second operator is the **swap** operation, in which we swap the location of two hypothesis fragments. This again is done in an exhaustive fashion such that all possible swaps are made. Each fragment swaps places with all neighboring fragments within a certain range, each swap yielding a new hypothesis. An extra parameter specifies the maximum range over which a swap can occur; we set this at two, allowing swaps with neighboring fragments and their immediate neighbors.

The success of the decoding algorithm depends on the score function it maximises. For each hypothesis, a score is computed that is an expression of the quality of the hypothesis. A good translation should preferably maximise both *fidelity* and *fluency*. These two components are present in the decoder developed for the present study. A quantification of fluency is provided by a trigram-based statistical language model with back-off smoothing on the target language, while fidelity is expressed by the probabilities generated by the memory-based translation model. More precisely, the score function for a hypothesis H is made up of the product of the translation probability and distortion score of the given hypothesis:

$$\text{TranslationModel}(H) = \text{ClassifierScore}(H) \cdot \text{DistScore}(H) \tag{1}$$

$$\text{ClassifierScore}(H) = \prod_{i=1}^{|H|} P(classification_i^{weight}) \tag{2}$$

$$\text{DistScore}(H) = \prod_{i=1}^{|H|} distortion\_constant^{distance(hfragment_i, hfragment_{i-1})} \tag{3}$$

We see here an expression of the translation probability of a hypothesis. This can be computed by taking the product of the translation probabilities of all selected hypothesis fragments that make up the hypothesis (Equation 2). Recall that these translation probabilities come directly from the classifier output, which predicted a distribution of hypotheses fragments as illustrated in Figure 2. The translation probabilities can be given an extra weight parameter by raising them to a certain power.

In addition, a distortion score DistScore(H) is also computed by raising a distortion constant to the power of a measure of distance between two fragments in the original source sentence (Equation 3). This is an admittedly crude factor that may fit the language pair, but will tend to favor ungrammatical and undistorted target sequences over grammatical but reordered target sequences.

3 Results

Experiments were performed on two parallel corpora, in which we focused only on Dutch to English translation. The first is OpenSubtitles [13], which consists of user-contributed subtitles for movies. The used training set consists of 286,160 sentence pairs. The second corpus is EMEA [13], a medical and largely formulaic text corpus represented by a training set of 871,180 sentence pairs. From both corpora a development and test set of 1,000 sentences each was selected.

First, several simple parameter optimisation experiments were tried in order to assess the effect of the decoder and some of the parameters. One outcome of these explorations is that omitting target-side context, as used in prior research where target-side fragments constituted trigrams of words [2, 4], greatly improves results. In an experiment on the OpenSubtitles corpus, a BLEU score of 0.1211 with target-side context, rises to 0.2184 when target-side context is removed and only target-side phrases are predicted. We attribute this effect mainly to the increase in sparsity of classes when adding context, adding to the sparsity of the phrases themselves.

The main question addressed in this study is whether a phrase-based approach to MBMT (PBMBMT) improves upon the previous word-based approaches (MBMT [2], CSIMT [4]). In both these methods, the feature vector as well as the class consist of a trigram, one focus word, one left-context word, and one right-context word. CSIMT successfully employs a more powerful decoder based on Constraint Satisfaction Inference, hence we compare against this variant. For completeness, we also compare to the MBMT variant described in [2] that maps source trigrams to output trigrams, and uses a decoder that is purely based on target trigram overlap. This overlap-based decoder was introduced in [3], where it was shown to outperform a marker-based variant of MBMT.

Other questions addressed in this study are: How do the three phrase-extraction methods perform? What example format is best? Indications for answering these questions can be found in Table 1, which shows the main results. Note that in this table, the PBMBMT decoder was also run without using any phrase-extraction method (named wb-PBMBMT), making it operate on a word-based level like in word-oriented CSIMT, with the notable difference that target-side context is excluded in all PBMBMT experiments. This word-based system offers a baseline for assessing the effectiveness of the phrase-extraction methods, and it can be compared to the word-based decoders reported in earlier work [2, 4]. Note that in further comparisons with phrase-based SMT systems, the Moses system was found to attain BLEU scores of 0.33 on OpenSubtitles, and 0.47 on EMEA, clearly outperforming our best scores.

With respect to the three phrase extraction methods, the Moses [9] phrase-table method performs best overall. The other two methods, especially marker-based chunking, perform below the word-based PBMBMT baseline. This may be attributed to the fact that the phrase-translation table is computed using the two-way alignment statistics of the parallel corpus, whilst the other two methods only rely on source-side statistics, and the aligned counterparts of the phrases are sought in an ad-hoc and per-sentence fashion. The two predecessor systems, MBMT [2] and CSIMT [4], are both outperformed by the Moses phrase-table method, and as reported earlier in [4], the CSIMT method tends to outperform the MBMT method with the trigram overlap-based decoder on all metrics. On the EMEA corpus, CSIMT outperforms the word-based PBMBMT, and performs relatively close to PBMBMT with the Moses phrase-table approach.

We thus observe that the advantage of phrase-based MBMT compared to earlier word-oriented ap-

OpenSubtitles

Decoder	Extraction mMethod	Single / multi classifier	Translation performance metrics BLEU	NIST	METEOR	WER	PER
MBMT	-	-	0.1631	4.243	0.3835	68.39	61.33
CSIMT	-	-	0.2002	4.750	0.4431	68.42	55.18
wb-PBMBMT	-	-	0.2163	**5.136**	**0.4644**	55.23	**48.22**
PBMBMT	phrase table	single	**0.2300**	5.055	0.4623	54.47	49.18
PBMBMT	phrase table	multi	0.2256	5.004	0.4583	55.28	49.74
PBMBMT	phrase list	single	0.2190	4.980	0.4543	**54.09**	48.77
PBMBMT	phrase list	multi	0.2184	4.975	0.4529	**54.09**	48.79
PBMBMT	marker-based	single	0.1003	2.935	0.3057	76.79	71.16
PBMBMT	marker-based	multi	0.1394	3.360	0.3437	66.40	62.38

EMEA

Decoder	Extraction method	Multi or single classifier	Translation performance metrics BLEU	NIST	METEOR	WER	PER
MBMT	-	-	0.2533	5.115	0.4801	72.78	63.66
CSIMT	-	-	0.3013	5.938	0.5333	63.00	**50.85**
wb-PBMBMT	-	-	0.2715	5.600	0.5381	65.99	57.25
PBMBMT	phrase table	single	0.3075	6.011	**0.5455**	59.00	52.02
PBMBMT	phrase table	multi	**0.3078**	**6.019**	0.5449	**58.76**	51.63
PBMBMT	phrase list	single	0.2440	5.352	0.4946	62.74	56.67
PBMBMT	phrase list	multi	0.2440	5.378	0.4967	62.82	56.86
PBMBMT	marker-based	multi	0.2370	4.612	0.4513	74.37	66.78

Table 1: Main results on the **OpenSubtitles** and **EMEA** corpora, Dutch to English.

proaches, including the closest comparable system, the word-based PBMBMT baseline (wb-PBMBMT in the table) that restricts itself to words and uses the same decoder as PBMBMT, turns out to be limited. This is a surprising outcome. We may posit that sparsity plays a role here; phrases are by definition less prevalent than single words, which complicates the classification process. The omission of context in classes (in contrast to CSIMT, which maps to trigrams of words) attempts to compensate for this to a certain extent. Another reason for the lack of a clear difference between word-based and phrase-based MBMT may be sought in the fact that even in word-oriented CSIMT there is already a significant but implicit role for phrasal context. Essentially we are comparing *phrasal context inherent to the phrases themselves* in PBMBMT, against *phrasal context implicit in the input and output word trigrams* in CSIMT. Often, such as with phrases of length 3, the two approaches are mapping about the same input to the same output. The limited gain of the phrase-based approach may stem from the added value of the fact that PBMBMT is not restricted to trigrams, and can vary between whatever is the strongest n-gram.

Concerning the example format, reserving space for a fixed number of position-specific features (i.e. words) in a single classifier versus distributing different phrase-lengths over multiple classifiers perform more or less on a par.

4 Conclusions and future research

The study described in this paper has demonstrated how memory-based machine translation can be extended from translating fixed-length word trigrams to translating phrases of arbitrary length. We compared three methods of phrase extraction, of which the Moses phrase-translation table approach emerges as the best solution.

Prior research in MBMT such as the recent CSIMT approach [4] relied partly on target-side context, making use of the overlap between predicted target-side fragments (word trigrams) in decoding. The current study shows that ignoring target-side context produces significantly better results in a phrase-based approach. This can be credited to the decrease in sparsity in the output class space. Moreover,

removing this context can be justified by the fact that context becomes less relevant in phrase-based approaches, as target-side phrases capture enough internal context themselves.

Nevertheless, the impact of phrases in comparison to word-based MBMT has been shown to be limited. A potential explanation for this limited effect is that earlier word-based MBMT approaches can be seen as implicitly phrase-based already. The approach followed in [2, 4] maps trigrams of source-side words to trigrams of target-side words, implicitly capturing all phrases up to length three. In this perspective, our current approach changes this only slightly by turning the source-side trigrams into variable-width examples of Moses phrases surrounded by their left and right neighboring words, and predicting variable-width target-side phrases at the output, starting from single words.

Besides further hyperparameter optimisation of the memory-based classifier and the inclusion of richer (e.g. linguistic) features, we think that most improvement can be obtained by improving the decoder. In future work it could be extended with more operations, such as a delete operation powered by a null model; moreover, an alternative should be sought for its current crude distortion factor. The findings with regard to omission of target-side context could be tested and incorporated into the strategy proposed in CSIMT [4]. Future work should also focus on different language pairs and datasets.

References

[1] H. Somers. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157, 1999.

[2] A. van den Bosch and P. Berck. Memory-based machine translation and language modeling. *The Prague Bulletin of Mathematical Linguistics*, 91:17–26, 2009.

[3] A. van den Bosch, N. Stroppa, and A. Way. A memory-based classification approach to marker-based EBMT. In *Proceedings of the METIS-II Workshop on New Approaches to Machine Translation*, pages 63–72, 2007.

[4] S. Canisius and A. van den Bosch. A constraint satisfaction approach to machine translation. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-2009)*, pages 182–189, 2009.

[5] W. Daelemans, A. van den Bosch, and T. Weijters. Igtree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):407–423, 1997.

[6] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. Timbl: Tilburg memory based learner, version 6.1, reference guide. Technical Report ILK-07-07, Tilburg University, Tilburg, The Netherlands, 2007.

[7] P. Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In R.E. Frederking and K. Taylor, editors, *Proceedings of the American Machine Translation Association*, volume 3265 of *Lecture Notes in Computer Science*, pages 115–124. Springer, 2004.

[8] F.J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics, 2007.

[10] T. Green. The necessity of syntax markers. two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496, 1979.

[11] N. Gough and A. Way. Robust large-scale EBMT with marker-based segmentation. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2004)*, pages 95–104, Baltimore, Maryland, 2004.

[12] A. Way and N. Gough. Comparing example-based and statistical machine translation. *Natural Language Engineering*, 11(3):295–309, 2005.

[13] J. Tiedemann and L. Nygaard. The OPUS corpus - parallel and free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC04)*, pages 26–28, 2004.