Translating with non-contiguous phrases

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman,
Eric Gaussier, Cyril Goutte, Kenji Yamada
Xerox Research Centre Europe
FirstName.FamilyName@xrce.xerox.com

Philippe Langlais
RALI/DIRO Université de Montréal
felipe@iro.umontreal.ca

Arne Mauser
RWTH Aachen University
arne.mauser@rwth-aachen.de

Abstract

This paper presents a phrase-based statistical machine translation method, based on non-contiguous phrases, i.e. phrases with *gaps*. A method for producing such phrases from a word-aligned corpora is proposed. A statistical translation model is also presented that deals such phrases, as well as a training method based on the maximization of translation accuracy, as measured with the NIST evaluation metric. Translations are produced by means of a beam-search decoder. Experimental results are presented, that demonstrate how the proposed method allows to better generalize from the training data.

1 Introduction

Possibly the most remarkable evolution of recent years in statistical machine translation is the step from word-based models to phrase-based models (Och et al., 1999; Marcu and Wong, 2002; Yamada and Knight, 2002; Tillmann and Xia, 2003). While in traditional word-based statistical models (Brown et al., 1993) the atomic unit that translation operates on is the word, phrase-based methods acknowledge the significant role played in language by multi-word expressions, thus incorporating in a statistical framework the insight behind Example-Based Machine Translation (Somers, 1999).

However, Phrase-based models proposed so far only deal with multi-word units that are sequences of contiguous words on both the source and the target side. We propose here a model designed to deal with multi-word expressions that need not be contiguous in either or both the source and the target side.

The rest of this paper is organised as follows. Section 2 provides motivations, definition and extraction procedure for non-contiguous phrases. The log-linear conditional translation model we adopted is the object of Section 3; the method used to train its parameters is described in Section 4. Section 5 briefly describes the decoder. The experiments we conducted to asses the effectiveness of using non-contiguous phrases are presented in Section 6.

2 Non-contiguous phrases

Why should it be a good thing to use phrases composed of possibly non-contiguous sequences of words? In doing so we expect to improve translation quality by better accounting for additional linguistic phenomena as well as by extending the effect of contextual semantic disambiguation and example-based translation inherent in phrase-based MT. An example of a phenomenon best described using non-contiguous units is provided by English phrasal verbs. Consider the sentence "Mary *switches* her table lamp *off*". Word-based statistical models would be at odds when selecting the appropriate translation of the verb. If French were the target language, for instance, corpus evidence would come from both examples in which "switch" is translated as "allumer" (to switch on) and as "éteindre" (to switch off). If many-to-one word alignments are not allowed from English to French, as it is usually the

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 755–762, Vancouver, October 2005.

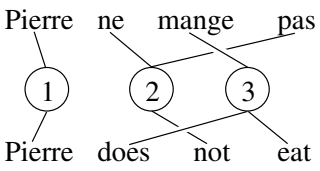

Figure 1: An example of a complex alignment associated with different syntax for negation in English and French.

case, then the best thing a word-based model could do in this case would be to align "off" to the empty word and hope to select the correct translation from "switch" only, basically a 50-50 bet. While handling inseparable phrasal verbs such as "to run out" correctly, previously proposed phrase-based models would be helpless in this case. A comparable behavior is displayed by German separable verbs. Moreover, non-contiguous linguistic units are not limited to verbs. Negation is formed, in French, by inserting the words "ne" and "pas" before and after a verb respectively. So, the sentence "Pierre ne mange pas" and its English translation display a complex word-level alignment (Figure 1) current models cannot account for.

Flexible idioms, allowing for the insertion of linguistic material, are other phenomena best modeled with non-contiguous units.

2.1 Definition and library construction

We define a *bi-phrase* as a pair comprising a *source phrase* and a *target phrase*: $b = \langle \tilde{s}, \tilde{t} \rangle$. Each of the source and target phrases is a sequence of words and gaps (indicated by the symbol $\diamond$); each gap acts as a placeholder for exactly one unspecified word. For example, $\tilde{w} = w_1 w_2 \diamond w_3 \diamond\diamond w_4$ is a phrase of length 7, made up of two contiguous words w_1 and w_2, a first gap, a third word w_3, two consecutive gaps and a final word w_4. To avoid redundancy, phrases may not begin or end with a gap. If a phrase does not contain any gaps, we say it is *contiguous*; otherwise it is *non-contiguous*. Likewise, a bi-phrase is said to be *contiguous* if both its phrases are contiguous.

The translation of a source sentence s is produced by combining together bi-phrases so as to cover the source sentence, and produce a well-formed target-language sentence (i.e. without gaps). A complete translation for s can be described as an ordered sequence of bi-phrases $b_1 ... b_K$. When piecing together the final translation, the target-language portion $\tilde{t}_1$ of the first bi-phrase b_1 is first layed down, then each subsequent $\tilde{t}_k$ is positioned on the first "free" position in the target language sentence, i.e. either the leftmost gap, or the right end of the sequence. Figure 2 illustrates this process with an example.

To produce translations, our approach therefore relies on a collection of bi-phrases, what we call a *bi-phrase library*. Such a library is constructed from a corpus of existing translations, aligned at the word level.

Two strategies come to mind to produce non-contiguous bi-phrases for these libraries. The first is to align the words using a "standard" word alignement technique, such as the *Refined Method* described in (Och and Ney, 2003) (the intersection of two IBM Viterbi alignments, forward and reverse, enriched with alignments from the union) and then generate bi-phrases by combining together individual alignments that co-occur in the same pair of sentences. This is the strategy that is usually adopted in other phrase-based MT approaches (Zens and Ney, 2003; Och and Ney, 2004). Here, the difference is that we are not restricted to combinations that produce strictly contiguous bi-phrases.

The second strategy is to rely on a word-alignment method that naturally produces many-to-many alignments between non-contiguous words, such as the method described in (Goutte et al., 2004). By means of a matrix factorization, this method produces a parallel partition of the two texts, seen as sets of word tokens. Each token therefore belongs to one, and only one, subset within this partition, and corresponding subsets in the source and target make up what are called *cepts*. For example, in Figure 1, these cepts are represented by the circles numbered 1, 2 and 3; each cept thus connects word tokens in the source and the target, regardless of position or contiguity. These cepts naturally constitute bi-phrases, and can be used directly to produce a bi-phrase library.

Obviously, the two strategies can be combined, and it is always possible to produce increasingly large and complex bi-phrases by combining together co-occurring bi-phrases, contiguous or not. One problem with this approach, however, is that the resulting libraries can become very large. With con-

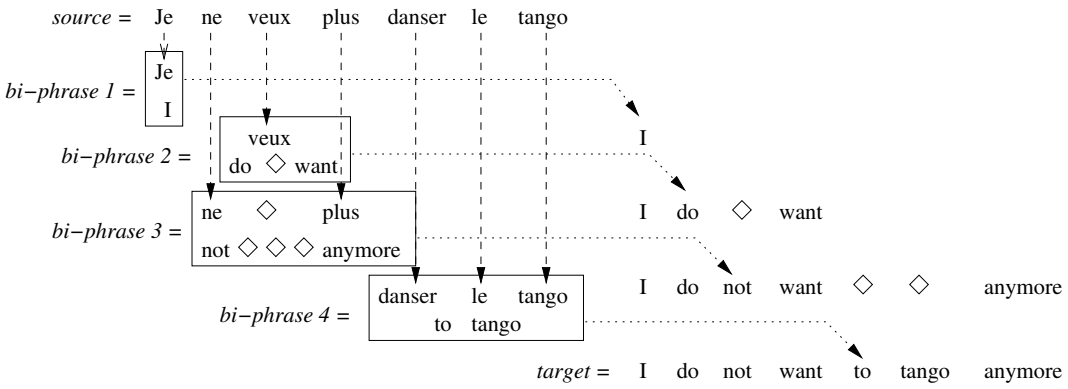

Figure 2: Combining bi-phrases to produce a translation.

tiguous phrases, the number of bi-phrases that can be extracted from a single pair of sentences typically grows quadratically with the size of the sentences; with non-contiguous phrases, however, this growth is exponential. As it turns out, the number of available bi-phrases for the translation of a sentence has a direct impact on the time required to compute the translation; we will therefore typically rely on various filtering techniques, aimed at keeping only those bi-phrases that are more likely to be useful. For example, we may retain only the most frequently observed bi-phrases, or impose limits on the number of cepts, the size of gaps, etc.

3 The Model

In statistical machine translation, we are given a source language input $s_1^J = s_1 ... s_J$, and seek the target-language sentence $t_1^I = t_1 ... t_I$ that is its most likely translation:

$$\hat{t}_1^I = \text{argmax}_{t_1^I} Pr(t_1^I | s_1^J) \quad (1)$$

Our approach is based on a direct approximation of the posterior probability $Pr(t_1^I | s_1^J)$, using a log-linear model:

$$Pr(t_1^I | s_1^J) = \frac{1}{Z_{s_1^J}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(t_1^I, s_1^J)\right)$$

In such a model, the contribution of each *feature function* h_m is determined by the corresponding model parameter λ_m; $Z_{s_1^J}$ denotes a normalization constant. This type of model is now quite widely used for machine translation (Tillmann and Xia, 2003; Zens and Ney, 2003)[1].

Additional variables can be introduced in such a model, so as to account for hidden characteristics, and the feature functions can be extended accordingly. For example, our model must take into account the actual set of bi-phrases that was used to produce this translation:

$$Pr(t_1^I, b_1^K | s_1^J) = \frac{1}{Z_{s_1^J}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(t_1^I, s_1^J, b_1^K)\right)$$

Our model currently relies on seven feature functions, which we describe here.

- The *bi-phrase* feature function h_{bp}: it represents the probability of producing t_1^I using some set of bi-phrases, under the assumption that each source phrase produces a target phrase independently of the others:

$$h_{bp}(t_1^I, s_1^J, b_1^K) = \sum_{k=1}^{K} \log Pr(\tilde{t}_k | \tilde{s}_k) \quad (2)$$

  Individual bi-phrase probabilities $Pr(\tilde{t}_k | \tilde{s}_k)$ are estimated based on occurrence counts in the word-aligned training corpus.

- The *compositional bi-phrase* feature function h_{comp}: this is introduced to compensate for

[1]Recent work from Chiang (Chiang, 2005) addresses similar concerns to those motivating our work by introducing a Synchronous CFG for bi-phrases. If on one hand SCFGs allow to better control the order of the material inserted in the gaps, on the other gap size does not seem to be taken into account, and phrase dovetailing such as the one involving "do ⋄want" and "not ⋄⋄⋄anymore" in Fig. 2 is disallowed.

h_{bp}'s strong tendency to overestimate the probability of rare bi-phrases; it is computed as in equation (2), except that bi-phrase probabilities are computed based on individual word translation probabilities, somewhat as in IBM model 1 (Brown et al., 1993):

$$Pr(\tilde{t}|\tilde{s}) = \frac{1}{|\tilde{s}||\tilde{t}|} \prod_{t \in \tilde{t}} \sum_{s \in \tilde{s}} Pr(t|s)$$

- The *target language* feature function h_{tl}: this is based on a N-gram language model of the target language. As such, it ignores the source language sentence and the decomposition of the target into bi-phrases, to focus on the actual sequence of target-language words produced by the combination of bi-phrases:

$$h_{tl}(t_1^I, s_1^J, b_1^K) = \sum_{i=1}^{I} \log Pr(t_i|t_{i-N+1}^{i-1})$$

- The *word-count* and *bi-phrase count* feature functions h_{wc} and h_{bc}: these control the length of the translation and the number of bi-phrases used to produce it:

$$h_{wc}(t_1^I, s_1^J, b_1^K) = I \quad h_{bc}(t_1^I, s_1^J, b_1^K) = K$$

- The *reordering* feature function $h_{reord}(t_1^I, s_1^J, b_1^K)$: it measures the amount of reordering between bi-phrases of the source and target sentences.

- the *gap count* feature function h_{gc}: It takes as value the total number of gaps (source and target) within the bi-phrases of b_1^K, thus allowing the model some control over the nature of the bi-phrases it uses, in terms of the discontiguities they contain.

4 Parameter Estimation

The values of the λ parameters of the log-linear model can be set so as to optimize a given criterion. For instance, one can maximize the likelyhood of some set of training sentences. Instead, and as suggested by Och (2003), we chose to maximize directly the quality of the translations produced by the system, as measured with a machine translation evaluation metric.

Say we have a set of source-language sentences S. For a given value of λ, we can compute the set of corresponding target-language translations T. Given a set of *reference* ("gold-standard") translations R for S and a function $E(T, R)$ which measures the "error" in T relative to R, then we can formulate the parameter estimation problem as[2]:

$$\hat{\lambda} = \text{argmin}_{\lambda} E(T, R)$$

As pointed out by Och, one notable difficulty with this approach is that, because the computation of T is based on an argmax operation (see eq. 1), it is not continuous with regard to λ, and standard gradient-descent methods cannot be used to solve the optimization. Och proposes two workarounds to this problem: the first one relies on a direct optimization method derived from Powell's algorithm; the second introduces a smoothed (continuous) version of the error function $E(T, R)$ and then relies on a gradient-based optimization method.

We have opted for this last approach. Och shows how to implement it when the error function can be computed as the sum of errors on individual sentences. Unfortunately, this is not the case for such widely used MT evaluation metrics as BLEU (Papineni et al., 2002) and NIST (Doddington, 2002). We show here how it can be done for NIST; a similar derivation is possible for BLEU.

The NIST evaluation metric computes a weighted n-gram precision between T and R, multiplied by a factor $B(S, T, R)$ that penalizes short translations. It can be formulated as:

$$B(S, T, R) \times \sum_{n=1}^{N} \frac{\sum_{s \in S} I_n(t_s, r_s)}{\sum_{s \in S} C_n(t_s)} \qquad (3)$$

where N is the largest n-gram considered (usually $N = 4$), $I_n(t_s, r_s)$ is a weighted count of common n-grams between the target (t_s) and reference (r_s) translations of sentence s, and $C_n(t_s)$ is the total number of n-grams in t_s.

To derive a version of this formula that is a continuous function of λ, we will need multiple translations $t_{s,1}, ..., t_{s,K}$ for each source sentence s. The general idea is to weight each of these translations

[2]For the sake of simplicity, we consider a single reference translation per source sentence, but the argument can easily be extended to multiple references.

by a factor $w(\lambda, s, k)$, proportional to the score $m_\lambda(t_{s,k}|s)$ that $t_{s,k}$ is assigned by the log-linear model for a given λ:

$$w(\lambda, s, k) = \left[\frac{m_\lambda(t_{s,k}|s)}{\sum_{k'} m_\lambda(t_{s,k'}|s)} \right]^\alpha$$

where α is the *smoothing factor*. Thus, in the smoothed version of the NIST function, the term $I_n(t_s, r_s)$ in equation (3) is replaced by $\sum_k w(\lambda, s, k) I_n(t_{s,k}, r_s)$, and the term $C_n(t_s)$ is replaced by $\sum_k w(\lambda, s, k) C_n(t_{s,k})$. As for the brevity penalty factor $B(S, T, R)$, it depends on the total length of translation T, i.e. $\sum_s |t_s|$. In the smoothed version, this term is replaced by $\sum_s \sum_k w(\lambda, s, k)|t_{s,k}|$. Note that, when $\alpha \to \infty$, then $w(\lambda, s, k) \to 0$ for all translations of s, except the one for which the model gives the highest score, and so the smooth and normal NIST functions produce the same value. In practice, we determine some "good" value for α by trial and error (5 works fine).

We thus obtain a scoring function for which we can compute a derivative relative to λ, and which can be optimized using gradient-based methods. In practice, we use the *OPT++* implementation of a quasi-Newton optimization (Meza, 1994). As observed by Och, the smoothed error function is not convex, and therefore this sort of minimum-error rate training is quite sensitive to the initialization values for the λ parameters. Our approach is to use a random set of initializations for the parameters, perform the optimization for each initialization, and select the model which gives the overall best performance.

Globally, parameter estimation proceeds along these steps:

1. Initialize the training set: using random parameter values λ_0, for each source sentence of some given set of sentences S, we compute multiple translations. (In practice, we use the M-best translations produced by our decoder; see Section 5).

2. Optimize the parameters: using the method described above, we find λ that produces the best smoothed NIST score on the training set.

3. Iterate: we then re-translate the sentences of S with this new λ, combine the resulting multiple translations with those already in the training set, and go back to step 2.

Steps 2 and 3 can be repeated until the smooothed NIST score does not increase anymore[3].

5 Decoder

We implemented a version of the beam-search stack decoder described in (Koehn, 2003), extended to cope with non-contiguous phrases. Each translation is the result of a sequence of *decisions*, each of which involves the selection of a bi-phrase and of a target position. The final result is obtained by combining decisions, as in Figure 2. *Hypotheses*, corresponding to partial translations, are organised in a sequence of priority stacks, one for each number of source words covered. Hypotheses are extended by filling the first available uncovered position in the target sentence; each extended hypotheses is then inserted in the stack corresponding to the updated number of covered source words. Each hypothesis is assigned a score which is obtained as a combination of the actual feature function values and of admissible heuristics, adapted to deal with gaps in phrases, estimating the future cost for completing a translation. Each stack undergoes both threshold and histogram pruning. Whenever two hypotheses are indistinguishable as far as the potential for further extension is concerned, they are merged and only the highest-scoring is further extended. Complete translations are eventually recovered in the "last" priority stack, i.e. the one corresponding to the total number of source words: the best translation is the one with the highest score, and that does not have any remaining gaps in the target.

6 Evaluation

We have conducted a number of experiments to evaluate the potential of our approach. We were particularly interested in assessing the impact of non-contiguous bi-phrases on translation quality, as well as comparing the different bi-phrase library contruction strategies evoked in Section 2.1.

[3]It can be seen that, as the set of possible translations for S stabilizes, we eventually reach a point where the procedure converges to a maximum. In practice, however, we can usually stop much earlier.

6.1 Experimental Setting

All our experiments focused exclusively on French to English translation, and were conducted using the Aligned Hansards of the 36th Parliament of Canada, provided by the Natural Language Group of the USC Information Sciences Institute, and edited by Ulrich Germann. From this data, we extracted three distinct subcorpora, which we refer to as the *bi-phrase-building set*, the *training set* and the *test set*. These were extracted from the so-called *training*, *test-1* and *test-2* portions of the Aligned Hansard, respectively. Because of efficiency issues, we limited ourselves to source-language sentences of 30 words or less. More details on the evaluation data is presented in Table 1[4].

6.2 Bi-phrase Libraries

From the bi-phrase-building set, we built a number of libraries. A first family of libraries was based on a word alignment "A", produced using the *Refined method* described in (Och and Ney, 2003) (combination of two IBM-Viterbi alignments): we call these the A libraries. A second family of libraries was built using alignments "B" produced with the method in (Goutte et al., 2004): these are the B libraries. The most notable difference between these two alignments is that B contains "native" non-contiguous bi-phrases, while A doesn't.

Some libraries were built by simply extracting the cepts from the alignments of the bi-phrase-building corpus: these are the A^1 and B^1 libraries, and variants. Other libraries were obtained by combining cepts that co-occur within the same pair of sentences, to produce "composite" bi-phrases. For instance, the A^2 libraries contain combinations of 1 or 2 cepts from alignment A; B^3 contains combinations of 1, 2 or 3 cepts, etc.

Some libraries were built using a "gap-size" filter. For instance library A^2-g3 contains those bi-phrases obtained by combining 1 or 2 cepts from alignment A, and in which neither the source nor the target phrase contains more than 3 gaps. In particular, library B^1-g0 does not contain any non-contiguous bi-phrases.

[4]Preliminary experiments on different data sets allowed us to establish that 800 sentences constituted an acceptable size for estimating model parameters. With such a corpus, the estimation procedure converges after just 2 or 3 iterations.

Finally, all libraries were subjected to the same two filtering procedures: the first excludes all bi-phrases that occur only once in the training corpus; the second, for any given source-language phrase, retains only the 20 most frequent target-language equivalents. While the first of these filters typically eliminates a large number of entries, the second only affects the most frequent source phrases, as most phrases have less than 20 translations.

6.3 Experiments

The parameters of the model were optimized independantly for each bi-phrase library. In all cases, we performed only 2 iterations of the training procedure, then measured the performance of the system on the test set in terms of the NIST and BLEU scores against one reference translation. As a point of comparison, we also trained an IBM-4 translation model with the *GIZA++* toolkit (Och and Ney, 2000), using the combined *bi-phrase building* and *training* sets, and translated the test set using the *ReWrite* decoder (Germann et al., 2001)[5].

Table 2 describes the various libraries that were used for our experiments, and the results obtained for each.

System/library	bi-phrases	NIST	BLEU
ReWrite		6.6838	0.3324
A^1	238 K	6.6695	0.3310
A^2-g0	642 K	6.7675	0.3363
A^2-g3	4.1 M	6.7068	0.3283
B^1-g0	193 K	6.7898	0.3369
B^1	267 K	6.9172	0.3407
B^2-g0	499 K	6.7290	0.3391
B^2-g3	3.3 M	6.9707	0.3552
B^1-g1	206 K	6.8979	0.3441
B^1-g2	213 K	6.9406	0.3454
B^1-g3	218 K	6.9546	0.3518
B^1-g4	222 K	6.9527	0.3423

Table 2: Bi-phrase libraries and results

The top part of the table presents the results for the A libraries. As can be seen, library A^1 achieves approximately the same score as the baseline system; this is expected, since this library is essentially

[5]Both the *ReWrite* and our own system relied on a trigram language model trained on the English half of the bi-phrase building set.

Subset	sentences	source words	target words
bi-phrase-building set	931,000	17.2M	15.2M
training set	800	11,667	10,601
test set	500	6726	6041

Table 1: Data sets.

made up of one-to-one alignments computed using IBM-4 translation models. Adding contiguous bi-phrases obtained by combining pairs of alignments does gain us some mileage (+0.1 NIST)[6]. Again, this is consistent with results observed with other systems (Tillmann and Xia, 2003). However, the addition of non-contiguous bi-phrases (A^2-g3) does not seem to help.

The middle part of Table 2 presents analogous results for the corresponding B libraries, plus the B^1-g0 library, which contains only those cepts from the B alignment that are contiguous. Interestingly, in the experiments reported in (Goutte et al., 2004), alignment method B did not compare favorably to A under the widely used *Alignment Error Rate* (AER) metric. Yet, the B^1-g0 library performs better than the analogous A^1 library on the translation task. This suggests that AER may not be an appropriate metric to measure the potential of an alignment for phrase-based translation.

Adding non-contiguous bi-phrases allows another small gain. Again, this is interesting, as it suggests that "native" non-contiguous bi-phrases are indeed useful for the translation task, i.e. those non-contiguous bi-phrases obtained directly as cepts in the B alignment.

Surprisingly, however, combining cepts from the B alignment to produce contiguous bi-phrases (B^2-G0) does not turn out to be fruitful. Why this is so is not obvious and, certainly, more experiments would be required to establish whether this tendency continues with larger combinations (B^3-g0, B^4-g0...). Composite non-contiguous bi-phrases produced with the B alignments (B^2-g3) seem to bring improvements with regard to "basic" bi-phrases (B_1), but it is not clear whether these are significant.

[6]While the differences in scores in these and other experiments are relatively small, we believe them to be significant, as they have been confirmed systematically in other experiments and, in our experience, by visual inspection of the translations.

Visual examination of the B^1 library reveals that many non-contiguous bi-phrases contain long-spanning phrases (i.e. phrases containing long sequences of gaps). To verify whether or not these were really useful, we tested a series of B^1 libraries with different gap-size filters. It must be noted that, because of the final histogram filtering we apply on libraries (retain only the 20 most frequent translations of any source phrase), library B^1-g1 is not a strict subset of B^1-g2. Therefore, filtering on gap-size usually represents a tradeoff between more frequent long-spanning bi-phrases and less frequent short-spanning ones.

The results of these experiments appear in the lower part of Table 2. While the differences in score are small, it seems that concentrating on bi-phrases with 3 gaps or less affords the best compromise. For small libraries such as those under consideration here, this sort of filtering may not be very important. However, for higher-order libraries (B^2, B^3, etc.) it becomes crucial, because it allows to control the exponential growth of the libraries.

7 Conclusions

In this paper, we have proposed a phrase-based statistical machine translation method based on non-contiguous phrases. We have also presented a estimation procedure for the parameters of a log-linear translation model, that maximizes a smooth version of the NIST scoring function, and therefore lends itself to standard gradient-based optimization techniques.

From our experiments with these new methods, we essentially draw two conclusions. The first and most obvious is that non-contiguous bi-phrases can indeed be fruitful in phrase-based statistical machine translation. While we are not yet able to characterize which bi-phrases are most helpful, some of those that we are currently capable of extracting are well suited to cover some short-distance phenomena.

The second conclusion is that alignment quality is crucial in producing good translations with phrase-based methods. While this may sound obvious, our experiments shed some light on two specific aspects of this question. The first is that the alignment method that produces the most useful bi-phrases need not be the one with the best *alignment error rate* (AER). The second is that, depending on the alignments one starts with, constructing increasingly large bi-phrases does not necessarily lead to better translations. Some of our best results were obtained with relatively small libraries (just over 200,000 entries) of short bi-phrases. In other words, it's not how many bi-phrases you have, it's how good they are. This is the line of research that we intend to pursue in the near future.

Acknowledgments

The authors are grateful to the anonymous reviewers for their useful suggestions. [7]

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270, Ann Arbor, Michigan.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. ARPA Workshop on Human Language Technology*.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast Decoding and Optimal Decoding for Machine Translation. In *Proceedings of ACL 2001*, Toulouse, France.

Cyril Goutte, Kenji Yamada, and Eric Gaussier. 2004. Aligning words using matrix factorisation. In *Proc. ACL'04*, pages 503–510.

Philipp Koehn. 2003. *Noun Phrase Translation*. Ph.D. thesis, University of Southern California.

[7]This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP 02)*, Philadelphia, PA.

J. C. Meza. 1994. OPT++: An Object-Oriented Class Library for Nonlinear Optimization. Technical Report SAND94-8225, Sandia National Laboratories, Albuquerque, USA, March.

F. J. Och and H. Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL 2000*, pages 440–447, Hongkong, China, October.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.

Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VCL 99)*, College Park, MD.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *ACL'03: 41st Ann. Meet. of the Assoc. for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evalution of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318, Philadelphia, USA.

Harold Somers. 1999. Review Article: Example-based Machine Translation. *Machine Translation*, 14:113–157.

Christoph Tillmann and Fei Xia. 2003. A phrase-based unigram model for statistical machine translation. In *Proc. of the HLT-NAACL 2003 Conference*, Edmonton, Canada.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proc. of the 40th Annual Conf. of the Association for Computational Linguistics (ACL 02)*, Philadelphia, PA.

Richard Zens and Hermann Ney. 2003. Improvements in Phrase-Based Statistical Machine Translation. In *Proc. of the HLT-NAACL 2003 Conference*, Edmonton, Canada.