Statistical Phrase-Based Translation

Philipp Koehn, Franz Josef Och, Daniel Marcu
Information Sciences Institute
Department of Computer Science
University of Southern California
koehn@isi.edu, och@isi.edu, marcu@isi.edu

Abstract

We propose a new phrase-based translation model and decoding algorithm that enables us to evaluate and compare several, previously proposed phrase-based translation models. Within our framework, we carry out a large number of experiments to understand better and explain why phrase-based models outperform word-based models. Our empirical results, which hold for all examined language pairs, suggest that the highest levels of performance can be obtained through relatively simple means: heuristic learning of phrase translations from word-based alignments and lexical weighting of phrase translations. Surprisingly, learning phrases longer than three words and learning phrases from high-accuracy word-level alignment models does not have a strong impact on performance. Learning only syntactically motivated phrases degrades the performance of our systems.

1 Introduction

Various researchers have improved the quality of statistical machine translation system with the use of phrase translation. Och et al. [1999]'s alignment template model can be reframed as a phrase translation system; Yamada and Knight [2001] use phrase translation in a syntax-based translation system; Marcu and Wong [2002] introduced a joint-probability model for phrase translation; and the CMU and IBM word-based statistical machine translation systems[1] are augmented with phrase translation capability.

Phrase translation clearly helps, as we will also show with the experiments in this paper. But what is the best method to extract phrase translation pairs? In order to investigate this question, we created a uniform evaluation framework that enables the comparison of different ways to build a phrase translation table.

Our experiments show that high levels of performance can be achieved with fairly simple means. In fact, for most of the steps necessary to build a phrase-based system, tools and resources are freely available for researchers in the field. More sophisticated approaches that make use of syntax do not lead to better performance. In fact, imposing syntactic restrictions on phrases, as used in recently proposed syntax-based translation models [Yamada and Knight, 2001], proves to be harmful. Our experiments also show, that small phrases of up to three words are sufficient for obtaining high levels of accuracy.

Performance differs widely depending on the methods used to build the phrase translation table. We found extraction heuristics based on word alignments to be better than a more principled phrase-based alignment method. However, what constitutes the best heuristic differs from language pair to language pair and varies with the size of the training corpus.

[1]Presentations at DARPA IAO Machine Translation Workshop, July 22-23, 2002, Santa Monica, CA

2 Evaluation Framework

In order to compare different phrase extraction methods, we designed a uniform framework. We present a phrase translation model and decoder that works with any phrase translation table.

2.1 Model

The phrase translation model is based on the noisy channel model. We use Bayes rule to reformulate the translation probability for translating a foreign sentence $\mathbf{f}$ into English $\mathbf{e}$ as

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p(\mathbf{e})$$

This allows for a language model $p(\mathbf{e})$ and a separate translation model $p(\mathbf{f}|\mathbf{e})$.

During decoding, the foreign input sentence $\mathbf{f}$ is segmented into a sequence of I phrases $\bar{f}_1^I$. We assume a uniform probability distribution over all possible segmentations.

Each foreign phrase $\bar{f}_i$ in $\bar{f}_1^I$ is translated into an English phrase $\bar{e}_i$. The English phrases may be reordered. Phrase translation is modeled by a probability distribution $\phi(\bar{f}_i|\bar{e}_i)$. Recall that due to the Bayes rule, the translation direction is inverted from a modeling standpoint.

Reordering of the English output phrases is modeled by a relative distortion probability distribution $d(a_i - b_{i-1})$, where a_i denotes the start position of the foreign phrase that was translated into the ith English phrase, and b_{i-1} denotes the end position of the foreign phrase translated into the $(i-1)$th English phrase.

In all our experiments, the distortion probability distribution $d(\cdot)$ is trained using a joint probability model (see Section 3.3). Alternatively, we could also use a simpler distortion model $d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|}$ with an appropriate value for the parameter α.

In order to calibrate the output length, we introduce a factor ω for each generated English word in addition to the trigram language model p_{LM}. This is a simple means to optimize performance. Usually, this factor is larger than 1, biasing longer output.

In summary, the best English output sentence $\mathbf{e}_{\text{best}}$ given a foreign input sentence $\mathbf{f}$ according to our model is

$$\begin{aligned}\mathbf{e}_{\text{best}} &= \text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) \\ &= \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p_{\text{LM}}(\mathbf{e}) \omega^{\text{length}(\mathbf{e})}\end{aligned}$$

where $p(f|e)$ is decomposed into

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

For all our experiments we use the same training data, trigram language model [Seymore and Rosenfeld, 1997], and a specialized decoder.

2.2 Decoder

The phrase-based decoder we developed for purpose of comparing different phrase-based translation models employs a beam search algorithm, similar to the one by Jelinek [1998]. The English output sentence is generated left to right in form of partial translations (or hypotheses).

We start with an initial empty hypothesis. A new hypothesis is expanded from an existing hypothesis by the translation of a phrase as follows: A sequence of untranslated foreign words and a possible English phrase translation for them is selected. The English phrase is attached to the existing English output sequence. The foreign words are marked as translated and the probability cost of the hypothesis is updated.

The cheapest (highest probability) final hypothesis with no untranslated foreign words is the output of the search.

The hypotheses are stored in stacks. The stack s_m contains all hypotheses in which m foreign words have been translated. We recombine search hypotheses as done by Och et al. [2001]. While this reduces the number of hypotheses stored in each stack somewhat, stack size is exponential with respect to input sentence length. This makes an exhaustive search impractical.

Thus, we prune out weak hypotheses based on the cost they incurred so far and a future cost estimate. For each stack, we only keep a beam of the best n hypotheses. Since the future cost estimate is not perfect, this leads to search errors. Our future cost estimate takes into account the estimated phrase translation cost, but not the expected distortion cost.

We compute this estimate as follows: For each possible phrase translation anywhere in the sentence (we call it a *translation option*), we multiply its phrase translation probability with the language model probability for the generated English phrase. As language model probability we use the unigram probability for the first word, the bigram probability for the second, and the trigram probability for all following words.

Given the costs for the translation options, we can compute the estimated future cost for any sequence of consecutive foreign words by dynamic programming. Note that this is only possible, since we ignore distortion costs. Since there are only $n(n+1)/2$ such sequences for a foreign input sentence of length n, we can pre-compute these cost estimates beforehand and store them in a table.

During translation, future costs for uncovered foreign words can be quickly computed by consulting this table. If a hypothesis has broken sequences of untranslated foreign words, we look up the cost for each sequence and take the product of their costs.

The beam size, e.g. the maximum number of hypotheses in each stack, is fixed to a certain number. The number of translation options is linear with the sentence length. Hence, the time complexity of the beam search is quadratic with sentence length, and linear with the beam size.

Since the beam size limits the search space and therefore search quality, we have to find the proper trade-off between speed (low beam size) and performance (high beam size). For our experiments, a beam size of only 100 proved to be sufficient. With larger beams sizes, only few sentences are translated differently. With our decoder, translating 1755 sentence of length 5-15 words takes about 10 minutes on a 2 GHz Linux system. In other words, we achieved fast decoding, while ensuring high quality.

3 Methods for Learning Phrase Translation

We carried out experiments to compare the performance of three different methods to build phrase translation probability tables. We also investigate a number of variations. We report most experimental results on a German-English translation task, since we had sufficient resources available for this language pair. We confirm the major points in experiments on additional language pairs.

As the first method, we learn phrase alignments from a corpus that has been word-aligned by a training toolkit for a word-based translation model: the Giza++ [Och and Ney, 2000] toolkit for the IBM models [Brown et al., 1993]. The extraction heuristic is similar to the one used in the alignment template work by Och et al. [1999].

A number of researchers have proposed to focus on the translation of phrases that have a linguistic motivation [Yamada and Knight, 2001; Imamura, 2002]. They only consider word sequences as phrases, if they are constituents, i.e. subtrees in a syntax tree (such as a noun phrase). To identify these, we use a word-aligned corpus annotated with parse trees generated by statistical syntactic parsers [Collins, 1997; Schmidt and Schulte im Walde, 2000].

The third method for comparison is the joint phrase model proposed by Marcu and Wong [2002]. This model learns directly a phrase-level alignment of the parallel corpus.

3.1 Phrases from Word-Based Alignments

The Giza++ toolkit was developed to train word-based translation models from parallel corpora. As a by-product, it generates word alignments for this data. We improve this alignment with a number of heuristics, which are described in more detail in Section 4.5.

We collect all aligned phrase pairs that are consistent with the word alignment: The words in a legal phrase pair are only aligned to each other, and not to words outside [Och et al., 1999].

Given the collected phrase pairs, we estimate the phrase translation probability distribution by relative frequency:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f},\bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f},\bar{e})}$$

No smoothing is performed.

3.2 Syntactic Phrases

If we collect all phrase pairs that are consistent with word alignments, this includes many non-intuitive phrases. For instance, translations for phrases such as "house the" may be learned. Intuitively we would be inclined to believe that such phrases do not help: Restricting possible phrases to syntactically motivated phrases could filter out such non-intuitive pairs.

Another motivation to evaluate the performance of a phrase translation model that contains only syntactic phrases comes from recent efforts to built syntactic translation models [Yamada and Knight, 2001; Wu, 1997]. In these models, reordering of words is restricted to reordering of constituents in well-formed syntactic parse trees. When augmenting such models with phrase translations, typically only translation of phrases that span entire syntactic subtrees is possible. It is important to know if this is a helpful or harmful restriction.

Consistent with Imamura [2002], we define a syntactic phrase as a word sequence that is covered by a single subtree in a syntactic parse tree.

We collect syntactic phrase pairs as follows: We word-align a parallel corpus, as described in Section 3.1. We then parse both sides of the corpus with syntactic parsers [Collins, 1997; Schmidt and Schulte im Walde, 2000]. For all phrase pairs that are consistent with the word alignment, we additionally check if both phrases are subtrees in the parse trees. Only these phrases are included in the model.

Hence, the syntactically motivated phrase pairs learned are a subset of the phrase pairs learned without knowledge of syntax (Section 3.1).

As in Section 3.1, the phrase translation probability distribution is estimated by relative frequency.

3.3 Phrases from Phrase Alignments

Marcu and Wong [2002] proposed a translation model that assumes that lexical correspondences can be established not only at the word level, but at the phrase level as well. To learn such correspondences, they introduced a phrase-based joint probability model that simultaneously generates both the Source and Target sentences in a parallel corpus. Expectation Maximization learning in Marcu and Wong's framework yields both (i) a joint probability distribution $\phi(\bar{e}, \bar{f})$, which reflects the probability that phrases $\bar{e}$ and $\bar{f}$ are translation equivalents; (ii) and a joint distribution $d(i, j)$, which reflects the probability that a phrase at position i is translated into a phrase at position j. To use this model in the context of our framework, we simply marginalize to conditional probabilities the joint probabilities estimated by Marcu and Wong [2002]. Note that this approach is consistent with the approach taken by Marcu and Wong themselves, who use conditional models during decoding.

Method	Training corpus size					
	10k	20k	40k	80k	160k	320k
AP	84k	176k	370k	736k	1536k	3152k
Joint	125k	220k	400k	707k	1254k	2214k
Syn	19k	24k	67k	105k	217k	373k

Table 1: Size of the phrase translation table in terms of distinct phrase pairs (maximum phrase length 4)

4 Experiments

We used the freely available Europarl corpus [2] to carry out experiments. This corpus contains over 20 million words in each of the eleven official languages of the European Union, covering the proceedings of the European Parliament 1996-2001. 1755 sentences of length 5-15 were reserved for testing.

In all experiments in Section 4.1-4.6 we translate from German to English. We measure performance using the BLEU score [Papineni et al., 2001], which estimates the accuracy of translation output with respect to a reference translation.

4.1 Comparison of Core Methods

First, we compared the performance of the three methods for phrase extraction head-on, using the same decoder (Section 2) and the same trigram language model. Figure 1 displays the results.

In direct comparison, learning all phrases consistent with the word alignment (AP) is superior to the joint model (Joint), although not by much. The restriction to only syntactic phrases (Syn) is harmful. We also included in the figure the performance of an IBM Model 4 word-based translation system (M4), which uses a greedy decoder [Germann et al., 2001]. Its performance is worse than both AP and Joint. These results are consistent over training corpus sizes from 10,000 sentence pairs to 320,000 sentence pairs. All systems improve with more data.

Table 1 lists the number of distinct phrase translation pairs learned by each method and each corpus. The number grows almost linearly with the training corpus size, due to the large number of singletons. The syntactic restriction eliminates over 80% of all phrase pairs.

Note that the millions of phrase pairs learned fit easily into the working memory of modern computers. Even the largest models take up only a few hundred megabyte of RAM.

4.2 Weighting Syntactic Phrases

The restriction on syntactic phrases is harmful, because too many phrases are eliminated. But still, we might suspect, that these lead to more reliable phrase pairs.

[2]The Europarl corpus is available at http://www.isi.edu/~koehn/europarl/

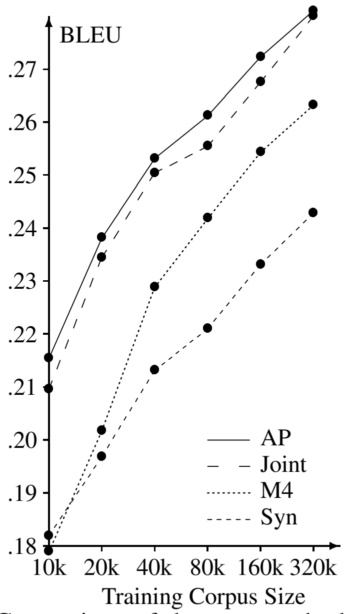

Figure 1: Comparison of the core methods: all phrase pairs consistent with a word alignment (AP), phrase pairs from the joint model (Joint), IBM Model 4 (M4), and only syntactic phrases (Syn)

One way to check this is to use all phrase pairs and give more weight to syntactic phrase translations. This can be done either during the data collection – say, by counting syntactic phrase pairs twice – or during translation – each time the decoder uses a syntactic phrase pair, it credits a bonus factor to the hypothesis score.

We found that neither of these methods result in significant improvement of translation performance. Even penalizing the use of syntactic phrase pairs does not harm performance significantly. These results suggest that requiring phrases to be syntactically motivated does not lead to better phrase pairs, but only to fewer phrase pairs, with the loss of a good amount of valuable knowledge.

One illustration for this is the common German “es gibt”, which literally translates as “it gives”, but really means “there is”. “Es gibt” and “there is” are not syntactic constituents. Note that also constructions such as “with regard to” and “note that” have fairly complex syntactic representations, but often simple one word translations. Allowing to learn phrase translations over such sentence fragments is important for achieving high performance.

4.3 Maximum Phrase Length

How long do phrases have to be to achieve high performance? Figure 2 displays results from experiments with different maximum phrase lengths. All phrases consistent with the word alignment (AP) are used. Surprisingly, limiting the length to a maximum of only three words

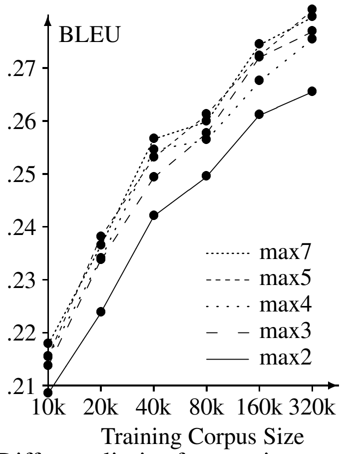

Figure 2: Different limits for maximum phrase length show that length 3 is enough

Max. Length	Training corpus size					
	10k	20k	40k	80k	160k	320k
2	37k	70k	135k	250k	474k	882k
3	63k	128k	261k	509k	1028k	1996k
4	84k	176k	370k	736k	1536k	3152k
5	101k	215k	459k	925k	1968k	4119k
7	130k	278k	605k	1217k	2657k	5663k

Table 2: Size of the phrase translation table with varying maximum phrase length limits

per phrase already achieves top performance. Learning longer phrases does not yield much improvement, and occasionally leads to worse results. Reducing the limit to only two, however, is clearly detrimental.

Allowing for longer phrases increases the phrase translation table size (see Table 2). The increase is almost linear with the maximum length limit. Still, none of these model sizes cause memory problems.

4.4 Lexical Weighting

One way to validate the quality of a phrase translation pair is to check, how well its words translate to each other. For this, we need a lexical translation probability distribution $w(f|e)$. We estimated it by relative frequency from the same word alignments as the phrase model.

$$w(f|e) = \frac{\text{count}(f,e)}{\sum_{f'} \text{count}(f',e)}$$

A special English NULL token is added to each English sentence and aligned to each unaligned foreign word.

Given a phrase pair $\bar{f}, \bar{e}$ and a word alignment a between the foreign word positions $i = 1, ..., n$ and the English word positions $j = 0, 1, ..., m$, we compute the lexical weight p_w by

$$p_w(\bar{f}|\bar{e}, a) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(f_i|e_j)$$

```
       f1 f2 f3
NULL   -- -- ##
  e1   ## -- --
  e2   -- ## --
  e3   -- ## --
```

$$\begin{aligned} p_w(\bar{f}|\bar{e}, a) &= p_w(f_1 f_2 f_3 | e_1 e_2 e_3, a) \\ &= w(f_1|e_1) \\ &\quad \times \frac{1}{2}(w(f_2|e_2) + w(f_2|e_3)) \\ &\quad \times w(f_3|\text{NULL}) \end{aligned}$$

Figure 3: Lexical weight p_w of a phrase pair $(\bar{f}, \bar{e})$ given an alignment a and a lexical translation probability distribution $w(\cdot)$

See Figure 3 for an example.

If there are multiple alignments a for a phrase pair $(\bar{f}, \bar{e})$, we use the one with the highest lexical weight:

$$p_w(\bar{f}|\bar{e}) = \max_a p_w(\bar{f}|\bar{e}, a)$$

We use the lexical weight p_w during translation as a additional factor. This means that the model $p(f|e)$ is extended to

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1}) p_w(\bar{f}_i|\bar{e}_i, a)^\lambda$$

The parameter λ defines the strength of the lexical weight p_w. Good values for this parameter are around 0.25.

Figure 4 shows the impact of lexical weighting on machine translation performance. In our experiments, we achieved improvements of up to 0.01 on the BLEU score scale. Again, all phrases consistent with the word alignment are used (Section 3.1).

Note that phrase translation with a lexical weight is a special case of the alignment template model [Och et al., 1999] with one word class for each word. Our simplification has the advantage that the lexical weights can be factored into the phrase translation table beforehand, speeding up decoding. In contrast to the beam search decoder for the alignment template model, our decoder is able to search all possible phrase segmentations of the input sentence, instead of choosing one segmentation before decoding.

4.5 Phrase Extraction Heuristic

Recall from Section 3.1 that we learn phrase pairs from word alignments generated by Giza++. The IBM Models that this toolkit implements only allow at most one English word to be aligned with a foreign word. We remedy this problem with a heuristic approach.

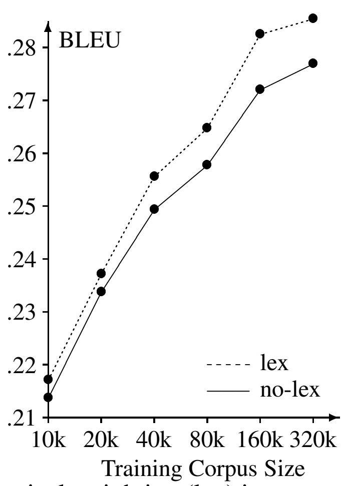

Figure 4: Lexical weighting (lex) improves performance

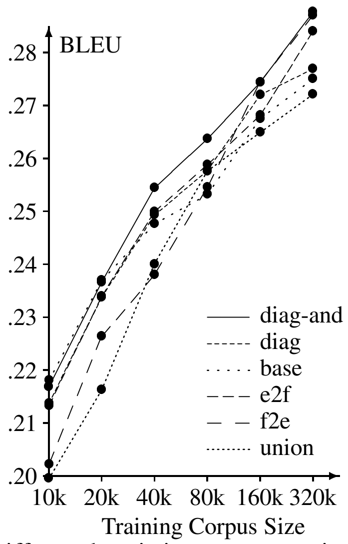

Figure 5: Different heuristics to symmetrize word alignments from bidirectional Giza++ alignments

First, we align a parallel corpus bidirectionally – foreign to English and English to foreign. This gives us two word alignments that we try to reconcile. If we intersect the two alignments, we get a high-precision alignment of high-confidence alignment points. If we take the union of the two alignments, we get a high-recall alignment with additional alignment points.

We explore the space between intersection and union with expansion heuristics that start with the intersection and add additional alignment points. The decision which points to add may depend on a number of criteria:

- In which alignment does the potential alignment point exist? Foreign-English or English-foreign?
- Does the potential point neighbor already established points?
- Does "neighboring" mean directly adjacent (block-distance), or also diagonally adjacent?
- Is the English or the foreign word that the potential point connects unaligned so far? Are both unaligned?
- What is the lexical probability for the potential point?

The base heuristic [Och et al., 1999] proceeds as follows: We start with intersection of the two word alignments. We only add new alignment points that exist in the union of two word alignments. We also always require that a new alignment point connects at least one previously unaligned word.

First, we expand to only directly adjacent alignment points. We check for potential points starting from the top right corner of the alignment matrix, checking for alignment points for the first English word, then continue with alignment points for the second English word, and so on. This is done iteratively until no alignment point can be added anymore. In a final step, we add non-adjacent alignment points, with otherwise the same requirements.

Figure 5 shows the performance of this heuristic (base) compared against the two mono-directional alignments (e2f, f2e) and their union (union). The figure also contains two modifications of the base heuristic: In the first (diag) we also permit diagonal neighborhood in the iterative expansion stage. In a variation of this (diag-and), we require in the final step that both words are unaligned.

The ranking of these different methods varies for different training corpus sizes. For instance, the alignment f2e starts out second to worst for the 10,000 sentence pair corpus, but ultimately is competitive with the best method at 320,000 sentence pairs. The base heuristic is initially the best, but then drops off.

The discrepancy between the best and the worst method is quite large, about 0.02 BLEU. For almost all training corpus sizes, the heuristic diag-and performs best, albeit not always significantly.

4.6 Simpler Underlying Word-Based Models

The initial word alignment for collecting phrase pairs is generated by symmetrizing IBM Model 4 alignments. Model 4 is computationally expensive, and only approximate solutions exist to estimate its parameters. The IBM Models 1-3 are faster and easier to implement. For IBM Model 1 and 2 word alignments can be computed efficiently without relying on approximations. For more information on these models, please refer to Brown et al. [1993]. Again, we use the heuristics from the Section 4.5 to reconcile the mono-directional alignments obtained through training parameters using models of increasing complexity.

How much is performance affected, if we base word alignments on these simpler methods? As Figure 6 indi-

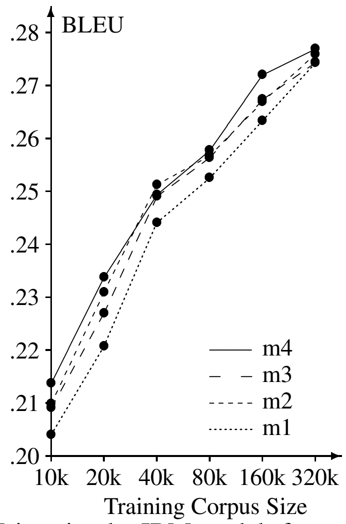

Figure 6: Using simpler IBM models for word alignment does not reduce performance much

Language Pair	Model4	Phrase	Lex
English-German	0.2040	0.2361	0.2449
French-English	0.2787	0.3294	0.3389
English-French	0.2555	0.3145	0.3247
Finnish-English	0.2178	0.2742	0.2806
Swedish-English	0.3137	0.3459	0.3554
Chinese-English	0.1190	0.1395	0.1418

Table 3: Confirmation of our findings for additional language pairs (measured with BLEU)

cates, not much. While Model 1 clearly results in worse performance, the difference is less striking for Model 2 and 3. Using different expansion heuristics during symmetrizing the word alignments has a bigger effect.

We can conclude from this, that high quality phrase alignments can be learned with fairly simple means. The simpler and faster Model 2 provides similar performance to the complex Model 4.

4.7 Other Language Pairs

We validated our findings for additional language pairs. Table 3 displays some of the results. For all language pairs the phrase model (based on word alignments, Section 3.1) outperforms IBM Model 4. Lexicalization (Lex) always helps as well.

5 Conclusion

We created a framework (translation model and decoder) that enables us to evaluate and compare various phrase translation methods. Our results show that phrase translation gives better performance than traditional word-based methods. We obtain the best results even with small phrases of up to three words. Lexical weighting of phrase translation helps.

Straight-forward syntactic models that map constituents into constituents fail to account for important phrase alignments. As a consequence, straight-forward syntax-based mappings do not lead to better translations than unmotivated phrase mappings. This is a challenge for syntactic translation models.

It matters how phrases are extracted. The results suggest that choosing the right alignment heuristic is more important than which model is used to create the initial word alignments.

References

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

Collins, M. (1997). Three generative, lexicalized models for statistical parsing. In *Proceedings of ACL 35*.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceedings of ACL 39*.

Imamura, K. (2002). Application of translation knowledge acquired by hierarchical phrase alignment for pattern-based mt. In *Proceedings of TMI*.

Jelinek, F. (1998). *Statistical Methods for Speech Recognition*. The MIT Press.

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of ACL 38*.

Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28.

Och, F. J., Ueffing, N., and Ney, H. (2001). An efficient A* search algorithm for statistical machine translation. In *Data-Driven MT Workshop*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176(W0109-022), IBM Research Report.

Schmidt, H. and Schulte im Walde, S. (2000). Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of COLING*.

Seymore, K. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of Eurospeech*.

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

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of ACL 39*.