Iterative refinement of lexicon and phrasal alignment

Jae Dong Kim
Language Technologies Institute
5000 Forbes av.
Pittsburgh, PA 15213
jdkim@cs.cmu.edu

Stephan Vogel
Language Technologies Institute
5000 Forbes av.
Pittsburgh, PA 15213
stephan.vogel@cs.cmu.edu

Abstract

In a data-driven machine translation system, the lexicon is a core component. Sometimes it is used directly in translation, and sometimes in building other resources, such as a phrase table. But up to now little attention has been paid to how the information contained in these resources can also used backwards to help build or improve the lexicon. The system we propose here alternates lexicon building and phrasal alignment. Evaluation on Arabic to English translation showed a statistically significant 1.5 BLEU point improvement.

1 Introduction

In data-driven machine translation paradigms such as Statistical Machine Translation (SMT) and Example-Based Machine Translation (EBMT), the lexicon is an essential component since the systems look up translation candidates from the lexicon either as the primary or as the secondary resource.

In word-based SMT(Brown et al., 1993b), when an input sentence is given, the system looks up,in the lexicon, candidate translations for each token in the input sentence and then uses fertility and distortion information to to determine the number of translations and their proper placement in a hypothesis sentence. And even in an advanced system such as a phrase-based SMT (Koehn et al., 2003; Vogel et al., 2003) using a phrase table, the lexicon is still a core component which is looked up together with the phrase table.

In string-based EBMT (Nirenburg et al., 1994; Brown, 1996), when an input sentence is given, the system first retrieves the longest matches from the stored examples and then, in the lexicon, looks up for the words which don't have matches. In other EMBT systems (Sumita and Iida, 1991; Veale and Way, 1997), after the closest examples are found, the lexicon is used to find translations for the parts that differ between the retrieved source example and the input sentence.

In addition to its use in data-driven methods, a lexicon can also be used in different ways in other machine translation systems. For example, the Context-Based Machine Translation system (Carbonell et al., 2006) uses a hand-made lexicon to produce a lattice given an input sentence. Later it uses a large monolingual corpus in the target language to select and place translation tokens properly. Because of the great cost of a hand-built lexicon, it can also be replaced by a statistically generated one.

The prevalence of the lexicon in the various machine translation systems above indicates that any improvement in lexicon quality has the potential to make a significant contribution in the field.

1.1 Motivation

Word alignment has been a core part in lexicon building while phrasal alignment has been used in phrase table building. Phrasal aligners exploit a lexicon built based on word alignment and output phrase pairs based on lexical scores. But there has been little research investigating in what way available phrase pairs may be used to improve lexicon building.

The following two observations may clarify the motivation behind this approach. First, word alignment is more accurate for short sentences. Longer sentences have word duplicates and complex structures which have been obstacles in word alignment. Second, additional bilingual information can help improve alignment if it is of sufficient quality.

A phrase table satisfies both observations provided that it has high quality phrase pairs. It has shorter n-gram pairs which are extracted using lexical scores from a lexicon and additional statistical clues.

In this paper, we assess a new method which instantiates the above two ideas. The main idea in this method is to boost both algorithms by using alignment output from the other iteratively. In other words, we feed a word aligner a phrase table built by a phrasal aligner and this word aligner updates the lexicon which will be then fed to the phrase aligner to generate a better phrase table. We repeat these two steps until we don't observe any more benefit.

1.2 Previous Work

There have been many studies on lexicon building. Some researchers have studied non-probabilistic methods which use similarity functions between a source word and a target word and then use a threshold to filter out less reliable pairs (Gale and Church, 1991; Wu and Xia, 1994). Others have studied probabilistic methods such as IBM Models (Brown et al., 1993a) and HMM Model (Vogel et al., 1996) based on the word-to-word translation assumption.

On the other hand, SMT researchers noticed the limitation of the word-to-word assumption and developed phrasal alignment methods. Since word-to-word translation cannot convey local reordering and context, they tried to extract phrase pairs based on lexical scores using heuristics.

(Och and Ney, 2004) suggested an alignment template method that finds alignment templates by replacing words with their word classes. The word class information was automatically generated by a word clustering algorithm. (Chiang et al., 2005) extracted hierarchical structural alignment information from the word alignments and built grammar-like rules which are used in decoding in his HIERO system. (Koehn, 2004) extracted a phrase table from word alignment and used it in his phrasal decoder directly. While the above systems extract phrase pairs from word alignment information directly, PESA (Vogel, 2005) and SPA (Kim et al., 2005) extract target phrases given any n-gram source phrase on the fly. Both systems as the best target phrase which has the highest bi-directional translation score.

2 System Design

Our system was designed as illustrated in Figure 1. The system consists of a lexicon refining system and an evaluation system. The lexicon refining system consists of a *Lexicon Builder* and a *Phrasal Aligner* and the evaluation system consists of a *Decoder*.

- *Lexicon Builder*: This component first finds word-to-word alignments in both directions using IBM Model 1. It then combines them using a union operation at the sentence level and gathers word-to-word mapping statistics to finally build a lexicon. In the actual system, the Statistical Translation Tool Kit (STTK) (Vogel et al., 2003) was used as *Lexicon Builder*

- *Phrasal Aligner*: Using the lexicon built by the *Lexicon Builder*, this component extracts phrase pairs which will be given back to the *Lexicon Builder* either as a parallel corpus, or as concatenated to the original parallel corpus with an associated weight. PESA, which finds the most likely contiguous target phrase given a source phrase, was used as the *Phrasal Aligner*.

- *Decoder*: After updating the lexicon in each iteration, the *Decoder* is invoked to evaluate it. The *Decoder* does Minimum Error Rate (MER) training on a development set to get an optimized parameter set, which is then used for unseen data evaluation. MER training was done three times from three different starting points in an effort to avoid local optimum convergence. STTK was used as the *Decoder*.

The system starts with a training set. The *Lexicon Builder* builds a lexicon which will be given to the *Phrasal Aligner* to extract phrase pairs from the original training set. These phrase pairs are combined with the original training set and used in updating the lexicon using the *Lexicon Builder*. The resulting lexicon is then given to the *Phrasal Aligner*

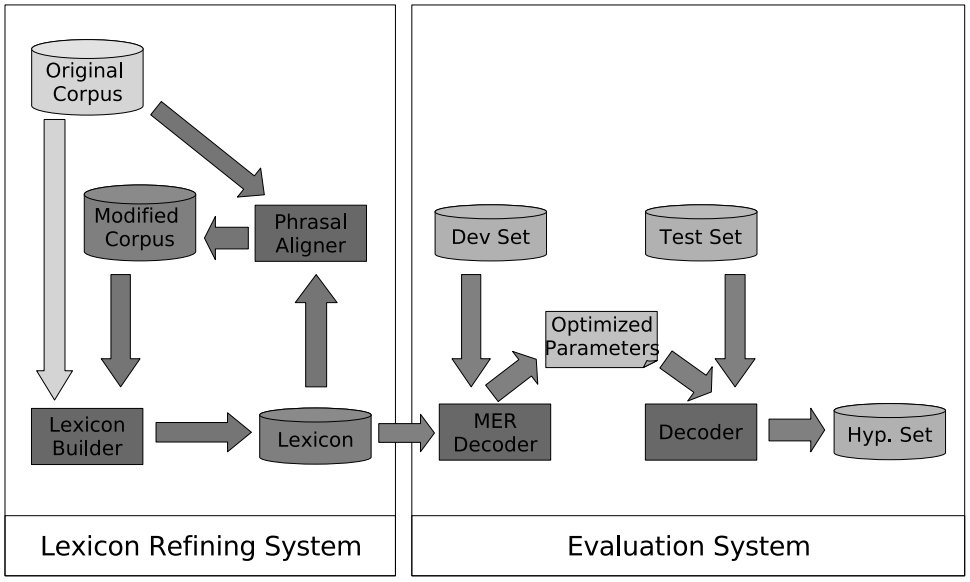

Figure 1: The system block diagram

again. Whenever the lexicon is updated, it is used using the *Decoder* to evaluate its quality. These steps are repeated until a given stopping criterion is satisfied. In our experiments, the system halts either when it meets a given number of iterations or when the development score indicates performance drops. The algorithm was described in the following:

1. corpus ← the original training set
2. build a lexicon
3. evaluate the lexicon (i.e., do translation using the lexicon)
4. if it satisfies the stopping criterion, halt
5. run the phrasal aligner to extract phrase pairs.
6. corpus ← a linear combination of the phrase pairs and the original corpus
7. goto 2.

3 Experimental Setup

In this project, we investigated three issues:

1. Do different maximum phrase lengths in phrase table extraction affect translation performance?
2. Do different combinations of the original training set and the extracted phrase table affect translation performance?
3. How can we find phrase pairs actually helpful for translation?

For the first issue, we trained and assessed the system with different maximum phrase lengths, in as far as this was possible given the relatively short sentences in the training set.

For the second issue, we used two kinds of ”combinations” of the phrase table and the original corpus: first, the phrase table by itself, and second, a weighted combinations of the phrase table with the original training set (see section 5 for more detail).

For the final issue, we extracted the best pairs for each source phrase and used them in lexicon building.

3.1 Data

For the initial experiments described in this paper, we used the Arabic/English data supplied with the 2006 International Workshop on Spoken Language Translation (International Workshop on Spoken Language Translation, 2006), which consists of short conversational sentences in the travel domain. The average source sentence length in our training set is about 7 and most of sentences are shorter than 10 words.

To study how the maximum phrase length affects system performance, we performed experiments with maximum phrase length 1 through 7 and

Set	# Sentences	# Tokens
Training - Source side	19847	137948
Training - Target side	19847	170014
Development	500	2159
Test	506	2060

Table 1: Data sets used

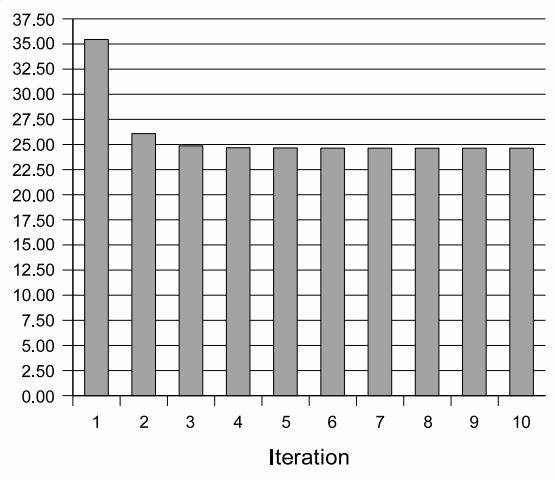

Figure 2: Log perplexity on the training set

10. The minimum phrase length was always set to 1 in all the cases.

For a development set, we used *devset2_IWSLT04* of 500 source sentences with 16 references. This set was used in parameter optimization in Minimum Error Rate(MER) training.

For an unseen test set, we used *devset3_IWSLT05* of 506 source sentences with 16 references.

3.2 Evaluation Metric

For evaluation, we used BLEU(Papineni et al., 2001) which is widely used in machine translation evaluation.

4 Results

4.1 Convergence

In figure 2, the training set log perplexity converges fast and there is no significant change after the third iteration. Because of this, we limited the number of iterations to three for all the following experiments.

Iteration	Best-DEV	TEST
1	0.4637	0.4324

Table 2: Baseline

4.2 The effects of different maximum phrase lengths

In table 3, *Phrase Table Only* and *Phrase Table + Original Corpus* show the system performance with different maximum phrase lengths when we use only the phrase table and a combination of the phrase table and the original training set as an input to the lexicon builder respectively.

For each maximum phrase length, the score at each iteration was measured on a lexicon built on a corpus generated in the previous iteration. So, the score at the first iteration was measured on the lexicon built on the original corpus, and the score at the second iteration was measured on the lexicon which was built on a phrase table built at the first iteration, and so forth. For this reason the score at the first iteration score is the same as the baseline score given in table 2, and does not depend on the maximum phrase length or the *Lexicon Builder* input. Note that this score does not change either as the maximum phrase length changes or as the input for *Lexicon Builder* changes.

The number of phrase pairs added at each iteration was reported later in table 5

Best-DEV denotes the best of three MER training scores on the development set at each iteration and *TEST* means the test set score. Due to many local optima, MER optimization converges to a local optimum depending on its initial configuration. So, we ran MER three times with different start configurations and take the one which gives the best result on the development set. The system parameter set which gives the best MER score was then used in *TEST* evaluation. At each maximum phrase length, the best *Best-DEV* and corresponding *TEST* are written in boldface font. These emphasized *TEST* scores are plotted in figure 3. In this figure, *pt_only* is for *Phrase Table Only* and *pt+org* is for *Phrase Table + Original Corpus*.

In the case of *Phrase Table Only*, we see that performance is below baseline for small maximum phrase lengths (1 or 2), but exceeds it for larger val-

		Phrase Table Only		Phrase Table Only + Original Corpus	
Phrase Length	Iteration	Best-DEV	TEST	Best-DEV	TEST
1	2	0.4509	0.4178	0.4552	0.4431
	3	**0.4588**	**0.4286**	**0.4584**	**0.4385**
2	2	0.4592	0.4313	**0.4650**	**0.4305**
	3	**0.4611**	**0.4224**	0.4596	0.4301
3	2	**0.4664**	**0.4318**	0.4673	0.4287
	3	0.4628	0.4381	**0.4678**	**0.4369**
4	2	**0.4720**	**0.4428**	**0.4744**	**0.4438**
	3	0.4637	0.4446	0.4691	0.4467
5	2	0.4731	0.4441	**0.4697**	**0.4405**
	3	**0.4739**	**0.4491**	0.4691	0.4484
6	2	0.4707	0.4456	0.4671	0.4395
	3	**0.4729**	**0.4388**	**0.4716**	**0.4395**
7	2	0.4705	0.4462	**0.4732**	**0.4445**
	3	**0.4740**	**0.4451**	0.4721	0.4430
10	2	0.4739	0.4428	0.4758	0.4379
	3	**0.4758**	**0.4431**	**0.4768**	**0.4443**

Table 3: Comparison of two different inputs for lexicon builder

ues (4 or more). This improvement, on both *Best-DEV* and *TEST*, is significant, as attested by significance testing using bootstrapping for NIST/BLEU confidence intervals(Zhang and Vogel, 2004).

The reason why we had score drops at maximum phrase length 1 and 2 is discussed in section 5. Overall, we see score improvement on both *Best-DEV* and *TEST* and the test set improvement is more than 1.5 BLEU points.

In the case of *Phrase Table + Original Corpus*, we see improvement when the maximum phrase length is 1. This time, we use the original corpus together with the phrase table and this mitigates the effect of errors in the phrase table. But we also see performance degradation when the maximum phrase length is 2 and this is also discussed in section 5.

We have a slightly better score than the baseline when the maximum phrase length is 3, and improvement on *TEST* with the maximum phrase length 4 or higher. We see score improvement on both *Best-DEV* and *TEST* and the latter exceeding 1.2 BLEU points.

In both cases, there is a certain amount of noise in the scores, to be attributed to the variations resulting from MER training. Even so, the overall trend is clear and significant.

n	Iteration	Best-DEV	TEST
0	1 (baseline)	0.4581	0.4278
1	2	0.4590	0.4368
2	3	0.4594	0.4278
3	4	0.4614	0.4404
4	5	0.4671	0.4391
5	6	0.4619	0.4343

Table 4: Top *n* alternatives for each source phrase

The comparison between *Phrase Table Only* and *Phrase Table + Original Corpus*, on the other hand, shows no statistically significant difference, with potential exception of the case of a phrase table length of 1, which is of limited relevance.

4.3 Phrase table filtering

To investigate which part of the phrase table is helpful, we slightly modified the experimental setup. We fixed the maximum phrase length to 5 and used only the best *n* phrase pairs for each source phrase instead

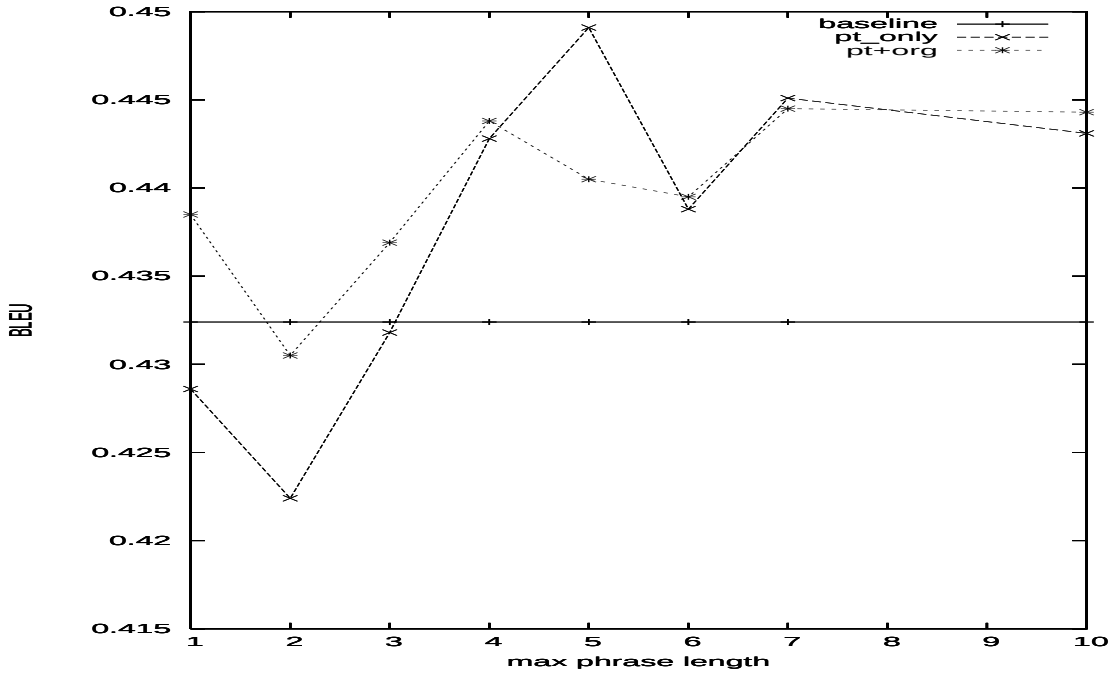

Figure 3: Test set score comparison

of using the whole phrase table. We started with n=0 and increased it by 1 at each iteration. So, in table 4, scores at the nth iteration were achieved using a lexicon built on a combination of the original training set and a filtered phrase table with *n-1*. Thus the base line is when n=0, and we see a peak of *Best-DEV* at the 5th iteration which uses a lexicon on the original training set and a phrase table filtered with n=4. In this experiment, we observed statistically significant increase of translation quality by more than 1 BLEU point.

Please note that the baseline here is different from that in table 2. Because MER training takes a lot of time, we had tighter beam in this experiment and got lower scores.

4.4 Lexicon building time

We also tracked the time required for lexicon building as this will be of great importance for more realistic amount of data. Table 5 shows the number of phrase pairs and the time elapsed in lexicon building with different maximum phrase lengths. *Phrase Table + Original Corpus* takes longer than *Phrase Table Only* because it includes the original training set. In both combinations, maximum phrase length 4 took about twice as long to train on as the original training corpus, which is the shortest maximum phrase length that gives improvement in both cases. We also see the time difference becomes smaller as the maximum phrase length grows because the sentence length governs the time complexity.

Max Phrase Length	# Phrase pairs	Phrase Table Only(sec.)	Phrase Table + Original(sec.)
0	0	-	34
1	157795	11	39
2	295743	33	50
3	413844	48	67
4	512216	70	86
6	654432	111	125
7	702762	132	145

Table 5: Lexicon building time

5 Discussion

From the results reported, we saw that iterative lexicon refinement helps translation system performance. However, we have not measured alignment improvement directly, and this may be a useful future exercise, but we can infer a corresponding im-

provement to the one in translation quality (which was, after all, our main objective).

In the experiments, iterative use of a word-to-word alignment and a phrasal alignment for mutual boosting showed translation improvement as measured by the BLEU metric. With sufficiently long maximum phrase lengths, we achieved more than 1.5 higher BLEU points in *Phrase Table Only* and more than 1.2 BLEU points in *Phrase Table + Original Corpus*. The score drop we observed for small maximum phrase length values can be explained as follows. First, sometimes one word aligns to several words in the other language, and these may be missing if the phrases in the other language are restricted too much. Second, since the phrasal aligner finds contiguous target phrases, they may include erroneous words that have more negative effects when phrases are shorter. It would be interesting to measure this effect by means of a proper phrasal alignment metric. Third, a lexicon is less 'smooth' when summing over fewer target words and this also means that alignment errors have a sharper effect.

In our experiments, with both kinds of combination, we had improvement from the maximum phrase length 4. This means that the target phrases of source phrases of length 4 or higher don't have significant amount of noise inside or relevant target words outside. This means the structural difference between two languages is local. If we have a very structurally different language pair, we have to have higher minimum maximum phrase length value. For instance, the structural difference between Korean and English is larger than that of Arabic and English, we think we will see improvement with a maximum phrase length higher than 4.

The phrase table filtering experiment showed that *Best-DEV* monotonically increased until the fifth iteration and *TEST* also shows improvement up to that point. This indicates that we may be able to shrink the phrase table conveniently by excluding less helpful pairs. Another possibility would be to specify the retained phrase as a relative value (percentage of pairs in the phrase table) instead of an absolute number.

With regard to the way of combining the phrase table and the original training set, one can think of assigning different weights to the different training data:

$$NewInput \leftarrow \lambda \times PhraseTable + (1-\lambda) \times OriginalCorpus \quad (1)$$

(In this point of view, the combinations we did were when $\lambda \leftarrow 1$ and $\lambda \leftarrow 0.5$.) Or one can think of combining the lexicon built on only the original training set and the current lexicon:

$$UpdatedScore \leftarrow \lambda \times CurrentScore + (1-\lambda) \times OrigScore \quad (2)$$

But since we observed no significant difference between *Phrase Table Only* and *Phrase Table+Original Corpus*, assessing these methods may not be quite meaningful.

6 Conclusion

The experimental results in this paper show that iterative refinement of lexicon and phrasal alignment builds a better lexicon in terms of translation quality: when we used a phrase table with the maximum phrase length 4 or higher, we observed a statistically significant increase of translation quality by 1.5 BLEU points. The different combinations of original corpus and phrase table data we explored, however, yielded no statistically significant benefit.

We also discussed ways of selecting 'convincing' phrase pairs from a phrase table. Besides reducing the amount of phrase table data and the training time, this did in fact lead to a significantly higher translation score.

Acknowledgments

We thank Sanjika Hewavitharana for helping us set up experiments and run STTK and PESA toolkits. We also thank Peter Jansen for giving us many helpful comments.

References

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1993a. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2).

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993b. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Ralf D. Brown. 1996. Example-Based Machine Translation in the PANGLOSS System. In *Proceedings of the Sixteenth International Conference on Computational Linguistics*, pages 169–174, Copenhagen, Denmark. `http://www.cs.cmu.edu/~ralf/papers.html`.

Jaime Carbonell, Steve Klein, David Miller, Michael Steinbaum, Tomer Grassiany, and Jochen Frey. 2006. Context-Based Machine Translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 19–28. The Association for Machine Translation in the Americas.

David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. 2005. The Hiero Machine Translation System: Extensions, Evaluation, and Analysis. In *HLT/EMNLP*.

W. A. Gale and K. W. Church. 1991. Identifying Word Correspondences in Parallel Texts. In *Proc. of the Speech and Natural Language Workshop*, page 152, Pacific Grove, CA.

International Workshop on Spoken Language Translation. 2006. International workshop on spoken language translation. Kyoto, Japan. http://www.slc.atr.jp/IWSLT2006/.

Jae Dong Kim, Ralf D. Brown, Peter J. Jansen, and Jaime G. Carbonell. 2005. Symmetric Probabilistic Alignment for Example-Based Translation. In *Proceedings of the Tenth Workshop of the European Assocation for Machine Translation (EAMT-05)*, pages 153–159, May.

Philipp Koehn, Franz Joseph Och, , and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada, May 27-June 1.

Philipp Koehn. 2004. Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *AMTA*, pages 115–124.

Sergei Nirenburg, Stephen Beale, and Constantine Domashnev. 1994. A Full-Text Experiment in Example-Based Machine Translation. In *New Methods in Language Processing*, Manchester, England.

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

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation.

E. Sumita and H. Iida. 1991. Experiments and Prospects of Example-Based Machine Translation. In *ACL '91*.

Tony Veale and Andy Way. 1997. Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation. In *Proceedings of the NeMNLP'97, New Methods in Natural Language Processessing*, Sofia, Bulgaria, September.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *COLING '96: The 16th International Conference on Computational Linguistics*, pages pp. 836–841. ACL'96.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU Statistical Translation System. In *Proceedings of MT Summit IX*, New Orleans, LA, September.

Stephan Vogel. 2005. PESA: Phrase Pair Extraction as Sentence Splitting. In *Proceedings of MT Summit X*, Phuket, Thailand, September.

D. Wu and X. Xia. 1994. Learning an English-Chinese Lexicon from a Parallel Corpus. In *AMTA-94, Association for Machine Translation in the Americas*, pages 206–213, Columbia, Maryland, October.

Ying Zhang and Stephan Vogel. 2004. Measuring Confidence Intervals for the Machine Translation Evaluation Metrics. In *Proceedings of The 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, October.