Reducing Boundary Friction Using Translation-Fragment Overlap

Ralf D. Brown[2]
ralf+@cs.cmu.edu

Rebecca Hutchinson[1]
rah+@cs.cmu.edu

Paul N. Bennett[1]
pbennett+@cs.cmu.edu

Jaime G. Carbonell[2]
jgc+@cs.cmu.edu

Peter Jansen[2]
pjj+@cs.cmu.edu

[1]Computer Science Department and [2]Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Many corpus-based Machine Translation (MT) systems generate a number of partial translations which are then pieced together rather than immediately producing one overall translation. While this makes them more robust to ill-formed input, they are subject to disfluencies at phrasal translation boundaries even for well-formed input. We address this "boundary friction" problem by introducing a method that exploits overlapping phrasal translations and the increased confidence in translation accuracy they imply. We specify an efficient algorithm for producing translations using overlap. Finally, our empirical analysis indicates that this approach produces higher quality translations than the standard method of combining non-overlapping fragments generated by our Example-Based MT (EBMT) system in a peak-to-peak comparison.

1 Introduction

Corpus-Based Machine Translation approaches, including Statistical MT (SMT) (Brown et al., 1990; Brown et al., 1993; Yamada and Knight, 2002), and Example-Based MT (EBMT) (Nagao, 1984; Nirenburg et al., 1994; Sumita and Iida, 1991; Veale and Way, 1997; Brown, 2001) use a sentence-aligned bilingual corpus to train translation models. The former relies on word and n-gram statistics to seek the most probable translation, and the latter relies on finding translated maximal-length phrases that combine to form a translation. Each method has its strengths and weaknesses: EBMT can exploit long translated phrases, but does not combine phrasal translations well, whereas SMT combines word and short n-gram translations well, but cannot readily exploit long pre-translated phrases. This paper addresses in part the major shortcoming of EBMT: how to better combine phrasal translations. When standard corpus-based approaches find several long n-grams with known translations in the sentence being translated, they can only exploit these if the fragments are non-overlapping. Similarly, multi-engine MT (MEMT) systems may be forced to select a poorer translation for one portion of the input after selecting the best translation for another portion.

We have developed a method of combining overlapping fragments so that if they also have consistent overlapping translations, they compose a legitimate translation more likely to be accurate than sequentially-abutting translated fragments. We call this method "maximal left-overlap compositional MT", or for short "maximal overlap MT." Although we had previously experimented with one-word overlap at EBMT fragment boundaries, the n-word overlap version is clearly more powerful, particularly in conjunction with multiple translation engines.

This paper is organized as follows. First, we give a presentation and illustration of the maximal overlap MT method. Then we describe the lattice search method and how to incorporate overlap into the search. Finally, we present clear results demonstrating the power of Overlap MT on the Hansard Corpus.

Fragment	Source	Target
Input:	Je doute qu'il soit nécessaire de lancer une enquête complète pour l'instant.	
1	Je doute qu'il	I do not think it is
2	Je doute qu'il soit	I doubt whether that will be
3	qu'il soit nécessaire de	not think it is necessary to
4	nécessaire de lancer	necessary to start
5	une enquête complète	a full investigation
6	pour l'instant.	for the moment.

Human reference translation:
"I do not think it is necessary to launch a full inquiry at this time."

Standard MEMT selection combines fragments 2, 4, 5, and 6, to produce the output:
"I doubt whether *that* will be necessary to start a full investigation for the moment."

MEMT selection with overlap combines fragments 1, 3, 4, 5, and 6, to produce the output:
"I do not think it is necessary to start a full investigation for the moment."

Figure 1: A Portion of an MEMT Translation Lattice. Fragments 1 and 3 overlap while 2 and 3 do not (since they do not match in the target language). The standard method cannot combine Fragments 2 and 3 because of their source overlap, and thus outputs a syntactically coherent but incorrect translation (as a result of the use of "that" which implies a referent), whereas the method using overlap is both syntactically and semantically correct. The full translation lattice for this example has approximately 60 fragments.

2 Maximal Overlap Method

When the EBMT engine is given a sentence for translation, it outputs a list of source fragments (contiguous portions) from the input sentence and candidate translations obtained from its word-level alignment of the example translations it was originally given. Each source/target fragment pair has its own quality score, and we refer to a pair as simply a fragment below (specifying source or target as necessary). A general method that considers overlap must balance the fragment scores obtained from the translation engine(s) with the amount of overlap between these fragments as well as other possible factors (*e.g.*, fragment length).

Figure 1 shows an excerpt from an MEMT translation lattice into which the fragments that the EBMT engine retrieved from the parallel training corpus it was given have been placed. Translation proceeds by finding a path through the translation lattice that combines the fragments. Traditionally, such combination procedures have required the source fragments to have no overlap. Our method stems from the motivation that when both the source and target of two adjacent fragments overlap, then there is an increased likelihood their combination is an accurate translation.[1]

In the example in Figure 1, the standard combination procedure yields a syntactically coherent but semantically incorrect translation. The result is a sentence where the use of "that" implies a referent, and thus, the statement is interpreted as, "A specific condition is not required to start a full investigation." The combination procedure that uses overlap produces a translation with the correct semantics, "It is the speaker's opinion that a full investigation is not necessary." This is a direct result of consider-

[1] Sometimes there is no opportunity to exploit overlap when translating a sentence, because the full sentence and its translation occur verbatim in the training corpus, or because some portion of the input produces no translations at all, leaving an unbridgeable gap.

ing overlap in the fragments. The reason is that the "il" in the context of "qu'il...nécessaire de" should never be translated as a word with a referent. Thus, a training set with correct translations will never contain a fragment such as Fragment 2 that extends all the way to "de". However, when overlapping fragments are used, an example of the initial portion of the phrase (Fragment 1) and an example continuing with the idiomatic "qu'il soit" (Fragment 3) can be combined to produce an accurate translation. In general, both syntactic and semantic problems can occur at the boundaries of fragments when overlap is not considered.

3 Incorporating Overlap

3.1 The EBMT Engine

The EBMT system that we used for our experiments was intended from the very beginning (Brown, 1996) to act as one engine in a multi-engine machine translation (MEMT) system (Frederking et al., 1994). As a result, it differs in a number of aspects from most implementations of EBMT. For our purposes, the important difference is that the engine itself need not find a single best overall translation because its output is intended to be fed into a separate selection step. Instead, the EBMT engine outputs translations of all the phrasal matches it finds in the training corpus and is able to align at the word level within the example containing the phrase. These partial translations may be ambiguous and can overlap (either partially or subsuming some shorter translations), as was illustrated in Figure 1.

The EBMT engine assigns each candidate translation a quality score, which is computed as a linear combination of an alignment score and the translation probability for that candidate. The alignment score is based on a weighted set of heuristics and indicates the engine's confidence that it has selected the proper target-language phrase corresponding to the source-language phrase which was found in the training corpus. The translation probability is simply the proportion of times each distinct alternative translation was encountered out of all successful alignments for a particular source-language phrase.

3.2 The Multi-Engine Machine Translation Architecture

Our multi-engine MT system (Brown and Frederking, 1995) applies several differing translation techniques in parallel, and then selects the best overall translation from among the partial translations (fragments) generated by the various engines. Each engine is permitted to segment the input text in whichever way is appropriate for it, e.g. a dictionary lookup would generate a separate fragment for each source word, the EBMT engine generates one fragment for each distinct source phrase found in the corpus, and a knowledge-based system might generate a fragment for each semantic unit.

All translation candidates are placed into a common lattice, from which a best overall path is selected. Each fragment is weighted by the quality score assigned by the engine which generated it, an overall weight assigned to that engine, and bonus and penalty factors – bonuses are given for longer fragments, penalties for length mismatches between source and target halves, untranslated words, etc. A multi-level beam search guided by the fragment weights and a target-language trigram language model (with smoothing and back-off where necessary) is then used to select the optimal set of fragments that produces a complete translation. Prior to the work described in this paper, this optimal set was restricted to non-overlapping fragments.

We have, in the past, used varying combinations of word-for-word dictionaries, phrasal glossaries, EBMT, SMT, knowledge-based MT, and transfer-rule engines in the multi-engine architecture. For the purposes of the experiments described in this paper, we used only the EBMT engine. This simplifies the process of tuning the system for the various experimental conditions and reduces any confounding factors.

3.3 Modifying the Search Procedure

The search procedure in our MEMT system is a multi-level beam search. A separate priority queue is maintained for each word position in the input text, and they are processed in order from left to right. As each search node is removed from the active queue, it is expanded by adding on each fragment that could possibly extend it, and the new nodes are added to the priority queues corresponding to the last input word covered by each extended path. The priority queues are pruned on each addition by removing the lower-scoring of duplicate nodes (those with the same last two target-language words, since earlier words can no longer affect scores) or the lowest-scoring node if the addition would cause the length of the queue to exceed the specified beam width.

Updating the search procedure to handle overlapping fragments required only two changes: a new definition of "fragment that can extend the current partial path" and a means of giving a bonus to the path's score when an overlapping fragment is added.

The original set of fragments which could extend a path was simply those which started with the word immediately following the last one covered by the path. In the event that no engine generated a candidate covering a particular word, a dummy arc with a large penalty is inserted before the search begins, so that there is always a possible extension available. To permit overlapping fragments, we defined the notions of "source-language overlap" and "target-language overlap", and the criteria for allowable ranges for those two values.

The source-language overlap is simply the number of words in common between the two fragments, based on their starting and ending word numbers in the input. For example, if the current path ends with word number 6, then a fragment starting on word 6 would have a one-word source overlap, one starting on word 5 would have a two-word source overlap, etc. This is trivial to compute since the start and end points are provided by the translation engine(s) to identify where the candidate is to be placed in the lattice.

For target-language overlap, we have chosen the simplest possible definition likely to produce acceptable results: the number of words in the suffix of the left-hand fragment that exactly matches a prefix of the right-hand fragment. Should there be multiple possible values, the length closest to the source-language overlap is chosen. The target-language overlap can be ambiguous in cases where a word is repeated, and selecting the phrase which is most similar in length to the common section on the source-language half will yield the phrase most likely to be the common section's translation. For example, if we have fragments with translations

was	the	best	of	the	
	the	best	of	the	rest

the alignment shown would be appropriate for a three- or four-word source overlap, but rarely for a single-word source overlap. In the latter case, we probably want

was	the	best	of	the			
				the	best	of	the	rest

The criteria for allowing a fragment to be added to a path through the lattice are now:

- the source-language overlap is no more than $max\_source\_overlap$;
- the target-language overlap is within a certain range of the source-language overlap; and
- the fragment extends the path by at least one word on the source-language side.

Setting $max\_source\_overlap$ to zero produces the previous behavior of allowing only non-overlapping arcs in the path. The limit on target-language overlap may be specified as either an absolute difference in length between source and target overlaps, or as the minimum ratio between the shorter and the longer of the two values (i.e. a value of 0.5 means that a source-overlap of 2 words yields allowable target overlaps between one word [$1/2 \geq 0.5$] and four words [$2/4 \geq 0.5$]).

Our means for giving a bonus for using overlapping fragments is to boost the score assigned to the words in the overlap region. The overall score for a path through the translation lattice is the arithmetic average of the scores for each target word on that path (this avoids systematic biases toward shorter or longer outputs). In the absence of overlap, the score for an individual target-language word is a combination of the weight of the arc containing that word and the language-model score for the trigram ending on that word. When overlap occurs, the individual scores for each word in the overlap region are increased by some multiple weighting W of their existing weights. We have found that values of W between 3 and 6 generally perform well, though for some training corpora substantially larger values can produce a slight increase in evaluation metrics over values in this range. This method automatically gives preference to larger overlaps by boosting the scores of more words when the overlap region is larger.

The overall score for a path is thus

$$\frac{1}{n}\sum_{i=1}^{n}(1+O_i) \times e_i b_i p_i s_i \times P(w_i|w_{i-2}w_{i-1}),$$

$$O_i = W\ if\ overlap\ used\ for\ w_i,\ 0\ otherwise$$

where n is the number of target-language words in the path, e_i is the sum of the engine weights for the translation engines producing the fragment containing word i, $b_i \geq 1$ is the bonus factors for the fragment, $p_i \leq 1$ is the penalty factors for the fragment, and s_i is the engine-assigned quality score for the fragment, and $P(w_i|w_{i-2}w_{i-1})$ is the language-model probability. To maintain consistency, w_{-1} and w_0 are the sentence-end and sentence-start context cues, respectively.

The lattice search is quite efficient, typically taking only a few milliseconds per sentence, though some long and highly ambiguous sentences can take considerably longer. Adding overlapping fragments to the search roughly doubles the search time (e.g. from 2.4 to 3.9 seconds for the 1000 sentences in our test sets), primarily as a result of increasing the search space. However, since this represents only a small portion of the total run time (470 seconds), the overall impact on translation speed is negligible.

4 Experiments

4.1 Data Set

All of the data used for the primary experiments described below came from the Hansard corpus, which consists of parallel French and English versions of Canadian parliamentary debates and related documents. In all experiments using the Hansard corpus, the source language was French and the target language was English. The corpus as available from the Linguistic Data Consortium consists of 273 files of 10,000 sentences for each language. We constructed a training set consisting of all sentence pairs from files 090 through 099, for a total of 100,000 sentence pairs. The two validation sets used to optimize the parameters for the EBMT engine consisted of the first one hundred sentences in files 020 and 040, respectively. The test data consisted of ten different mutually-exclusive one-hundred sentence segments drawn from files 060 and 080. (Some of our early experiments used a larger training set consisting of files 000 through 099, omitting files 020, 040, 060, and 080 for use as validation and test data.)

4.2 Machine Translation Quality Scoring Function

For empirical evaluation, we use the metric proposed by IBM, called *BLEU* (Papineni et al., 2002) and the metric developed by NIST based on BLEU, called simply *NIST* below ((NIST), 2002). Both metrics try to assess how close a machine translation is to a set of reference translations generated by humans. Our experiments use the single reference translation provided by the Hansard transcripts.

The BLEU and NIST metrics are based on n-gram co-occurrences between the system output and the reference translations, for each size of n-gram from unigram to 4-gram (BLEU) or 5-gram (NIST). BLEU computes the geometric mean of the n-gram precisions and then applies a brevity penalty for system outputs which are shorter than the reference translations. NIST computes the arithmetic mean of the information-weighted n-gram co-occurrences and then applies a brevity penalty which is less sensitive to small variations but more extreme than BLEU's for very brief outputs. The *i*nformation weight places greater emphasis on n-grams for which the last word is "suprising", i.e. unlikely to be predicted by the preceding words of the n-gram based on the totality of the text in the reference translations.

Both BLEU and NIST scores are sensitive to the number of reference translations, yielding higher scores with more reference translations. Although both are also sensitive to the number of words in the reference corpus, NIST is much more so because of the language model implied by the information weights, which are often zero for large n in small corpora.

Note that BLEU and NIST scores in DARPA evaluations are against multiple reference translations for each test sentence. We only have one reference translation per sentence, and hence *scores are uniformly lower than in recent DAPRA evaluations by as much as a factor of 2 solely as a result of this scoring artifact.*

4.3 Experimental Conditions

We compared four variations of the multi-engine combination, each evaluated at its peak performance levels according to the BLEU and NIST scores, for a total of eight experimental conditions. The four variations, and the names by which they will be identified are

- "noOV-noLM": disallow overlapping fragments and use uniform trigram probability (i.e. ignore language-model information)
- "OV-noLM": allow overlapping fragments to combine (where none are present, continue to use non-overlapping fragments) and use uniform trigram probabilities
- "noOV-LM": disallow overlapping fragments, but make use of a trigram language model of the target language; for our experiments, the language model was built from the Xinhua portion of the LDC GigaWord English corpus (Consortium, 2003), approximately 150 million words.
- "OV-LM": use both overlapping fragments (where available) and a language model.

The `noOV-LM` case was the normal mode of operation for our multi-engine system prior to the work described in this paper.

4.4 Parameter Optimization

To ensure a peak-to-peak comparison of the highest possible performance for all four systems we empirically optimized key parameters for each variation.

The EBMT engine takes four parameters to define its search space. We tuned these four parameters by running the system on each validation set using the cross-product of several values for each parameter, then (as appropriate) re-running with more closely-spaced values for the parameters. Typical values for the parameters are $max\_duplicates$=500-1000 (number of occurrences of a phrase to examine), max_align=50-100 (number of occurrences to align and use for computing translation probabilities), $\max\_alternatives$=1-3 (number of alternative translations to generate), and $thresh$=0.05-0.15 (required word-alignment score). While the peak of the performace space is relatively flat (one can change the parameters substantially away from the exact optimum without losing more than 1-2% on the metrics), it is somewhat irregular. Therefore, after finding the optimal settings for the validation sets drawn from Hansard files 020 and 040, we then selected in-between values for the test runs.

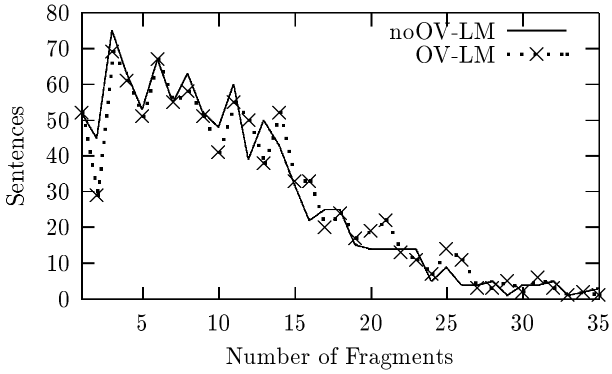

Figure 2: Number of fragments combined to form the output translation for each sentence.

4.5 Experimental Results

In Table 1, we show results for each of the experimental conditions described above. The scores in Table 1 are evaluated using only one reference translation, which yields lower scores than the multiple reference translations used in DARPA evaluations. The highest performing system is highlighted in the table for both BLEU and NIST. We also report two types of significance tests to compare the systems using overlap against those not using it. The first is a sign test, performed on the set of individual sentence scores of the test set. The null hypothesis is that each of the two systems being compared translates a given sentence better about half the number of times that they receive different scores on a sentence. The second test is a two-sided t-test on the difference between each pair of scores over the ten files that comprised the test set. The null hypothesis is that the difference is zero.

5 Discussion

We conclude from the results presented above that the `OV-LM` system is superior to the standard `noOV-LM` system. For the Hansard training corpus, adding overlap to language modeling improves the

Training Set	System	Mean BLEU	St. Dev. BLEU	Mean NIST	St. Dev. NIST
Hansard 100k pairs	noOV-noLM	0.1412	0.0248	4.3017	0.3914
	noOV-LM	*0.1549**	0.0302	4.3591	0.3771
	OV-noLM	*0.1553**	0.0293	***4.4433****	0.4121
	OV-LM	***0.1707****+	0.0359	***4.4578****+	0.3942

Table 1: A performance summary of the described methods. Starred and plus-marked results are highly significant against the `noOV-noLM` and `noOV-LM` systems, respectively, according to the two-sided t-test ($p \leq 0.002$). Italicized and boldface results are significant against `noOV-noLM` and `noOV-LM` according to the sign test ($p \leq 0.05$).

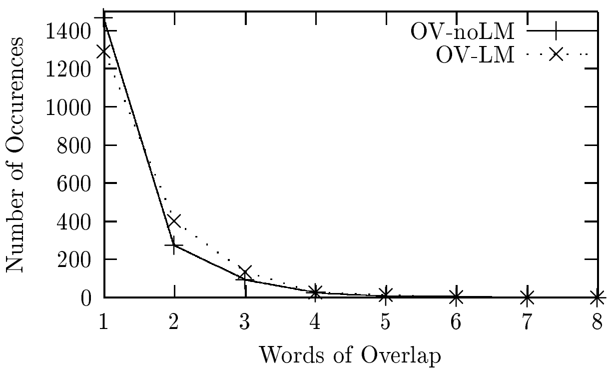

Figure 3: Target-language words of overlap between each selected fragment and its predecessor.

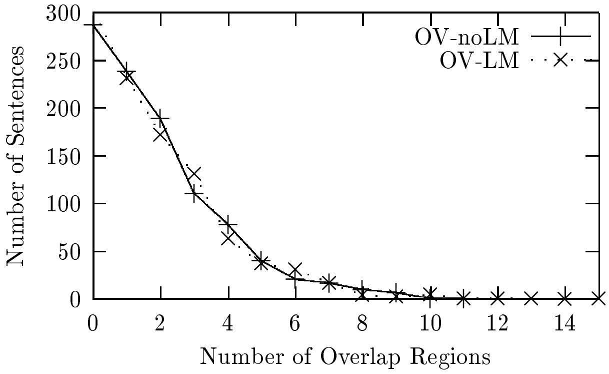

Figure 4: Number of overlapping regions per sentence.

BLEU score by 10.2% and the NIST score by 2.3%. Both the t-test and sign test rate these differences as highly significant, with $p \leq 0.001$.

Figure 2 shows that on average, the `OV-LM` system uses more fragments per sentence than the `noOV-LM` system. This is as expected, since an overlapping fragment of a given size extends the path through the lattice less than a non-overlapping fragment. Figures 3 and 4 show that both the number of overlap regions in a sentence and the size of those overlaps skews higher when a language model is added. This may be a consequence of the slightly higher average fragment length when using a language model.

713 of the 1000 sentences in the test files made use of overlap in the `OV-noLM` system, 698 in the `OV-LM` system. Those two systems had 1870 of 9546 and 1870 of 9309, respectively, non-initial fragments that were selected for the final translation overlapping their left neighbors. It is interesting that overlap provides as much of a performance boost as a language model trained on substantially more text than is contained in the EBMT corpus despite being inapplicable in three out of every ten sentences and only being used at about 20% of fragment boundaries. An improvement to allow more overlaps should thus improve performance even further.

6 Conclusions and Future Work

In summary, we have presented a method of combining partial translations (whether from a single translation engine or multiple engines) that exploits the reinforcement inherent in overlapping translated phrases. Our overlap method produces a statistically significant improvement in translation quality over a system in the traditional non-overlapping paradigm. While we have not performed detailed analyses or peak-to-peak comparisons in other cases, we consistently – across multiple test sets in multiple languages – achieve a 1-4% increase in NIST score and

3-6% increase in BLEU score when enabling overlap in our multi-engine system running both a word-for-word dictionary engine and the EBMT engine. Our statistical MT group has implemented a version of the overlap method into their system's decoder, and reports similar increases in scores when enabling overlap in conjunction with phrasal transducers.

Overlap seems to be beneficial in two ways. The first is that it allows a system to use long phrasal translations that cannot be used by standard MEMT because they overlap with each other, while never preventing the use of any non-overlapping translations that are combinable by the standard system. Additionally, systems benefit when overlap occurs frequently enough to take advantage of consistent translations of shorter fragments.

We intend to pursue a number of extensions to this work. The first of these is to generalize the notion of target-language overlap to take into account word-order differences between source and target languages. The EBMT engine could provide its internal word-level alignments for the translation fragments that it outputs, in which case the overlap mechanism could select fragments where the matching words are not necessarily a suffix of the left-hand or prefix of the right-hand fragment. We also plan to investigate using overlap in conjunction with grammar rules.

References

Peter Brown, J. Cocke, S. DellaPietra, V. DellaPietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2).

Peter Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19:263–312.

Ralf Brown and Robert Frederking. 1995. Applying Statistical English Language Modeling to Symbolic Machine Translation. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, pages 221–239. `www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. 1996. Example-Based Machine Translation in the Pangloss System. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING-96)*, pages 169–174 (vol 1), Copenhagen.

Ralf D. Brown. 2001. Transfer-Rule Induction for Example-Based Translation. In *Proceedings of the Workshop on Example-Based Machine Translation*, September.

Linguistic Data Consortium. 2003. English Gigaword. Catalog number LDC2003T05.

Robert Frederking, Sergei Nirenburg, David Farwell, Steven Helmreich, Eduard Hovy, Kevin Knight, Stephen Beale, Constantin Domashnev, Donna Attardo, Dean Grannes, and Ralf Brown. 1994. Integrating Translations from Multiple Sources within the PANGLOSS Mark III Machine Translation. In *Proceedings of the first conference of the Association for Machine Translation in the Americas*, Columbia, Maryland.

Makoto Nagao. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In Alick Elithorn and Ranan Banerji, editors, *Artificial and Human Intelligence*, pages 173–180, Lyon, France, October, 1981. Elsevier Science Publishers B.V. Proceedings of the International NATO Symposium.

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

National Institute for Standards and Technology (NIST). 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. `http://www.nist.gov/speech/tests/-mt/doc/ngram-study.pdf`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 311–318.

E. Sumita and H. Iida. 1991. Experiments and Prospects of Example-Based Machine Translation. In *Proceedings of the 29th Annual Meeting of the Association of Computational Linguistics (ACL)*.

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.

K. Yamada and K. Knight. 2002. A Decoder for Syntax-Based Statistical MT. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*.