Maximal Lattice Overlap
in Example-Based Machine Translation

Rebecca Hutchinson Paul N. Bennett Jaime Carbonell
Peter Jansen Ralf Brown

June 6, 2003
CMU-CS-03-138

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Also appears as Language Technologies Institute Technical Report
CMU-LTI-03-174

Abstract

Example-Based Machine Translation (EBMT) retrieves pre-translated phrases from a sentence-aligned bilingual training corpus to translate new input sentences. EBMT uses long pre-translated phrases effectively but is subject to disfluencies at phrasal translation boundaries. We address this problem by introducing a novel 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 EBMT in a peak-to-peak comparison.

Email: rah+@cs.cmu.edu, pbennett+@cs.cmu.edu, jgc+@cs.cmu.edu, pjj+@cs.cmu.edu, ralf+@cs.cmu.edu

Work supported in part by the National Science Foundation under grant number IIS-9982226 (MLIAM: MUCHMORE: Multilingual Concept Hierarchies for Medical Information Organization and Retrieval) and award number IIS-9873009 (KDI: Universal Information Access: Translingual Retrieval, Summarization, Tracking, Detection and Validation).

Keywords: machine translation, example-based machine translation, boundary friction, mutual reinforcement

1 Introduction

Corpus-Based Machine Translation (MT), 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 exploit long pre-translated phrases. This paper addresses in part the major shortcoming of EBMT: how to better combine phrasal translations. When standard EBMT 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. We have developed a method of combining overlapping fragments whose translations are consistent. This method, which we call "maximal left-overlap compositional EBMT" or for short "maximal overlap EBMT", forms a translation more likely to be accurate than sequentially-abutting translated fragments. Although we had previously experimented with one-word overlap at EBMT fragment boundaries, the n-word overlap version is clearly more powerful.

This paper is organized as follows. First, we give a presentation and illustration of the maximal overlap EBMT method. Then we describe the scoring method for incorporating overlap into the EBMT lattice search and the new lattice search algorithm. Finally, we present results that clearly demonstrate the power of overlap EBMT on the Hansard Corpus.

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 alignment of the example translations it was originally given. Each source/target fragment pair has its own alignment score, and we refer to a pair as simply a fragment below (specifying source or target as necessary). A method that uses overlap must balance the fragment scores obtained from the EBMT engine with the amount of overlap between these fragments as well as other possible factors (*e.g.*, fragment length).

Table 1 shows an excerpt from an EBMT translation lattice. After the EBMT engine retrieves fragments from its parallel training corpus, translation proceeds by finding a path through the translation lattice that combines the fragments in some manner. 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 Table 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 considering 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 that begins like Fragment 2 and 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"

[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. One such amusing example is: "The government does not know what it is doing."

Input:	Je doute qu’il soit nécessaire de lancer une enquête complète pour l’instant.
Fragment	
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 EBMT translation 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.”
EBMT translation 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.”

Table 1: A Portion of an EBMT Translation Lattice. In order to combine fragments with overlap, they must match in both the source and target language. Thus, Fragments 1 and 3 can be combined while 2 and 3 cannot. The full translation lattice for this example has approximately 60 fragments.

(Fragment 3) can be combined to produce an accurate translation. In general, both syntactic and semantic problems can occur when overlap is not considered.

2.1 The EBMT Engine

Similar to (Frederking et al., 1994), the EBMT system that we used for our experiments was originally intended to act as one engine in a multi-engine machine translation (MEMT) system. Hence, 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 that it is able to align at the word level. These partial translations may be ambiguous and can overlap (either partially or subsuming some shorter translations). Having such a list of all candidate partial translations available as the engine’s output makes it straightforward to implement a new selection mechanism.

2.2 Scoring Function

The scoring function that guides fragment combination in our algorithm assigns an overall score to each candidate translation. This score is simply the sum of the scores for each constituent fragment. The score of a fragment depends only on that fragment and the immediately preceding fragment in the translation. By choosing an overall scoring function that is additive and dependent only on the fragment and its predecessor, we can conduct an efficient search for the best scoring translation candidate.

The scoring function weights four different attributes of a fragment: the gap between the source fragment and its predecessor, the length of the source fragment, the alignment score from the EBMT engine, and the overlap between a fragment and its predecessor. Four parameters, (g, l, a, o), weight the relative influence of each of these factors.

The score, $s(F)$, assigned to a fragment, F, is:

$$\begin{aligned} s(F) &= g * \mathit{SourceGapLength} * \mathit{GapPenalty} + s'(F) \\ \text{where} & \\ s'(F) &= 1/(a * \mathit{Align}(F) + o * \mathit{OverlapLength} + l * \mathit{SourceLength} + 1). \end{aligned}$$

Scores of $s(F)$ closer to zero are better. That is, we desire to minimize this function. We now describe each component of the scoring function.

The *SourceGapLength* is the number of untranslated words from the input sentence that fall between a fragment and its predecessor. The *GapPenalty* is defined dynamically in terms of s' as the s' score given to a perfectly aligned single word translation. We set $g{=}2$, so the final weight a gap in translation carries is twice as bad as replacing each word with a perfect dictionary look-up.

$\mathit{Align}(F)$ is the score obtained from the EBMT engine that indicates the engine's confidence that the source and target fragments are well-aligned.

OverlapLength is defined to be zero if the source of a fragment and its predecessor do not overlap in the input sentence. Therefore, a non-zero overlap implies that the gap length is zero. When there is source overlap between the end of a predecessor fragment and the beginning of the current fragment, then *OverlapLength* is the maximum number of target words from the end of the predecessor fragment that match the beginning of the current target fragment. In Table 1, Fragments 1 and 3 have an *OverlapLength* of four ("not think it is").

Finally, *SourceLength* is simply the length of the fragment in the source language. Each of the components, *Align*, *OverlapLength*, vary from zero to approximately *SourceLength*, with higher values being better. The coefficients l, a, and o were optimized empirically as described in Section 3.3.2.

2.3 Search Discipline

Since the scoring function is additive over fragments and dependent at each step only upon a fragment and its immediate predecessor, we can use a *dynamic programming* solution to avoid an exponential search problem. Dynamic programming approaches take advantage of problems where the best solution at a point A in a lattice can be defined in terms of the best solution at all points that can reach A in one step and the cost of that step.

Despite these efficiencies, the search is still computationally expensive since the EBMT engine can output hundreds of fragment pairs for a single sentence. Since the optimizations we have described so far result in an algorithm that is roughly $O(n^2)$ in the number of fragment pairs, this can sometimes be a computational burden.

Therefore, we further optimize our search by using a *best-first beam* search. A *best-first* search expands the best-scoring unfinished candidate translation first. A *beam* search only retains the top-n unfinished candidate translations. The top scoring finished candidate is stored and updated as better scoring translations are found. Once the beam is empty, the top scoring candidate is output as the best translation. Our experiments use a beam of twenty.

2.4 Overlap EBMT Algorithm

We are now in a position to give pseudocode for the overlap EBMT system. Given the set of output fragments *E* from the EBMT engine and a beam width, we have:

Algorithm 2.1: *OverlapTranslation*$(E, \textit{BeamWidth})$

```
Candidates ← BOS
Translation ← < EmptyTranslation >
while not empty(Candidates)
  do { F ← popBest(Candidates)
       NewCandidates ←
           Successors(F, E) ∪ EOS
       for each c′ ∈ NewCandidates
         do if UpdateScore(F, c′)
              then { comment: Best translation to end with c′
                     if c′ = EOS
                       then { Translation ←
                                BackTrace(c′)
                              delete(c′, NewCandidates) }
                   }
            else delete(c′, NewCandidates)
       comment: Update remaining candidates
       sort(NewCandidates ∪ Candidates)
       cut(Candidates, BeamWidth) }
Output Translation
```

For simplicity, we introduce two auxiliary fragments, *BOS* and *EOS* (for beginning and end of sentence, respectively). We assume the function *Successors(F,E)* returns the set of all fragments in *E* whose source starts after fragment *F* and can validly follow *F* — either source overlap of zero or both source and target overlap greater than zero. The function *UpdateScore(P,F)* looks-up the best score to end with fragment *P* in a table and updates the best score to end with fragment *F* if extending the path from *P* to *F* scores better; additionally, a backpointer from *F* to *P* is stored. The final translation is produced by tracing these backpointers.

Since the scoring function was carefully designed to depend only on the current fragment and its immediate predecessor, *UpdateScore* can be performed in constant time, $O(1)$. With a data-structure that provides a *popBest* and *popWorst*, the merge and sort can be done in $O(n)$ time instead of a full re-sort, $O(n \log n)$. Thus, even with an infinite beam, the overall complexity is $O(n^2)$ where n is the number of fragments output from the EBMT engine. Since the scoring function is additive and non-negative, a fragment is only popped from the candidate list once. A second time would imply that there was a cheaper path to some fragment which was not explored first, in contradiction to the best-first search. That is, the heuristic is admissible and with an infinite beam would be guaranteed to find the optimal candidate according to the scoring function. Finally, the score of the best full translation candidate can be used for early termination or candidate pruning.

3 Experiments

3.1 Data Set

All of the data used for the following experiments came from the Hansard corpus, which consists of parallel French and English versions of Canadian parliamentary debates and related documents. In all experiments the source language was French and the target language was English. The validation data used to optimize the parameters for the EBMT engine and the overlap scoring function consisted of two one-hundred sentence files selected arbitrarily. The test data consisted of ten different one-hundred sentence files also selected arbitrarily. At all times the training, test, and validation data sets were mutually exclusive. We constructed two training sets for the EBMT engine. The larger training set had 960,000 sentence pairs and the smaller set had 100,000 sentence pairs drawn from the large training set.

3.2 MT Quality Scoring Function

For empirical evaluation, we use the metric proposed by IBM, called *BLEU* [2] (Papineni et al., 2002) and the metric developed by NIST, 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. In our experiments this set consists of just the single reference translation provided by the Hansard transcripts.

To compute the BLEU score, one counts the number of n-word fragments (n-grams) in the candidate translations that have a match in the corresponding reference translations. The *n-gram precision* is this number divided by the total number of n-grams in the candidate translations. BLEU actually uses a *modified n-gram precision*, called p_n. This precision *clips* the count for each n-gram in any candidate translation to prevent it from exceeding the count of this n-gram in the best matching reference translation. The N different n-gram precisions are averaged geometrically (for BLEU $N = 4$ and $n = 1..4$), then multiplied by a *brevity penalty* to discourage short but high-precision translation candidates.[3] This leads to the following formula:

$$\text{BLEU} = e^{\min\left(1-\frac{r}{c},0\right)} . e^{\left(\sum_{n=1}^{N}(1/N)\log p_n\right)}.$$

Here, r and c are the total number of words in the reference and candidate translations respectively. The brevity penalty, $e^{\min\left(1-\frac{r}{c},0\right)}$, is less than one if $c < r$, *i.e.*, the candidate translations are shorter than the reference translations on average.

The NIST score is based on similar considerations, with three major differences. First, it incorporates an *information weight* to place more emphasis on infrequent n-grams. Second, it uses an arithmetic rather than geometric average to combine the scores for each n. Third, it uses a brevity penalty that is less sensitive to small variations.

The following formula describes the NIST metric:

$$\begin{aligned}\text{NIST} &= e^{\beta \log^2\left[\min\left(\frac{c}{r},1\right)\right]} \\ &* \sum_{n=1}^{N} \left(\frac{\sum_{\text{all co-occurring } w_1 \ldots w_n} \textit{Info}\,(w_1 \ldots w_n)}{|\text{all } w_1 \ldots w_n \text{ in candidate translation}|} \right).\end{aligned}$$

Here, n-grams up to $N = 5$ are considered, and β is set such that the brevity penalty is about 0.5 for a translation candidate length that is about $2/3$ of the reference translation length. The information weights are calculated over the reference translation corpus as:

$$\textit{Info}\,(w_1 \ldots w_n) = \log_2 \left(\frac{\#\text{ occurrences of } w_1 \ldots w_{n-1}}{\#\text{ occurrences of } w_1 \ldots w_n} \right).$$

[2]For BiLingual Evaluation Understudy.

[3]Note that it is not necessary to penalize overly long candidate translations as for those cases the n-gram precisions are already forced lower by the clipping.

Both BLEU and NIST scores are sensitive to the number of 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.

3.3 Experimental Conditions

We compare three different systems, each evaluated at their peak performance levels according to the BLEU scores for the two different sized sets of training data, for a total of six experimental conditions. We refer to our implementation of the overlap algorithm described above as *Overlap* below. In this section, we describe the other two systems we evaluated to understand the relative gains of the *Overlap* system. To ensure a peak-to-peak comparison of the highest possible performance of each system we optimized the parameters of all three systems empirically as described below.

Training Set	System	Mean BLEU	St. Dev. BLEU	Mean NIST	St. Dev. NIST
Small	Standard	0.1420	0.0409	4.9842	0.6610
	Overlap	**0.1640***	0.0380	**5.2327***	0.5711
	LM	0.1591*	0.0329	5.1306	0.6408
Large	Standard	0.1719	0.0379	5.4189	0.5191
	Overlap	**0.1900***	0.0386	5.6408*	0.5178
	LM	0.1897*	0.0428	**5.8247***	0.6389

Table 2: A performance summary of the described methods. The best performance for each measure is set in bold. Starred results are significant against the Standard system according to the t-test ($p \leq 0.01$). Underlined results are significant against the Standard system according to the sign test ($p \leq 0.05$).

3.3.1 Systems

Recall that *Overlap* uses the translation candidate fragments output by the EBMT engine described in Section 2.1. We have implemented a search mechanism over the EBMT engine output similar to *Overlap* except that it disallows fragment overlap to serve as a control system for our experiments. This system uses the same scoring function and search procedure as *Overlap* except that since overlap is not allowed to occur, the overlap coefficient is effectively set to zero. This search therefore is driven entirely by alignment scores on the fragments plus bonuses for longer fragments and a penalty for untranslated words. We refer to this control search mechanism over the EBMT engine output as *Standard* below.

We also compared the effectiveness of *Overlap* against our existing MEMT implementation when using only the EBMT engine and a language model component. Since the effective difference between this system and *Standard* is the language model, we refer to it as *LM*. The language model component of the MEMT system uses a trigram model of the target language (with smoothing and back-off where necessary) plus a number of other weighting factors to select the best-scoring overall translation with no overlapping fragments. The weighting factors include the quality scores assigned by the translation engines, the lengths of the fragments, and a penalty for untranslated words, just as are used in our new selection mechanism. The language model we used was built from approximately 250 million words of English text, including broadcast news transcriptions, Associated Press newswire, Wall Street Journal text, and other sources of news.

3.3.2 Parameter Optimization

It is important to note that the parameter optimizations for the EBMT engine and the overlap scoring function are with respect to the BLEU scoring method only. All optimizations were done *on the validation data*. The NIST scores for the same parameter settings are included in the results in Section 3.4 for completeness but have not been optimized.

The EBMT engine takes three parameters to define its search space. The engine sequentially searches through its parallel corpus of training data from the most recently added examples to the earliest added examples. As it searches, it returns matches between each unique source fragment of the input sentence and the corpus until any of the following conditions are met. First, the number of matches with perfect alignment scores equals the parameter-specified maximum (*Max-Perfect*). Second, the number of matches with alignment scores better than a given threshold equals the specified maximum (*Max-Near-Perfect*). Finally, the search ends when the number of matches considered equals a specified maximum (*Max-Alternatives*). Since these parameters control how many alternative translations a single source fragment has in the translation lattice, they interact with the size of the lattice and the quality of translations that can be produced by traversing the lattice.

We also optimized three of the parameters for the overlap scoring function: the coefficients for the length of the fragment (l), the length of the overlap (o); and the alignment score (a). Under the assumption that the ratio of these coefficients is more important than their specific values, we fixed the alignment coefficient to one and optimized the values for the length and overlap coefficients. This optimization was performed after optimizing the three EBMT parameters discussed above.

3.4 Experimental Results

In Table 2, we show results for each of the six experimental conditions described above. The highest performing systems are highlighted in the table for the small and large training sets. We also report two types of significance tests to compare the *Overlap* and *LM* systems against the *Standard* system. The first is a sign test. The null hypothesis is that each of the two systems being compared translates a given sentence better about half the number of times their performance score on a sentence differs. This test is performed on the set of individual sentence scores of the test set. 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. The significance levels of our results are summarized in Table 2.

The scores in Table 2 are evaluated using only one reference translation. Both BLEU and NIST metrics yield higher scores as the number of reference translations is increased. For example, a Chinese-to-English translation system evaluated on 993 sentences for which four reference translations were available as opposed to the average of scores obtained using the reference translations individually improved the BLEUscores by 85% and the NIST scores by 51%. The reader should bear this in mind when comparing the scores reported here to scores published elsewhere that evaluate performance over corpora with more reference translations.

Training Set	System	Avg. Words/Fragment
Small	Standard	2.92
	Overlap	3.02
Large	Standard	3.45
	Overlap	3.34

Table 3: Average number of target words per fragment in the *Overlap* and *Standard* systems.

4 Discussion

We conclude from the results presented above that the *Overlap* system is superior to the *Standard* system. For the large training set, *Overlap* provides a 10.53% improvement over the *Standard* method, and for the small training set it provides a 15.49% improvement. This shows that overlap is of greater importance when little training data is available.

There are several things to note about Figure 1.[4] First, it is clear that increasing the size of the training set increases the percentage of fragments in a sentence having overlap. However, the relative increase in training set size (10×) is much larger than the relative increase in overlap percentage. This difference indicates that we will get diminishing returns in overlap by further increasing the training set size. Another point to note about this figure is that even in the small training set, more than 75% of the translations use overlap (80% for the large training set).

In comparing the *Overlap* and *LM* systems, we note that the differences between the two are not statistically significant. Thus, the *Overlap* approach produces better or comparable translations. This is impressive given the fact that overlap is only used in translating 75-80% of the sentences. Additionally, we are currently pursuing a method of combining the *Overlap* and *LM* systems (Brown et al., 2003); current results indicate that using overlap in combination with a language model leads to significantly better results than using the either approach on its own. This further demonstrates that overlap and typical language model approaches take advantage of different types of information.

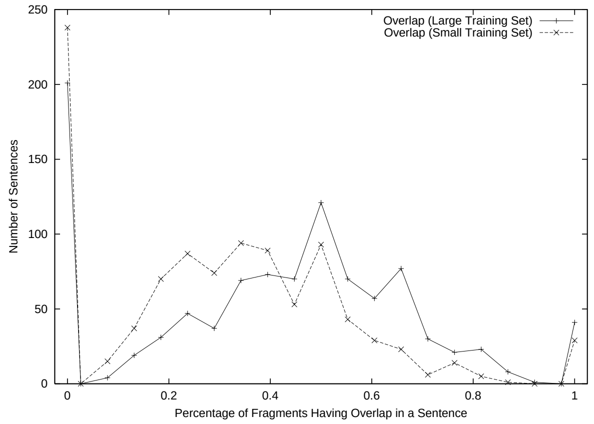

Figure 1: A histogram of the percentage of fragments in each output translation that have some overlap with their preceding fragments.

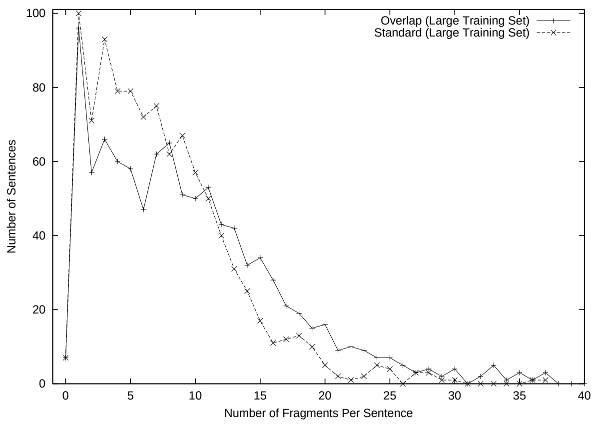

Figure 2: A histogram of the number of fragments combined to form the output translation for each sentence in the test set.

Figure 2 provides some insight into how *Overlap* produces better translations. This histogram shows that on average, the *Overlap* system uses more fragments per sentence than the *Standard* system. This trend also occurs when *Overlap* is compared to *Standard* using the small training set. To help understand how the *Overlap* system benefits from using more fragments, we are interested in whether the *Overlap* system is making use of longer or shorter fragments than the *Standard* system (Table 3). Surprisingly, the answer changes with the amount of training data available. In the small training data set, where the percentage of overlapping fragments is much lower, the *Overlap* system is able to use longer fragments than the *Standard* system (which cannot use the same fragments because they overlap). However, as the amount of training data increases and therefore the percentage of overlapping fragments, the *Overlap* method makes use of shorter

[4] To obtain the percentage of fragments in a sentence with overlap, the number having overlap is actually divided by the number of fragments minus one (since the first fragment has no predecessor to overlap).

fragments than the *Standard* method, but because of the frequent and consistent overlap in the fragments, it can still produce more accurate translations.

Based on the results of this paper, we are pursuing several promising extensions to this work. We are excited about the prospect of combining the *Overlap* method with other sources of information the EBMT system is given. For instance, we plan to investigate using overlap in conjunction with grammar rules.

5 Conclusions

In summary, we have presented an EBMT method 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. 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 EBMT because they overlap with each other. Additionally, systems benefit when overlap occurs frequently enough to take advantage of consistent translations of shorter fragments.

References

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Lafferty, J., Mercer, R., and Roossin., P. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, P., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312.

Brown, R., Hutchinson, R., Bennett, P. N., Carbonell, J. G., and Jansen, P. (2003). Reducing boundary friction using translation-fragment overlap. In Submission to *MT-Summit IX*.

Brown, R. D. (2001). Transfer-rule induction for example-based translation. In *Proceedings of the Workshop on Example-Based Machine Translation*. `http://www.eamt.org/summitVIII/papers/-brown.pdf`.

Frederking, R., Nirenburg, S., Farwell, D., Helmreich, S., Hovy, E., Knight, K., Beale, S., Domashnev, C., Attardo, D., Grannes, D., and Brown, R. (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.

Nagao, M. (1984). A framework of a mechanical translation between japanese and english by analogy principle. In Elithorn, A. and Banerji, R., editors, *Artificial and Human Intelligence*, pages 173–180. Elsevier Science Publishers B.V. Proceedings of the October, 1981, International NATO Symposium.

Nirenburg, S., Beale, S., and Domashnev, C. (1994). A full-text experiment in example-based machine translation. In *New Methods in Language Processing, Studies in Computational Linguistics*, Manchester, England.

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`.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 311–318.

Sumita, E. and Iida, H. (1991). Experiments and prospects of example-based machine translation. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL '91)*, pages 185–192.

Veale, T. and Way, A. (1997). Gaijin: A template-driven bootstrapping approach to example-based machine translation. In *Proceedings of the 1997 Conference on New Methods in Natural Language Processessing (NeMNLP'97)*, Sofia, Bulgaria.

Yamada, K. and Knight, K. (2002). A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 303–310.