Multi-Engine Machine Translation Guided by Explicit Word Matching

Shyamsundar Jayaraman and Alon Lavie

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213
USA
{shyamj, alavie}@cs.cmu.edu

Abstract. We describe a new approach for synthetically combining the output of several different Machine Translation (MT) engines operating on the same input. The goal is to produce a synthetic combination that surpasses all of the original systems in translation quality. Our approach uses the individual MT engines as "black boxes" and does not require any explicit cooperation from the original MT systems. An explicit word matcher is first used in order to identify the words that are common between the MT engine outputs. A decoding algorithm then uses this information, in conjunction with confidence estimates for the various engines and a trigram language model in order to score and rank a collection of sentence hypotheses that are synthetic combinations of words from the various original engines. The highest scoring sentence hypothesis is selected as the final output of our system. Experiments conducted using three Chinese-to-English online translation systems demonstrate that our multi-engine combination system provides an improvement of about 6% over the best original system, and is about equal in translation quality to an "oracle" capable of selecting the best of the original systems on a sentence-by-sentence basis. A second oracle experiment shows that our new approach produces synthetic combination sentence hypotheses that are far superior to the hypotheses currently selected by the system, but our current scoring is not yet capable of adequately identifying the best hypothesis.

1. Introduction

A variety of different paradigms for machine translation (MT) have been developed over the years, ranging from statistical systems that learn mappings between words and phrases in the source language and their corresponding translations in the target language, to Interlingua-based systems that perform deep semantic analysis. All of these approaches have well-known advantages and disadvantages. Corpus-based systems, such as statistical machine translation and example-based machine translation, provide broad coverage but rarely produce translations of quality that is comparable to that achieved by knowledge-based systems, which involve deeper syntactic and semantic analysis. Knowledge-Based systems that incorporate syntactic and semantic knowledge provide high-quality translations but require large amounts of human time to create and rarely have the coverage of corpus-based systems.

With such a wide range of approaches to machine translation, it would be beneficial to have an effective framework for combining these systems into an MT system that carries many of the advantages of the individual systems and suffers from few of their disadvantages. Attempts at combining outputs from different systems have proved useful in other areas of language technologies, such as the ROVER approach for speech recognition (Fiscus 1997). Several different multi-engine machine translation (MEMT) systems have also been explored in the past ten years, starting with the Pangloss system in 1994 (Frederking and Nirenburg). Several of these systems require significant coupling between the systems in the form of shared lattice structures (Frederking et al 1997; Tidhar & Küssner 2000; Lavie, Probst et al. 2004). Beyond the difficulty of obtaining compatible lattice representations of the various input systems, these approaches require standardizing confidence scores that come from the individual engines. Another proposed MEMT approach uses string alignment between the different translations and trains a finite state machine to produce a consensus translation (Bangalore et al. 2001). The alignment algorithm described in that work is the standard Levenshtein edit distance which only allows insertions, deletions and substitutions. This model does not accurately capture phrase movement like:

In the street, the children cried.

The children cried in the street.

The standard Levenshtein distance would create an alignment by first deleting the words "The children cried" in the second sentence and inserting them at the end of the sentence. By creating an alignment this way, the fact that the phrase "the children cried" occured in both sentences is lost.

In this paper, we propose a new way of combining the translations of multiple MT systems based on a more versatile word alignment algorithm. A "decoding" algorithm then uses these alignments, in conjunction with confidence estimates for the various engines and a trigram language model, in order to score and rank a collection of sentence hypotheses that are synthetic combinations of words from the various original engines. The highest scoring sentence hypothesis is selected as the final output of our system. We experimentally tested the new approach by combining translations obtained from three Chinese-to-English online translation systems. Translation quality is scored using the METEOR MT evaluation metric (Lavie, Sagae et al 2004; Lavie and Banerjee, 2005). Our experiments demonstrate that our new multi-engine combination system achieves an improvement of about 6% over the best original system, and is about equal in translation quality to an "oracle" capable of selecting the best of the original systems on a sentence-by-sentence basis. A second oracle experiment shows that our new approach produces synthetic combination sentence hypotheses that are even far superior to the hypotheses currently selected by the system, but our current scoring is not yet capable of adequately identifying the best combination.

The remainder of this paper is organized as follows. In section 2 we describe the algorithm for generating hypothesis sentence translations. Section 3 describes the experimental setup used to evaluate our new approach, and section 4 presents the results of our evaluation. Our conclusions and future work are presented in section 5.

2. The MEMT Algorithm

Our Multi-Engine Machine Translation (MEMT) system operates on the single "top-best" translation output produced by each of several MT systems operating on a common input sentence. MEMT first aligns the words of the output strings produced by different translation systems using a word matching sub-module. Then, using the alignments provided by the matcher, the system generates a set of synthetic sentence hypothesis translations. Each hypothesis translation is assigned a score based on the alignment information, the confidence of the individual systems, and a language model trained on a large target language corpus. The hypothesis translation with the best score is selected as the final output of the MEMT combination.

2.1. The Word Alignment Matcher

The task of the matcher is to produce a word-to-word alignment matching between the words of two given input strings. Identical words, ignoring case, that appear in both input sentences are potential matches. Since the same word may appear multiple times in the sentence, there are multiple ways to produce an alignment between the two input strings. This phenomenon is especially common with function words. The goal is to find the alignment that represents the best correspondence between the strings. This alignment is defined as the alignment that has the smallest number of "crossing edges", when a line is drawn between the two matched words in the two sentences. For example, let us consider the alignment between the following two sentences:

The boy walked the dog

A child walked the dog

Since “walked” and “dog” occur only once in each sentence, their alignments are fixed. The word “the” appears twice in the first sentence and only once in the second sentence. Therefore, there are two possible alignments for this pair of sentences. The intuitive alignment aligns the “the” in front of “dog” in the first sentence, with the “the” in front of “dog” in the second sentence. The alignment, when drawn, looks like this:

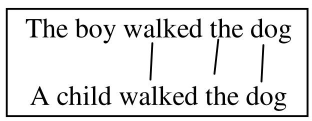

Figure 1: An Example of a Good Alignment

In this alignment, there are no crossing edges. The other alignment would align the “the” that is followed by “boy” in the first sentence with the “the” that is followed by “dog” in the second sentence. When drawn, the alignment would look like this:

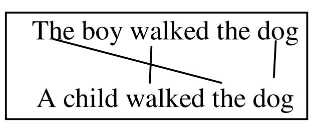

Figure 2: An Example of a Poor Alignment

In this alignment there is one pair of crossing edges. The matcher, in this case, would prefer the first alignment because there are fewer crossing edges and is the better correspondence.

The basic matching algorithm is extended to allow matches between words that are not identical but have the same stem. Stemming, using the porter stemmer, can be done on the original input strings being aligned, or as a second step, on the remaining un-aligned words, after exact matches have been found.

The matcher described so far works for a pair of sentences, but how does this extend to the alignment of more than two sentences? The matcher simply produces alignments for all pair-wise combinations of the sentences. Though these alignments are not guaranteed to be commutative, in practice they generally are. In the cases where the matches are not commutative, the MEMT system would perform suboptimally, but would not break.

In the first stage of our MEMT approach, we apply the word-alignment matcher to translation outputs of the individual MT systems for a given input sentence. In the context of its use within our MEMT approach, the word-alignment matcher provides three main benefits. First, it explicitly identifies translated words that appear in multiple MT translations. This reinforcement increases the score assigned to such aligned words in the synthetic combinations produced by the MEMT algorithm. Second, the alignment information allows the algorithm to ensure that aligned words are not included in a synthetic combination more than once. And finally, by allowing long distance alignments, the synthetic combination generation algorithm can consider different plausible orderings of the matched words, based on their location in the original translations.

2.2. Basic Hypothesis Generation

After the matches have been found, the decoder goes to work. The hypothesis generator produces synthetic combinations of words and phrases from the original translations that satisfy a set of adequacy constraints. The generation algorithm is an iterative process and produces these translation hypotheses incrementally. Within each iteration, the existing set of partial hypotheses is extended by incorporating an “unused” word from one of the original translations. A data structure keeps track of the accounted for “used” words that are associated with any partial hypothesis. One underlying constraint observed by the generator is that the original translations are considered in principle to be word synchronous in the sense that selecting a word from one original translation normally implies “marking” a corresponding word from each of the other original translations as “used”. The way this is determined is explained below. Two partial hypotheses that have the same partial translation, but have a different set of words that have been accounted for are considered different. A hypothesis is considered "complete" if one or more original translation strings propose to end the sentence. At the start of hypothesis generation, there is one hypothesis, which has an empty string for its partial translation and does not account for any words in any of the individual systems.

In each iteration, the decoder extends a hypothesis by choosing the first unused word from one of the original translations or chooses to end the hypothesis, if a sentence-end is proposed by one or more of the original translations. A translation proposes a sentence-end when its last word was marked as used in the previous iteration. When the decoder chooses to extend a hypothesis by selecting word *w* from some MT output A, the decoder marks *w* as used. The decoder then proceeds to identify and mark as used a word in each of the other original translations. If *w* is aligned to words in any of the other original translations, then the words that are aligned with *w* are also marked as used. For example, in Figure 3, if the MEMT system chooses "the" from the first translation, the system also marks word six in the second translation because of the alignment generated by the matcher.

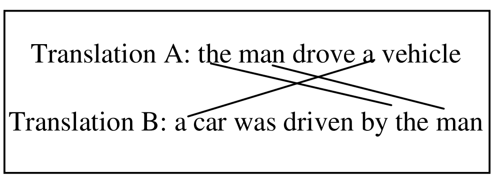

Figure 3: An Example Alignment

For original translations that do not have a word that aligns with *w*, we attempt to identify a word that is likely to be a different translation of the source language word that corresponds to *w*. With this in mind, the decoder tries to create an "artificial alignment" between *w* and a word in the original translation, where there was no alignment found by the matcher. The default choice is to select the first unused word that is not aligned to a word in any translation from which a word has already been used. If there isn't a word that matches this criterion, then the decoder marks this translation as used and proceeds. Otherwise, it marks this word and any words that are aligned to it as used. The decoder repeats this process until all translations have been accounted for. It is important to note that the order in which the translations are processed matters. For instance, suppose in the following example that the chosen word is "truck" from system C:

Translation A: green car drove street
Translation B: car drove around road
Translation C: truck puttered down road

Since the matcher would have found the aligment between the instances of "car", "drove", and "road", there are two possible ways to create an artificial alignment for the word "truck". The first one results from the decoder creating an artificial alignment first to translation A and then translation B. This alignment is shown in Figure 4.

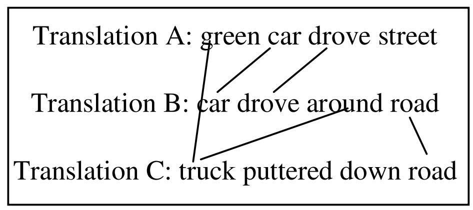

Figure 4: Example of an Artificial Alignment

"truck" was forced to align with "around" in translation B, because after aligning "truck" to "green", both words "car" and "drove" in translation B aligned to words in translation A, for which an alignment was already found. Therefore these words are not considered for artificial alignments.

If, on the other hand, the decoder first tried to create an artificial alignment between "truck" and a word in System B, then the resulting alignment would look like the Figure 5.

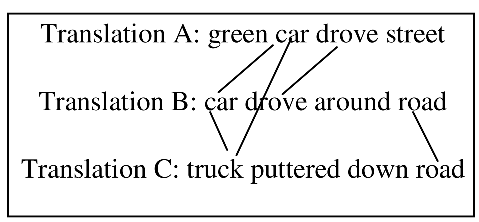

Figure 5: Example of Another Artificial Alignment

When there are different artificial alignments that are valid, the decoder produces a hypothesis for each one of the artificial alignments. The decoder continues to iterate until all expandable hypotheses, those that have not been completed, have been expanded.

2.3. Scoring of Hypotheses

In the final stage of the MEMT algorithm, the hypotheses produced by the hypothesis generator are scored and ranked, and the highest scoring hypothesis is selected as the final translation. Our current scoring includes two components – a language model and a confidence score assigned to each word that is included in the hypothesis. The language model score and the confidence score are combined multiplicatively to produce a joint score. In order to normalize scores across hypotheses of different lengths, we then calculate a geometric average per word score and use this score to rank teh various hypotheses that were created during decoding. This prevents an inherent bias for shorter hypotheses that is present in multiplicative cumulative scores.

2.3.1. The Confidence Score

One component in the scoring is a confidence score that is assigned to each of the words in the synthetic hypothesis. Each word supplied by any one of the original systems is given a confidence score equal to the confidence score associated with the system which produced it. If a word is aligned to a word in any of the other original systems, the confidence scores of the two systems are summed together to produce the confidence score for the word. This provides strong reinforcement for words that appear in multiple original system outputs.

There are many ways to set the confidence scores of the original systems. One way is to learn the optimal scores using a training set. This could be time consuming since many runs of MEMT are necessary to get the optimal value. One heuristic solution, *average score*, is to score the original systems on a common test set using a standard evaluation metric and then calculate confidences that are proportional to teh relative ratios of the resulting system scores. Another heuristic metric, *sentence count*, sets the scores to be relative to the percentage of time that any one system produces the best translation for a sentence. We have experimented with both methods.

2.3.2. Language Model Score

The language model score is assigned by a standard trigram language model that is trained on large corpora of target language text. The trigram language model assigns a probability score to the hypothesis translation that approximates how likely it is for the hypothesis translation to occur in the target language. This rewards synthetic hypotheses that are more fluent and grammatical.

2.4. Alignment Horizon

One of the problems that we identified in our early experiments is that the original process for marking used words in some cases allowed words that were not pre-aligned by the matcher to "linger" unused within their original system too long. This resulted in the lingering words being placed at later points in the hypothesis, where the language model score was high, even though the word did not belong at this point in the translation. To alleviate this problem, we introduced the notion of a horizon, which is defined as the number of words behind the current word in the hypothesis, for which a lingering word can still be considered for incorporation into the hypothesis. This horizon does not affect words that are aligned to words in other original systems and that are within the horizon. The MEMT system only discards words for which the word itself and all words to which it is aligned are beyond the horizon. Since at any time the MEMT system might have taken different number of words from different translation systems, the notion of "current" word is not clearly defined. For the sake of simplicity, the system defines the current word in this context as the *nth* word in a system, where *n* is the number of words in the partial translation hypothesis.

Let us look at an example. Assume that the alignment horizon is set to three, the partial translation hypothesis is "the sri lankan prime minister criticizes", and one of the original translations is "The President of the Sri Lankan Prime Minister Criticized the President of the Country." Since the length of the partial translation is six, the current word of the system output is six. With the lingering word horizon set at three, any word earlier than the third word is beyond the horizon. If either "the" or "president" was aligned to words in another MT system's output that are past the second word in the other output, these words would still be considered useful.

2.5. Matching Window

When choosing to artificially align an unaligned word to a selected word, our original algorithm allowed unlimited long distance alignments, which we found to be experimentally problematic. We consequently added a parameter to the algorithm that restricts how far ahead the decoder can look to artificially align a chosen word. The matching window is defined as the number of words ahead of the "current" word in the unused system the decoder is searching for an artificial alignment. When the decoder tries to artificially align word *w* from system A to a word in an unused system B, it first needs to calculate where the "current" word of system B is. The decoder finds the last word in the partial translation of the hypothesis where the word *x* was in both system A and B. If no such word *x* is found, then the decoder chooses the beginning of sentence marker as *x*. Then the decoder calculates the number of words in system A between *w* and *x*. It adds this value to the position of *x* in system B to produce the "current" word of B.

System A: The boy walked the large dog

System B: The child who is male walked a big pet

Figure 6: Example of a Matching Window

For example, let us take the two hypothesis translations and the alignments shown in Figure 6. If the partial translation is "The boy walked" and *w* is "the", then decoder first searches for the last word in the partial translation which existed in both systems A and B. In this case the word is "walked." The word "walked" is one word behind "the", which is the word for which the decoder is trying to create an artificial alignment. Therefore the current word in system B is one word after "walked", which in this case is the word "a". The window size parameter determines how many words ahead may be considered for an unaligned correspondence. If this parameter is set to a value of one, then "pet" cannot be considered as a potential artificial alignment for the word "the".

2.6. Part-of-Speech Based Matching

As stated earlier, "artificial" alignments are hypothesized in order to avoid incorporating words that are alternatives of each other in the same synthetic translation. A major issue that can arise from creating artificial alignments is that these alignments might not be valid, since there is very little evidence to support them. In the basic artificial alignment algorithm, nothing prevents a proper name from being aligned to the article "the". We discovered that this produced incoherent hypotheses that were sometimes still scored highly by the language model. This could lead to dropping content words in favor of function words in order to get a higher language model score. To address this, we modified the algorithm to only allow artificial matching between words that have the same part of speech. This prevents function words, which are generally in a closed part-of-speech class, from being matched with a content word. The system uses a dictionary that lists possible parts of speech for words in the target language. There is still an issue of coverage, since the dictionary cannot possibly have all the words and proper names in any particular language, but it does provide good coverage for the function words, which are the most problematic.

2.7. A Complete Example

Now that we have described all parts of the algorithm, let us see how one possible synthetic translation is produced from a set of original translations and alignments that were produced by the matcher. For this example, the alignment horizon is set to two and the matching window is set to one. The original translations and alignments are shown in Figure 7. We will walk through each iteration to see how the partial translation "green truck drove down road" is generated.

In the first iteration, the MEMT system chooses the word "green" from the translation A. Since there are no explicit alignments for this word, it attempts to find artificial alignments in translations B and C. The MEMT system calculates the boundary of the matching window for each translation. In both translations B and C, the matching window ends at the second word. The system first tries to create an artificial alignment between “green” and a word in translation B, but fails. The first word does not match the part of speech of “green” so that alignment is not valid. The second word is aligned to a word in translation A, so it cannot be used for an artificial alignment. Therefore, the MEMT system does not mark any words in translation B as used. When trying to create an artificial alignment between “green” and a word in translation C, the MEMT system finds that no alignments can be made, since the first two words in translation C are not adjectives. After the first iteration, the used words structure is shown in Figure 8.

Translation A: green car drove street

Translation B: the car drove around road

Translation C: truck puttered down road

Figure 7: Original Alignments for the Example

Translation A: ~~green~~ car drove street

Translation B: the car drove around road

Translation C: truck puttered down road

Figure 8: After First Iteration

In the next iteration, the MEMT system chooses “truck” from translation C. The boundary of the matching windows in translations A and B is at the second word, since the word that aligns with “truck” is expected to be in the first word. The system first creates an artificial alignment with “car” in translation A, which forces an artificial alignment with “car” in translation B. The used words data structure after the second iteration is shown in Figure 9.

Next, the MEMT system chooses “drove”, from translation A. This aligns with “drove” from translation B. Since there has been no word chosen so far that has had an explicit alignment between translation A and translation C, the word that aligns with “drove” in translation C is expected at the third word. Therefore the boundary of the matching window is at the fourth word in translation C. Since puttered is a verb, the system creates an artificial alignment between “drove” and “puttered”. This leads the used word structure shown in Figure 10.

Translation A: ~~green~~ ~~car~~ drove street

Translation B: the ~~car~~ drove around road

Translation C: ~~truck~~ puttered down road

Figure 9: After Second Iteration

Translation A: ~~green~~ ~~car~~ ~~drove~~ street

Translation B: the ~~car~~ ~~drove~~ around road

Translation C: ~~truck~~ ~~puttered~~ down road

Figure 10: After Third Iteration

In the fourth iteration, the MEMT chooses “down” from translation C. The system calculates the matching window for translations A and B. Since no word chosen in the partial translation has an explicit alignment between C and either A or B, the system expects the word that aligns with “down” to be the third word in both translations A and B. The boundary of the window is the fourth word for both of these original translations. It then tries to align “down” to word in translation A, but fails since the only available word is “street”, which is a noun and not a prepostion. In translation B, the system could not align “down” to “the” because the parts of speech don’t match, and hence creates an artificial alignment to “around”. At the end of the fourth iteration, the alignment horizon finally plays a role. The partial translation has four words so far and the alignment horizon is two, so any unaligned words in the original translation before the second word are no longer considered useful. In our example, "the", in translation B, fails the alignment horizon check. Therefore, it is marked as used as well. After the fourth iteration, the used words structure is shown in Figure 11.

Translation A: ~~green~~ ~~car~~ ~~drove~~ street

Translation B: ~~the~~ ~~car~~ ~~drove~~ ~~around~~ road

Translation C: ~~truck~~ ~~puttered~~ ~~down~~ road

Figure 11: After Fourth Iteration

In the fifth iteration, the system chooses "road" from the translation B, which is aligned to "road" from translation C. The system expects the word corresponding to "road" in translation A to be two words after "drove". Therefore the matching window boundary would be two words after "drove". The system aligns "road" to "street" since they both are nouns. Figure 12 shows the used word data structure after the fifth iteration.

Translation A: ~~green~~ ~~car~~ ~~drove~~ ~~street~~

Translation B: ~~the~~ ~~car~~ ~~drove~~ ~~around~~ ~~road~~

Translation C: ~~truck~~ ~~puttered~~ ~~down~~ ~~road~~

Figure 12: After Fifth Iteration

In the last iteration, all three systems have no more words left, so they all propose a sentence-end. Therefore the MEMT system would mark "green truck drove down road" as a completed translation and score it.

3. Experimental Setup

For our experimental evaluation, we combined outputs of three online machine translation systems (Systran, Netat, and Wordlingo) on the TIDES 2002 Chinese and TIDES 2003 Chinese evaluation sets. These test sets consisted of roughly 900 sentences each, of news wire text in simplified Chinese.

3.1. Evaluation Methodology

We used the 2002 data as a training set to obtain confidence scores for each of the original MT engines. We used the *sentence count* heuristic described previously.

We compare the results of MEMT to the individual online machine translation systems. We also compare the performance of MEMT to the score of an "oracle system" that chooses the best scoring output of the original systems for each sentence, as assessed by an aautomatic MT evaluation metric. Note that this "oracle" is not a realistic system, since a real system cannot determine at run-time which system is best on a sentence by sentence basis.

One goal of the evaluation was to see how rich the space of synthetic translations produced by our hypothesis generator is. To this end, we also compare the output selected by our current MEMT system to an "oracle system" that chooses the best synthetic translation that was generated by the decoder for each sentence. According to teh external MT evaluation metric. This too is not a realistic system, but it allows us to see how well our hypothesis scoring currently performs, and also approximates an upper-bound on how well we can expect our MEMT approach to perform in practice (Hogan and Frederking 1998).

Due to the computational complexity of running the Oracle system, several practical restrictions were imposed. First, the Oracle system only had access to the top 1000 translation hypotheses produced by MEMT for each sentence. While this does not guarantee finding the best translations that the decoder can produce, this method provides a good approximation. We also only ran the Oracle experiment on the first 149 sentences of the test sets due to computational time constraints.

All the system performances are measured using the METEOR MT evaluation metric (Lavie, Sagae et al., 2004; Lavie and Banerjee, 2005). METEOR was chosen since it is more reliable at scoring sentence level translations than BLEU, a property that is needed in order to run the proposed Oracle experiments. METEOR produces scores in the range of [0,1], based on a combination of unigram precision, unigram recall and an explicit penalty related to the average length of matched segments between the evaluated translation and its reference.

4. Results

On the 2002 TIDES data, the three original systems had similar METEOR scores. Table 1 shows the scores of the online systems, with the names changed to protect the privacy of the original systems. It also shows the score of MEMT's output as well as the score of the oracle system that chooses the best original translation on a sentnce-by-sentence basis. The score of the MEMT system does not surpass, but is in close range to the score of the oracle system. MEMT scored significantly higher than any of the online translators which it combined.

System	METEOR Score
Online Translator A	.5225
Online Translator B	.5309
Online Translator C	.5225
Oracle (best original)	.5740
MEMT System	**.5673**

Table 1: Scores on Full TIDES 2002 Dataset

Table 2 shows the METEOR scores on the trimmed down sample of the TIDES 2002 Dataset where we performed an oracle experiment on all the hypotheses generated by the decoder. In this sample of the dataset, the ordering stays the same, but scores were slightly higher. The hypotheses produced by the MEMT are still significantly better than any of the original sentences. The oracle system that selects that best hypothesis generated by MEMT is significantly better than the current MEMT system. MEMT was able to achieve 37% of the maximal improvement that was possible from the hypotheses that were produced.

System	METEOR Score
Online Translator A	.5314
Online Translator B	.5453
Online Translator C	.5321
Oracle (best original)	.5821
MEMT	**.5762**
Oracle (best hyp)	.6268

Table 2: Scores on Subset of TIDES 2002 Dataset

Table 3 shows the METEOR scores for the different online systems, the oracle that chose the best sentence from the original translations, and the MEMT system, on the full TIDES 2003 dataset. Overall, scores are somewhat lower for this dataset and the difference between the best online system and the worst online system is larger. Once again, MEMT produced translations that are higher in quality than any of the original web translations. MEMT again come close, but does not surpass the oracle system that chooses the best original translation.

System	METEOR Score
Online Translator A	.4886
Online Translator B	.5047
Online Translator C	.4855
Oracle (best original)	.5440
MEMT System	**.5347**

Table 3: Scores on Full TIDES 2003 Dataset

Table 4 shows the METEOR scores for the three online translators, the oracle system that chose the best original translation, MEMT, and the oracle system which chose the best hypothesis generated by MEMT on the subset of the 2003 TIDES dataset that was used for the oracle experiment. It is interesting to note that the second online translator performed worse while all the other systems performed better on this sample of the test set. Even though the confidence scores for the online systems were not tuned for the 2003 dataset, the MEMT scoring algorithm performed better on the 2003 dataset than on the 2002 dataset. On the sample of the 2003 dataset, the MEMT produced 41% of the improvement that is possible from the hypotheses that are generated.

System	METEOR Score
Online Translator A	.4917
Online Translator B	.4859
Online Translator C	.4910
Oracle (best original)	.5381
MEMT System	**.5301**
Oracle (best hyp)	.5840

Table 4: Scores on Subset of TIDES 2003 Dataset

5. Conclusion and Future Work

The MEMT algorithm described in this paper synthetically combines the output of several different MT engines operating on the same input, using the individual MT engines as "black boxes". Our experiments demonstrate that our new multi-engine combination system achieves an improvement of about 6% over the best original system, and is about equal in translation quality to an "oracle" capable of selecting the best of the original systems on a sentence-by-sentence basis. The MEMT decoder produces hypotheses that are even far superior in translation quality, but our current scoring algorithm is not yet capable of selecting the best generated hypothesis.

We are currently in the process of more rigorous testing of the features and parameters of the hypothesis generation and scoring module. The parameters, specifically the alignment horizon and the matching window, chosen for the MEMT system are based on early stages of experimentation. We are investigating whether these parameters are dependent on the quality of the specific MT systems that are combined, and whether they are dependent on other properties of the original systems.

We plan to focus significant effort on improving the scoring mechanism within the decoder. As indicated by our oracle experiments, the decoder is already producing far better synthetic hypotheses. We hope to identify salient features that will help us further improve scoring and hypothesis selection.

Another potential area for improvement is in the capabilities of the word matcher. We are developing a capability for detecting and matching synonymous words, using synsets from WordNet. This capability should significantly help us find what we termed "artificial word alignments".

6. Acknowledgements

The authors wish to thank Robert Frederking and Ralf Brown for their insightful comments and suggestions in the course of this work. This work was supported by a grant from the US Department of Defense.

7. References

Bangalore, S., G. Bordel, and G. Riccardi (2001). 'Computing Consensus Translation from Multiple Machine Translation Systems'. In Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop, Italy.

Fiscus, J. G.(1997). 'A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER)'. In IEEE Workshop on Automatic Speech Recognition and Understanding.

Frederking, R. and S. Nirenburg (1994). 'Three Heads are Better than One'. In Proceedings of the Fourth Conference on Applied Natural Language Processing (ANLP-94), Stuttgart, Germany.

Hogan, C. and R. Frederking (1998). 'An Evaluation of the Multi-engine MT Architecture'. In Proceedings of the Third Conference of the Association for Machine Translation in the Americas, pp. 113-123. Springer-Verlag, Berlin

Lavie, A., K. Probst, E. Peterson, S. Vogel, L.Levin, A. Font-Llitjos and J. Carbonell (2004). 'A Trainable Transfer-based Machine Translation Approach for Languages with Limited Resources'. In Proceedings of Workshop of the European Association for Machine Translation (EAMT-2004), Valletta, Malta

Lavie, A., K. Sagae and S. Jayaraman (2004). 'The Significance of Recall in Automatic Metrics for MT Evaluation'. In Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004), Washington, DC

Lavie, A. And Satanjeev Banerjee. `The METEOR Automatic Machine Translation Evaluation System'. http://www.cs.cmu.edu/~alavie/METEOR/

Tidhar, Dan and U. Kessner (2000). `Learning to Select a Good Translation'. In Proceedings of the 17th International Conference on Computational Linguistics (COLING-2000), Saarbrcken, Germany.