Improving Alignments for Better Confusion Networks for Combining Machine Translation Systems

Necip Fazil Ayan and **Jing Zheng** and **Wen Wang**
SRI International
Speech Technology and Research Laboratory (STAR)
333 Ravenswood Avenue
Menlo Park, CA 94025
{nfa,zj,wwang}@speech.sri.com

Abstract

The state-of-the-art system combination method for machine translation (MT) is the word-based combination using confusion networks. One of the crucial steps in confusion network decoding is the alignment of different hypotheses to each other when building a network. In this paper, we present new methods to improve alignment of hypotheses using word synonyms and a two-pass alignment strategy. We demonstrate that combination with the new alignment technique yields up to 2.9 BLEU point improvement over the best input system and up to 1.3 BLEU point improvement over a state-of-the-art combination method on two different language pairs.

1 Introduction

Combining outputs of multiple systems performing the same task has been widely explored in various fields such as speech recognition, word sense disambiguation, and word alignments, and it had been shown that the combination approaches yielded significantly better outputs than the individual systems. System combination has also been explored in the MT field, especially with the emergence of various structurally different MT systems. Various techniques include hypothesis selection from different systems using sentence-level scores, re-decoding source sentences using phrases that are used by individual systems (Rosti et al., 2007a; Huang and Papineni, 2007) and word-based combination techniques using confusion networks (Matusov et al., 2006; Sim et al., 2007; Rosti et al., 2007b). Among these, confusion network decoding of the system outputs has been shown to be more effective than the others in terms of the overall translation quality.

One of the crucial steps in confusion network decoding is the alignment of hypotheses to each other because the same meaning can be expressed with synonymous words and/or with a different word ordering in different hypotheses. Unfortunately, all the alignment algorithms used in confusion network decoding are insensitive to synonyms of words when aligning two hypotheses to each other. This paper extends the previous alignment approaches to handle word synonyms more effectively to improve alignment of different hypotheses. We also present a two-pass alignment strategy for a better alignment of hypotheses with similar words but with a different word ordering.

We evaluate our system combination approach using variants of an in-house hierarchical MT system as input systems on two different language pairs: Arabic-English and Chinese-English. Even with very similar MT systems as inputs, we show that the improved alignments yield up to an absolute 2.9 BLEU point improvement over the best input system and up to an absolute 1.3 BLEU point improvement over the old alignments in a confusion-network-based combination.

The rest of this paper is organized as follows. Section 2 presents an overview of previous system combination techniques for MT. Section 3 discusses the confusion-network-based system combination. In Section 4, we present the new hypothesis alignment techniques. Finally, Section 5 presents our experiments and results on two language pairs.

2 Related Work

System combination for machine translation can be done at three levels: Sentence-level, phrase-level or word-level.

Sentence-level combination is done by choosing one hypothesis among multiple MT system outputs (and possibly among n-best lists). The selection criterion can be a combination of translation model and language model scores with multiple comparison tests (Akiba et al., 2002), or statistical confidence models (Nomoto, 2004).

Phrase-level combination systems assume that the input systems provide some internal information about the system, such as phrases used by the system, and the task is to re-decode the source sentence using this additional information. The first example of this approach was the multi-engine MT system (Frederking and Nirenburg, 1994), which builds a chart using the translation units inside each input system and then uses a chart walk algorithm to find the best cover of the source sentence. Rosti et al. (2007a) collect source-to-target correspondences from the input systems, create a new translation option table using only these phrases, and re-decode the source sentence to generate better translations. In a similar work, it has been demonstrated that pruning the original phrase table according to reliable MT hypotheses and enforcing the decoder to obey the word orderings in the original system outputs improves the performance of the phrase-based combination systems (Huang and Papineni, 2007). In the absence of source-to-target phrase alignments, the sentences can be split into simple chunks using a recursive decomposition as input to MT systems (Mellebeek et al., 2006). With this approach, the final output is a combination of the best chunk translations that are selected by majority voting, system confidence scores and language model scores.

The word-level combination chooses the best translation units from different translations and combine them. The most popular method for word-based combination follows the idea behind the ROVER approach for combining speech recognition outputs (Fiscus, 1997). After reordering hypotheses and aligning to each other, the combination system builds a confusion network and chooses the path with the highest score. The following section describes confusion-network-based system combination in detail.

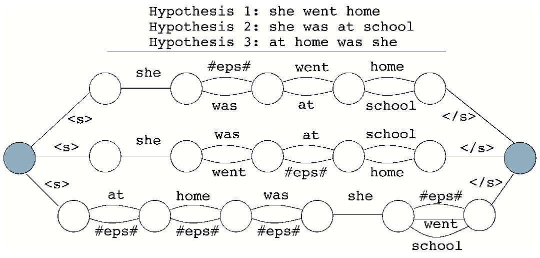

Figure 1: Alignment of three hypotheses to each other using different hypotheses as skeletons.

3 System Combination with Confusion Networks

The general architecture of a confusion-network-based system combination is as follows:

1. Extract n-best lists from MT systems.
2. Pick a skeleton translation for each segment.
3. Reorder all the other hypotheses by aligning them to the skeleton translation.
4. Build a confusion network from the reordered translations for each segment.
5. Decode the confusion network using various arc features and sentence-level scores such as LM score and word penalty.
6. Optimize feature weights on a held-out test set and re-decode.

In this framework, the success of confusion network decoding for system combination depends on two important choices: Selection of the skeleton hypothesis and alignment of other hypotheses to the skeleton.

For selecting the best skeleton, two common methods are choosing the hypothesis with the Minimum Bayes Risk with translation error rate (TER) (Snover et al., 2006) (i.e., the hypothesis with the minimum TER score when it is used as the reference against the other hypotheses) (Sim et al., 2007) or choosing the best hypotheses from each system and using each of those as a skeleton in multiple confusion networks (Rosti et al., 2007b). In this paper, we use the latter since it performs slightly better than the first method in our experiments. An example confusion network on three translations is presented in Figure 1.[1]

[1] In this paper, we use multiple confusion networks that are attached to the same start and end node. Throughout the rest of the paper, the term *confusion network* refers to one network among multiple networks used for system combination.

The major difficulty when using confusion networks for system combination for MT is aligning different hypotheses to the skeleton since the word order might be different in different hypotheses and it is hard to align words that are shifted from one hypothesis to another. Four popular methods to align hypotheses to each other are as follows:

1. Multiple string-matching algorithm based on Levenshtein edit distance (Bangalore et al., 2001)
2. A heuristic-based matching algorithm (Jayaraman and Lavie, 2005)
3. Using GIZA++ (Och and Ney, 2000) with possibly additional training data (Matusov et al., 2006)
4. Using TER (Snover et al., 2006) between the skeleton and a given hypothesis (Sim et al., 2007; Rosti et al., 2007b)

None of these methods takes word synonyms into account during alignment of hypotheses.[2] In this work, we extend the TER-based alignment to use word stems and synonyms using the publicly available WordNet resource (Fellbaum, 1998) when aligning hypotheses to each other and show that this additional information improves the alignment and the overall translation significantly.

4 Confusion Networks with Word Synonyms and Two-pass Alignment

When building a confusion network, the goal is to put the same words on the same arcs as much as possible. Matching similar words between two hypotheses is necessary to achieve this goal.

When we align two different hypotheses using TER, it is necessary that two words have the identical spelling to be considered a match. However, in natural languages, it is possible to represent the same meaning using synonyms of words in possibly different positions. For example, in the following sentences, “*at the same time*” and “*in the meantime*”, “*waiting for*” and “*expect*”, and “*set*” and “*established*” correspond to each other, respectively:

Skeleton: `at the same time expect israel to abide by the deadlines set by .`

Hypothesis: `in the meantime , we are waiting for israel to abide by the established deadlines .`

Using TER, synonymous words might be aligned to each other if they appear in the same position in two hypotheses but this is less likely when two words appear in different positions. Without knowing that two words are synonyms of each other, they are considered two separate words during TER alignment.

Our goal is to create equivalence classes for each word in the given translations and modify the alignment algorithm to give priority to the matching of words that are in the same equivalence class. In this paper, the equivalence classes are generated using WordNet by extracting synonyms of each word in the translations.

To incorporate matching of word synonyms into the alignment, we followed three steps:

1. Use WordNet to extract synonyms of the words that appear in all hypotheses.
2. Augment each skeleton word with all synonymous words that appear in all the hypotheses.
3. Modify TER script to handle words with alternatives using an additional *synonym matching* operation.

In the following subsections, we describe how each of these tasks is performed.

4.1 Extracting Synonyms from WordNet

The first step is to use WordNet to extract synonyms of each word that appears in all hypotheses. This is simply done using the publicly available WordNet processing tools to extract all synonyms of the given word. To allow matching words that have the same stem or variations of the same word with different part-of-the-speech (POS) tags, we extract all synonyms of the given word regardless of their POS tag in the given translation.[3]

In the example above, it is clear that the verbs *wait* and *expect* have the same meaning but TER is unable to align these two words to each other because of different word positions. Using WordNet, we get the following two entries for the words *waiting* and *expect*, indicating that two words can be considered as synonyms of each other:

`expect`: Sense 1: *expect, anticipate* ⇒ *evaluate, pass judgment, judge*. Sense 2: *ask, require, expect* ⇒ *demand*. Sense 3: *expect, look, await,*

[2] Note that the approach by Matusov et al. (2006) attempts to align synonyms and different morphological forms of words to each other but this is done implicitly, relying on the parallel text to learn word alignments.

[3] Our goal is to add as many synonyms as possible to increase the chances of a word aligning to one of its possible synonyms rather than to any other word. Therefore, we do not distinguish between the synonyms of the same word according to their confidence value or their POS tag.

```
ONE-PASS ALIGNMENT
khaddam                                 receives                 italian         house speaker
khaddam                                 receives the head of the italian chamber of    deputies
khaddam will meet with the president of the                      italian chamber of    deputies

TWO-PASS ALIGNMENT
khaddam       receives                             italian  house           speaker
khaddam       receives                 of the head italian  chamber  of the deputies
khaddam will  meet     with  president of the      italian  chamber  of the deputies
```

Figure 2: Example alignment of three hypotheses with one-pass and two-pass alignment

wait. Sense 4: *expect ⇒ see, consider, reckon, view, regard.* Sense 5: *expect ⇒ expect, look, await,* **wait**. Sense 6: *have a bun in the oven, bear, carry, gestate, expect ⇒ give birth, deliver, bear, birth, have.*

`waiting`: Sense 1: *wait.* Sense 2: *wait, hold off, hold back ⇒ act, move.* Sense 3: *wait,* **expect**, *look, await.* Sense 4: *wait, waitress ⇒ work.*

We should note that WordNet contains only *open-class words*, i.e., nouns, verbs, adjectives, and adverbs. There are no entries for determiners, prepositions, pronouns, conjunctions, and particles. For better matching of these additional POS tags, we manually created a different equivalence class for each POS tag that is not included in the WordNet so that words with the same POS tag can be considered synonymous.

4.2 Augmenting References

After extracting the synonyms of each word in the given translations, the next step is to augment each reference word with its synonyms. To avoid overgeneration of synonyms, we make the assumption that words w_i and w_j are synonyms of each other only if w_i appears in the synonym list of w_j, and w_j appears in the synonym list of w_i. To make the alignment task more efficient and faster, we restrict the synonym list to only words that appear in the given translations. In our running example, the augmented (extended) skeleton according to the second hypothesis is as follows:

Extended skeleton: at the same time_*meantime* expect_*waiting* israel to abide by the deadlines set_*established* by .

4.3 Modifications to TER Script

The final step is to modify TER script to favor matching of a word to its synonyms rather than to any other word. To achieve this goal, we modified the publicly available TER script, TERCOM (Snover et al., 2006), to match words in the same equivalence class at an additional *synonym cost*. In its original implementation, TERCOM builds a hash table for the n-grams that appear in both the reference and the hypothesis translation to determine possible shifts of words. To allow synonymous words to be shifted and aligned to each other, we extend the hash table for all possible synonyms of words in the skeleton. Formally, if the skeleton includes two consecutive words w_i_s_i and w_j_s_j, where s_i (s_j) is a synonym of w_i (w_j), we put all four possible combinations to the hash table: $w_i w_j$, $w_i s_j$, $s_i w_j$, and $s_i s_j$.[4]

To give higher priority to the exact matching of words (which has zero cost during edit distance computation), we used a slightly higher cost for synonym matching, a cost of 0.1.[5] All the other operations (i.e., insertion, deletion, substitution and shifting of words) have a cost of 1.0.

4.4 Two-pass Alignment Strategy

When building a confusion network, the usual strategy is first to align each hypothesis to the skeleton separately and reorder them so that the word ordering in the given hypothesis matches the word ordering in the skeleton translation. Next a confusion network is built between all these reordered hypotheses.

One of the major problems with this process occurs when the hypotheses include additional words that do not appear in the skeleton translation, as depicted in Figure 2. Since the alignments of two different hypotheses are done independently, two hypotheses other than the skeleton may not align perfectly, especially when the additional words appear in different positions.

To overcome this issue, we employ a two-pass alignment strategy. In the first pass, we align all hypotheses to the skeleton independently and build a confusion network. Next an intermediate reference sentence is created from the confusion network generated in the first pass. To create this intermediate reference, we find the best position for each word that appears in the confusion network

[4] Note that the hash table is built in an iterative fashion. We consider adding a new n-gram only if the previous $n-1$ words appear in the hypothesis as well.

[5] Synonym matching cost was determined empirically, trying different costs from 0 to 0.5.

```
WITHOUT SYNONYM MATCHING and ONE-PASS ALIGNMENT:
at the          same time                                expect israel to abide  by
at the          same time    we                          expect israel to abide  by
at the          same time    ,   we      are waiting for    israel to abide  by
at the          same time    we                          expect israel to abide  by
at the          same time    ,   we                      expect israel to abide  by
at the          same time    ,   waiting                 for    israel to comply with
in the meantime ,    waiting                             for    israel to abide  by

WITH SYNONYM MATCHING and TWO-PASS ALIGNMENT:
at the same  time                 expect       israel to abide  by
at the same  time        we       expect       israel to abide  by
at the same  time     ,  we  are  waiting for  israel to abide  by
at the same  time        we       expect       israel to abide  by
at the same  time     ,  we       expect       israel to abide  by
at the same  time     ,           waiting for  israel to comply with
in the       meantime ,           waiting for  israel to abide  by
```

Figure 3: Example alignment via confusion networks with and without synonym matching and two-pass alignment (using the first sentence as the skeleton)

using majority voting. The second pass uses this intermediate reference as the skeleton translation to generate the final confusion network.

When we create the intermediate reference, the number of positions for a given word is bounded by the maximum number of occurrences of the same word in any hypothesis. It is possible that two different words are mapped to the same position in the intermediate reference. If this is the case, these words are treated as synonyms when building the second confusion network, and the intermediate reference looks like the extended reference in Section 4.2.

Finally, Figure 3 presents our running example with the old alignments versus the alignments with synonym matching and two-pass alignment.

4.5 Features

Each word in the confusion network is represented by system-specific word scores. For computing scores, each hypothesis is assigned a score based on three different methods:

1. **Uniform weighting**: Each hypothesis in the n-best list has the same score of $1/n$.
2. **Rank-based weighting**: Each hypothesis is assigned a score of $1/(1+r)$, where r is the rank of the hypothesis.
3. **TM-based weighting:** Each hypothesis is weighted by the score that is assigned to the hypothesis by the translation model.

The total score of an arc with word w for a given system S is the sum of all the scores of the hypotheses in system S that contain the word w in the given position. The score for a specific arc between nodes n_i and n_j is normalized by the sum of the scores for all the arcs between n_i and n_j.

Our experiments demonstrated that rank-based weighting performs the best among all three weighting methods although the differences are small. In the rest of the paper, we report only the results with rank-based weighting.

Besides the arc scores, we employ the following features during decoding:

1. Skeleton selection features for each system,
2. NULL-word (or epsilon) insertion score,
3. Word penalty, and
4. Language model score.

Skeleton selection feature is intended to help choose the best skeleton among the input systems. NULL-word feature controls the number of epsilon arcs used in the chosen translation during the decoding and word penalty feature controls the length of the translation. For language model scores, we used a 4-gram LM that we used to train the input systems.

5 Evaluation and Results

In this section, we describe how we train the input systems and how we evaluate the proposed system combination method.

5.1 Systems and Data

To evaluate the impact of the new alignments, we tested our system combination approach using the old alignments and improved alignments on two language pairs: Arabic-English and Chinese-English. We ran the system combination on three system outputs that were generated by an in-house hierarchical phrase-based decoder, as in (Chiang, 2007). The major difference between the three systems is that they were trained on different subsets of the available training data using different word alignments.

For generating the system outputs, first a hierarchical phrase-based decoder was used to generate three sets of unique 3000-best lists. Nine fea-

Data for Training/Tuning/Testing	Arabic-English		Chinese-English	
	# of segments	# of tokens	# of segments	# of tokens
Training Data (System1)	14.8M	170M	9.1M	207M
Training Data (System2)	618K	8.1M	13.4M	199M
Training Data (System3)	2.4M	27.5M	13.9M	208M
Tuning Set (Input Systems)	1800	51K	1800	51K
Tuning Set (System Combination)	1259	37K	1785	55K
Test Set - NIST MTEval'05	1056	32K	1082	32K
Test Set - NIST MTEval'06	1797	45K	1664	41K
Test Set - NIST MTEval'08	1360	43K	1357	34K

Table 1: Number of segments and source-side words in the training and test data.

tures were used in the hierarchical phrase-based systems under the log-linear model framework: a 4-gram language model (LM) score (trained on nearly 3.6 billion words using the SRILM toolkit (Stolcke, 2002)), conditional rule/phrase probabilities and lexical weights (in both directions), rule penalty, phrase penalty, and word penalty. Rules and phrases were extracted in a similar manner as described in (Chiang, 2007) from the training data with word alignments generated by GIZA++. The n-best lists were then re-scored with three additional LMs: a count-based LM built from the Google Tera word corpus, an almost parsing LM based on super-ARV tagging, and an approximated full-parser LM (Wang et al., 2007).

For Arabic-English, the first system was trained on all available training data (see Table 1 for details), with long sentences segmented into multiple segments based on IBM model 1 probabilities (Xu et al., 2005). The second system was trained on a small subset of the training data, which is mostly newswire. The third system was trained on an automatically extracted subset of the training data according to n-gram overlap in the test sets.

For Chinese-English, the first system used all the training data without any sentence segmentation. The second system used all training data after IBM-1 based sentence segmentation, with different weightings on different corpora. The third system is the same as the second system except that it used different word alignment symmetrization heuristics (grow-diag-final-and vs. grow-diag-final (Koehn et al., 2003)).

5.2 Empirical Results

All input systems were optimized on a randomly selected subset of the NIST MTEval'02, MTEval'03, and MTEval'04 test sets using minimum error rate training (MERT) (Och, 2003) to maximize BLEU score (Papineni et al., 2002). System combination was optimized on the rest of this data using MERT to maximize BLEU score. As inputs to the system combination, we used 10-best hypotheses from each of the re-ranked n-best lists. To optimize system combination, we generated unique 1000-best lists from a lattice we created from the input hypotheses, and used MERT in a similar way to MT system optimization.

System	MT'05	MT'06	MT'08
System 1	53.4	43.8	43.2
System 2	53.9	46.0	42.8
System 3	56.1	45.3	43.3
No Syns, 1-pass	56.7	47.5	44.9
w/Syns, 2-pass	57.9	48.4	46.2

Table 2: Lowercase BLEU scores (in percentages) on Arabic NIST MTEval test sets.

We evaluated system combination with improved alignments on three different NIST MTEval test sets (MTEval'05, MTEval'06 NIST portion, and MTEval'08). The final MT outputs were evaluated using lowercased BLEU scores.[6]

Tables 2 and 3 present the BLEU scores (in percentages) for the input systems and for different combination strategies on three test sets in Arabic-English and Chinese-English, respectively.

On Arabic-English, the combination with synonym matching and two-pass alignment yields absolute improvements of 1.8 to 2.9 BLEU point on three test sets over the best input system. When compared to the combination algorithm with the old alignments (i.e., 1-pass alignment with no synonym matching), the improved alignments yield an additional improvement of 0.9 to 1.3 BLEU point on the three test sets.

For Chinese-English, the improvements over the previous combination algorithm are smaller. The

[6] We used the NIST script (version 11b) for BLEU with its default settings: case-insensitive matching of up to 4-grams, and the shortest reference sentence for the brevity penalty.

System	MT’05	MT’06	MT’08
System 1	35.8	34.3	27.6
System 2	35.9	34.2	27.8
System 3	36.0	34.3	27.8
No Syns, 1-pass	38.1	36.5	27.9
w/Syns, 2-pass	38.6	37.0	28.3
No Syns, 1-pass, tuning set w/webtext			28.4
w/Syns, 2-pass, tuning set w/webtext			29.3

Table 3: Lowercase BLEU scores (in percentages) on Chinese NIST MTEval test sets.

new combination system yields up to an absolute 2.6 BLEU point improvement over the best input system and up to 0.5 BLEU point improvement over the previous combination algorithm on three different test sets. Note that for Arabic-English, the individual systems show a high variance in translation quality when compared to Chinese-English systems. This might explain why the improvements on Chinese-English are modest when compared to Arabic-English results.

We also noticed that system combination yielded much smaller improvement on Chinese MTEval’08 data when compared to other test sets, regardless of the alignment method (only 0.5 BLEU point over the best input system). We suspected that this might happen because of a mismatch between the genres of the test set and the tuning set (the amount of web text data in MTEval’08 test set is high although the tuning set does not include any web text data). To test this hypothesis, we created a new tuning set for system combination, which consists of 2000 randomly selected sentences from the previous MTEval test sets and includes web text data. Using this new tuning set, combination with the improved alignments yields a BLEU score of 29.3 on MTEval’08 data (an absolute improvement of 1.5 BLEU point over the best input system, and 0.9 BLEU point improvement over the combination with the old alignments). These new results again validate the usefulness of the improved alignments when the tuning set matches the genre of the test set.

5.3 A Comparison of the Impact of Synonym Matching and Two-pass Alignment

One last evaluation investigated the impact of each component on the overall improvement. For this purpose, we ran system combination by turning on and off each component. Table 4 presents the system combination results in terms of BLEU scores on Arabic-English test sets when each component is used on its own or when they are used together.

Synon.	2-pass	MT’05	MT’06	MT’08
No	No	56.7	47.5	44.9
Yes	No	57.3	47.8	45.2
No	Yes	57.7	48.0	45.9
Yes	Yes	57.9	48.4	46.2

Table 4: Comparison of Synonym Matching and Two-pass Alignment on Arabic-English

The results indicate that synonym matching on its own yields improvements of 0.3-0.6 BLEU points over not using synonym matching. Two-pass alignment turns out to be more useful than synonym matching, yielding an absolute improvement of up to 1 BLEU point over one-pass alignment.

6 Conclusions

We presented an extension to the previous alignment approaches to handle word synonyms more effectively in an attempt to improve the alignments between different hypotheses during confusion network decoding. We also presented a two-pass alignment strategy for a better alignment of hypotheses with similar words but with a different word ordering.

We evaluated our system combination approach on two language pairs: Arabic-English and Chinese-English. Combination with improved alignments yielded up to an absolute 2.9 BLEU point improvement over the best input system and up to an absolute 1.3 BLEU point improvement over combination with the old alignments. It is worth noting that these improvements are obtained using very similar input systems. We expect that the improvements will be higher when we use structurally different MT systems as inputs to the combiner.

Our future work includes a more effective use of existing linguistic resources to handle alignment of one word to multiple words (e.g., *al-nahayan* vs. *al nahyan*, and *threaten* vs. *pose threat*) and matching of similar (but not necessarily synonymous) words (polls vs. elections). We are also planning to extend word lattices to include phrases from the individual systems (i.e., not just the words) for more grammatical outputs.

Acknowledgments This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

References

Akiba, Yasuhiro, Taro Watanabe, and Eiichiro Sumita. 2002. Using language and translation models to select the best among outputs from multiple MT systems. In *Proc. of the 19th Intl. Conf. on Computational Linguistics (COLING'2002)*.

Bangalore, Srinivas, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'2001)*.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Fellbaum, Christiane. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books, March. Available at http://wordnet.princeton.edu.

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

Frederking, Robert and Sergei Nirenburg. 1994. Three heads are better than one. In *Proc. of the 4th Conf. on Applied Natural Language Processing (ANLP'1994)*.

Huang, Fei and Kishore Papineni. 2007. Hierarchical system combination for machine translation. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP'2007)*.

Jayaraman, Shyamsundar and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT'2005)*.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Human Language Technology and the Meeting of the North American Chapter of the Association for Computational Linguistics Conf. (HLT/NAACL'2003)*.

Matusov, Evgeny, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. of the 11th Conf. of the European Chapter of the Association for Computational Linguistics (EACL'2006)*.

Mellebeek, Bart, Karolina Owczarzak, Josef Van Genabith, and Andy Way. 2006. Multi-engine machine translation by recursive sentence decomposition. In *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA'2006)*.

Nomoto, Tadashi. 2004. Multi-engine machine translation with voted language model. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*.

Och, Franz J. and Hermann Ney. 2000. Improved statistical alignment models. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'2000)*.

Och, Franz J. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'2003)*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'2002)*.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007a. Combining outputs from multiple machine translation systems. In *Proc. of the Human Language Technology and the Meeting of the North American Chapter of the Association for Computational Linguistics Conf. (HLT/NAACL'2007)*.

Rosti, Antti-Veikko, Spyros Matsoukas, and Richard Schwartz. 2007b. Improved word-level system combination for machine translation. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'2007)*.

Sim, Khe Chai, William J. Byrne, Mark J.F. Gales, Hichem Sahbi, and Phil C. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *Proc. of the 32nd Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'2007)*.

Snover, Matthew, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA'2006)*.

Stolcke, Andreas. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of the Intl. Conf. on Spoken Language Processing (ICSLP'2002)*.

Wang, Wen, Andreas Stolcke, and Jing Zheng. 2007. Reranking machine translation hypotheses with structured and web-based language models. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'2007)*.

Xu, Jia, Richard Zens, and Hermann Ney. 2005. Sentence segmentation using IBM word alignment model 1. In *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT'2005)*.