Combining Outputs from Multiple Machine Translation Systems

Antti-Veikko I. Rosti[a] and **Necip Fazil Ayan**[b] and **Bing Xiang**[a] and
Spyros Matsoukas[a] and **Richard Schwartz**[a] and **Bonnie J. Dorr**[b]
[a]BBN Technologies, 10 Moulton Street, Cambridge, MA 02138
{arosti,bxiang,smatsouk,schwartz}@bbn.com
[b]Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742
{nfa,bonnie}@umiacs.umd.edu

Abstract

Currently there are several approaches to machine translation (MT) based on different paradigms; e.g., phrasal, hierarchical and syntax-based. These three approaches yield similar translation accuracy despite using fairly different levels of linguistic knowledge. The availability of such a variety of systems has led to a growing interest toward finding better translations by combining outputs from multiple systems. This paper describes three different approaches to MT system combination. These combination methods operate on sentence, phrase and word level exploiting information from N-best lists, system scores and target-to-source phrase alignments. The word-level combination provides the most robust gains but the best results on the development test sets (NIST MT05 and the newsgroup portion of GALE 2006 dry-run) were achieved by combining all three methods.

1 Introduction

In recent years, machine translation systems based on new paradigms have emerged. These systems employ more than just the surface-level information used by the state-of-the-art phrase-based translation systems. For example, hierarchical (Chiang, 2005) and syntax-based (Galley et al., 2006) systems have recently improved in both accuracy and scalability. Combined with the latest advances in phrase-based translation systems, it has become more attractive to take advantage of the various outputs in forming consensus translations (Frederking and Nirenburg, 1994; Bangalore et al., 2001; Jayaraman and Lavie, 2005; Matusov et al., 2006).

System combination has been successfully applied in state-of-the-art speech recognition evaluation systems for several years (Fiscus, 1997). Even though the underlying modeling techniques are similar, many systems produce very different outputs with approximately the same accuracy. One of the most successful approaches is consensus network decoding (Mangu et al., 2000) which assumes that the confidence of a word in a certain position is based on the sum of confidences from each system output having the word in that position. This requires aligning the system outputs to form a consensus network and – during decoding – simply finding the highest scoring path through this network. The alignment of speech recognition outputs is fairly straightforward due to the strict constraint in word order. However, machine translation outputs do not have this constraint as the word order may be different between the source and target languages. MT systems employ various re-ordering (distortion) models to take this into account.

Three MT system combination methods are presented in this paper. They operate on the sentence, phrase and word level. The sentence-level combination is based on selecting the best hypothesis out of the merged N-best lists. This method does not generate new hypotheses – unlike the phrase and word-level methods. The phrase-level combination

Proceedings of NAACL HLT 2007, pages 228–235,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

is based on extracting sentence-specific phrase translation tables from system outputs with alignments to source and running a phrasal decoder with this new translation table. This approach is similar to the multi-engine MT framework proposed in (Frederking and Nirenburg, 1994) which is not capable of re-ordering. The word-level combination is based on consensus network decoding. Translation edit rate (TER) (Snover et al., 2006) is used to align the hypotheses and minimum Bayes risk decoding under TER (Sim et al., 2007) is used to select the alignment hypothesis. All combination methods use weights which may be tuned using Powell's method (Brent, 1973) on N-best lists. Both sentence and phrase-level combination methods can generate N-best lists which may also be used as new system outputs in the word-level combination.

Experiments on combining six machine translation system outputs were performed. Three systems were phrasal, two hierarchical and one syntax-based. The systems were evaluated on NIST MT05 and the newsgroup portion of the GALE 2006 dry-run sets. The outputs were evaluated on both TER and BLEU. As the target evaluation metric in the GALE program was human-mediated TER (HTER) (Snover et al., 2006), it was found important to improve both of these automatic metrics.

This paper is organized as follows. Section 2 describes the evaluation metrics and a generic discriminative optimization technique used in tuning of the various system combination weights. Sentence, phrase and word-level system combination methods are presented in Sections 3, 4 and 5. Experimental results on Arabic and Chinese to English newswire and newsgroup test data are presented in Section 6.

2 Evaluation Metrics and Discriminative Tuning

The official metric of the 2006 DARPA GALE evaluation was human-mediated translation edit rate (HTER). HTER is computed as the minimum translation edit rate (TER) between a system output and a targeted reference which preserves the meaning and fluency of the sentence (Snover et al., 2006). The targeted reference is generated by human post-editors who make edits to a reference translation so as to minimize the TER between the reference and the MT output without changing the meaning of the reference. Computing the HTER is very time consuming due to the human post-editing. It is desirable to have an automatic evaluation metric that correlates well with the HTER to allow fast evaluation of the MT systems during development. Correlations of different evaluation metrics have been studied (Snover et al., 2006) but according to various internal HTER experiments it is not clear whether TER or BLEU correlates better. Therefore it is probably safest to try and not degrade either.

The TER of a translation E is computed as

$$\mathrm{TER}(E, E_r) = \frac{\mathrm{Ins} + \mathrm{Del} + \mathrm{Sub} + \mathrm{Shft}}{N_r} \times 100\% \tag{1}$$

where N_r is the total number of words in the reference translation E_r. In the case of multiple references, the edits are counted against all references, N_r is the average number of words in the reference translations and the final TER is computed using the minimum number of edits. The NIST BLEU-4 is a variant of BLEU (Papineni et al., 2002) and is computed as

$$\mathrm{BLEU}(E, E_r) = \exp\Big(\frac{1}{4}\sum_{n=1}^{4}\log p_n(E, E_r)\Big)\gamma(E, E_r) \tag{2}$$

where $p_n(E, E_r)$ is the precision of n-grams in the hypothesis E given the reference E_r and $\gamma(E, E_r) \leq 1$ is a brevity penalty. The n-gram counts from multiple references are accumulated in estimating the precisions.

All system combination methods presented in this paper may be tuned to directly optimize either one of these automatic evaluation metrics. The tuning uses N-best lists of hypotheses with various feature scores. The feature scores may be combined with tunable weights forming an arbitrary scoring function. As the derivatives of this function are not usually available, Brent's modification of Powell's method (Brent, 1973) may be used to find weights that optimize the appropriate evaluation metric in the re-scored N-best list. The optimization starts at a random initial point in the p-dimensional parameter space, first searching through an initial set of basis vectors. As searching repeatedly through the set of basis vectors is inefficient, the direction of the vectors is gradually moved toward a larger positive change in the evaluation metric. To improve the chances of finding a global optimum, the algorithm is repeated with varying initial values. The modified Powell's method has been previously used in optimizing the weights of a standard feature-based MT decoder in (Och, 2003) where a more efficient algorithm for log-linear models was proposed. However, this is specific to log-linear models and cannot be easily extended for more complicated functions.

3 Sentence-Level Combination

The first combination method is based on re-ranking a merged N-best list. A confidence score from each system is assigned to each unique hypothesis in the merged list. The confidence scores for each hypothesis are used to produce a single score which, combined with a 5-gram language model score, determines a new ranking of the hypotheses.

3.1 Hypothesis Confidence Estimation

Generalized linear models (GLMs) have been applied for confidence estimation in speech recognition (Siu and Gish, 1999). The logit model, which models the log odds of an event as a linear function of the features, can be used in confidence estimation. The confidence P_{ij} for a system i generating a hypothesis j may be modeled as

$$\log \frac{P_{ij}}{1 - P_{ij}} = \sum_{l=1}^{L} w_{il} x_{ijl} \tag{3}$$

where each system has L weights w_{il}, and x_{ijl} is the lth feature for system i and hypothesis j. The features used in this work were:

1. Rank in the system's N-best list;
2. Sentence posterior with system-specific total score scaling factors;
3. System's total score;
4. Number of words in the hypothesis;
5. System-specific bias.

If the system i did not generate the hypothesis j, the confidence is set to zero. To prevent overflow in exponentiating the summation in Equation 3, the features have to be scaled. In the experiments, feature scaling factors were estimated from the tuning data to limit the feature values between $[0, 1]$. The same scaling factors have to be applied to the features obtained from the test data.

The total confidence score of hypothesis j is obtained from the system confidences P_{ij} as

$$P_j = \alpha \frac{N_j}{N_s} + \beta \frac{1}{N_s} \sum_{i=1}^{N_s} P_{ij} + \gamma \max_i P_{ij} + \delta \frac{1}{N_j} \sum_{i=1}^{N_s} P_{ij} \tag{4}$$

where N_j is the number of systems generating the hypothesis j (i.e., the number of non-zero P_{ij} for j) and N_s is the number of systems. The weights α through δ are constrained to sum to one; i.e., there are three free parameters. These weights can balance the total confidence between the number of systems generating the hypothesis (votes), and the sum, maximum and average of the system confidences.

3.2 Sentence Posterior Estimation

The second feature in the GLM is the sentence posterior estimated from the N-best list. A sentence posterior may simply be estimated from an N-best list by scaling the system scores for all hypotheses to sum to one. When combining several systems based on different translation paradigms and feature sets, the system scores may not be comparable. The total scores may be scaled to obtain more consistent sentence posteriors. The scaled posterior estimated from an N-best list may be written as

$$P_{ij} = \exp\Big(s_i L_{ij} - \log\Big(\sum_{k=1}^{N} \exp(s_i L_{ik})\Big)\Big) \tag{5}$$

where s_i is the scaling factor for system i and L_{ij} is the log-score system i assigns to hypothesis j. The scaling factors may be tuned to optimize the evaluation metric in the same fashion as the logit model weights in Section 3.1. Equation 4 may be used to assign total posteriors for each unique hypothesis and the weights may be tuned using Powell's method on N-best lists as described in Section 2.

3.3 Hypothesis Re-ranking

The hypothesis confidence may be log-linearly combined with a 5-gram language model (LM) score to yield the final score as follows

$$L_j = \log P_j + \nu L_j^{5gr} + \mu W_j \tag{6}$$

where W_j is the number of words in hypothesis j. The number of words is commonly used in LM re-scoring to balance the LM scores between hypotheses of different lengths. The number of free parameters in the sentence-level combination method is given by $N_s + N_s L + 8$ where N_s is the number of systems and L is the number of features; i.e., N_s system score scaling factors (s_i), three free interpolation weights (Equation 4) for the scaling factor estimation, $N_s L$ GLM weights (w_{il}), three free interpolation weights (Equation 4) for the hypothesis confidence estimation and two free LM re-scoring weights (Equation 6). All parameters may be tuned using Powell's method on N-best lists as described in Section 2.

The tuning of the sentence-level combination method may be summarized as follows:

1. Merge individual N-best lists to form a large N-best list with unique hypotheses;
2. Estimate total score scaling factors as described in Section 3.2;
3. Collect GLM feature scores for each unique hypothesis;
4. Estimate GLM feature scaling factors as described in Section 3.1;
5. Scale the GLM features;
6. Estimate GLM weights, combination weights and LM re-scoring weights as described above;
7. Re-rank the merged N-best list using the new weights.

Testing the sentence-level combination has the same steps as the tuning apart from all estimation steps; i.e., steps 1, 3, 5 and 7.

4 Phrase-Level Combination

The phrase-level combination is based on extracting a new phrase translation table from each system's target-to-source phrase alignments and re-decoding the source sentence using this new translation table and a language model. In this work, the target-to-source phrase alignments were available from the individual systems. If the alignments are not available, they can be automatically generated; e.g., using GIZA++ (Och and Ney, 2003). The phrase translation table is generated for each source sentence using confidence scores derived from sentence posteriors with system-specific total score scaling factors and similarity scores based on the agreement among the phrases from all systems.

4.1 Phrase Confidence Estimation

Each phrase has an initial confidence based on the sentence posterior P_{ij} estimated from an N-best list in the same fashion as in Section 3.2. The confidence of the phrase table entry is increased if several systems agree on the target words. The agreement is measured by four levels of similarity:

1. Same source interval, same target words, and same original distortion;
2. Same source interval, same target words, with different original distortion;
3. Overlapping source intervals with the same target words;
4. Overlapping target words.

S_{ilm} represents the similarity of a given phrase m to all the hypotheses in the system i at the similarity level l. Basically, if there is a similar phrase in a given hypothesis j in the system i to the phrase m, the similarity score S_{ilm} is increased by P_{ij}. Note that each phrase in one hypothesis is similar to another hypothesis at only one similarity level, so one hypothesis can contribute to S_{ilm} at only one similarity level. The final confidence of the phrase table entry is defined as

$$\begin{aligned} P_m = \log\Big(&\alpha \sum_{i,l} w_i v_l S_{ilm} \\ &+\beta \frac{1}{\sum_{i,l:S_{ilm}\neq 0} w_i v_l} \sum_{i,l:S_{ilm}\neq 0} w_i v_l S_{ilm} \\ &+\gamma \max_i \sum_l w_i v_l S_{ilm}\Big) \end{aligned} \tag{7}$$

where w_i are system weights and v_l are similarity score weights. The parameters α through γ interpolate between the sum, average and maximum of the similarity scores. These interpolation weights and the system weights w_i are constrained to sum to one. The number of tunable combination weights, in addition to normal decoder weights, is $N_s + N_l + 1$ where N_s is the number of systems and N_l is the number of similarity levels; i.e., $N_s - 1$ free system weights, N_l similarity score weights and two free interpolation weights.

4.2 Phrase-Based Decoding

The phrasal decoder used in the phrase-level combination is based on standard beam search (Koehn, 2004). The decoder features are: a trigram language model score, number of target phrases, number of target words, phrase distortion, phrase distortion computed over the original translations and phrase translation confidences estimated in Section 4.1. The total score for a hypothesis is computed as a log-linear combination of these features. The feature weights and combination weights (system and similarity) may be tuned using Powell's method on N-best lists as described in Section 2.

The phrase-level combination tuning can be summarized as follows:

1. Estimate sentence posteriors given the total score scaling factors;
2. Collect all M unique phrase table entries from each hypothesis accumulating the similarity scores S_{ilm};
3. Combine the similarity scores to form phrase confidences according to Equation 7;
4. Decode the source sentences using the current weights to generate an N-best list;
5. Estimate new decoder and combination weights as described above.

Testing the phrase-level combination is performed by following steps 1 through 4.

5 Word-Level Combination

The third combination method is based on confusion network decoding. In confusion network decoding, the words in all hypotheses are aligned against each other to form a graph with word alternatives (including nulls) for each alignment position. Each aligned word is assigned a score relative to the votes or word confidence scores (Fiscus, 1997; Mangu et al., 2000) derived from the hypotheses. The decoding is carried out by picking the words with the highest scores along the graph. In speech recognition, this results in minimum expected word error rate (WER) hypothesis (Mangu et al., 2000) or equivalently minimum Bayes risk (MBR) under WER with uniform target sentence posterior distribution (Sim et al., 2007).

In machine translation, aligning hypotheses is more complicated compared to speech recognition since the target words do not necessarily appear in the same order. So far, confusion networks have been applied in MT system combination using three different alignment procedures: WER (Bangalore et al., 2001), GIZA++ alignments (Matusov et al., 2006) and TER (Sim et al., 2007). WER alignments do not allow shifts, GIZA++ alignments require careful training and are not always reliable. TER alignments do not guarantee that similar but lexically different words are aligned correctly but TER does not require training new models and allows shifts (Snover et al., 2006). This work extends the approach proposed in (Sim et al., 2007).

5.1 Confusion Network Generation

Due to the varying word order in the MT hypotheses, the decision of confusion network *skeleton* is essential. The skeleton determines the general word order of the combined hypothesis. One option would be to use the output from the system with the best performance on some development set. However, it was found that this approach did not always yield better combination output compared to the best single system on all evaluation metrics. Instead of using a single system output as the skeleton, the hypothesis that best agrees with the other hypotheses on average may be used. In this paper, the minimum average TER score of one hypothesis against all other hypotheses was used as follows

$$E_r = \arg\min_i \sum_{j=1}^{N_s} \mathrm{TER}(E_j, E_i) \tag{8}$$

This may be viewed as the MBR hypothesis under TER given uniform target sentence posterior distribution (Sim et al., 2007). It is also possible to compute the MBR hypothesis under BLEU.

Finding the MBR hypothesis requires computing the TER against all hypotheses to be aligned. It was found that aligning more than one hypothesis ($N = 10$) from each system to the skeleton improves the combination outputs. However, only the rank-1 hypotheses were considered as skeletons due to the complexity of the TER alignment. The confidence score assigned to each word was chosen to be $1/(1 + rank)$ where the $rank$ was based on the rank of the aligned hypothesis in the system's N-best. This was found to yield better scores than simple votes.

5.2 Tunable System Weights

The word-level combination method described so far does not require any tuning. To allow a variety of outputs with different degrees of confidence to be combined, system weights may be used. A confusion network may be represented as a standard word lattice with all paths traveling via all nodes. The links in this lattice represent the alternative words (including nulls) at the corresponding position in the string. Confusion network decoding may be viewed as finding the highest scoring path through this lattice with summing all word scores along the path. The standard lattice decoding algorithms may also be used to generate N-best lists from the confusion network. The simplest way to introduce system weights is to accumulate system-specific scores along the paths and combine these scores linearly with the weights. The system weights may be tuned using Powell's method on N-best lists as described in Section 2.

The word-level combination tuning can be summarized as follows:

1. Extract 10-best lists from the MT outputs;
2. Align each 10-best against each rank-1 hypothesis using TER;
3. Choose the skeleton (Equation 8);
4. Generate a confusion network lattice with the current system weights;
5. Generate N-best list hypothesis and score files from the lattice;
6. Estimate system weights as described above;

Arabic	Newswire		Newsgroups	
	TER	BLEU	TER	BLEU
system A	42.98	49.58	59.73	20.36
system B	43.79	47.06	61.55	18.08
system C	43.92	47.87	60.81	18.08
system D	40.75	52.09	59.25	20.28
system E	42.19	50.86	59.85	19.73
system F	44.30	50.15	61.74	20.61
phrcomb	40.45	53.70	59.90	21.49
sentcomb	41.56	52.18	60.21	19.77
no weights 6	39.33	53.66	58.15	20.61
TER 6	39.41	54.37	58.21	20.85
TER 8	39.43	54.40	57.96	21.44

Table 1: Mixed-case TER and BLEU scores on Arabic NIST MT05 (newswire) and the newsgroups portion of the GALE 2006 dry-run data.

7. Re-rank the N-best list using the new weights.

Testing the word-level combination has the same steps as the tuning apart from steps 6 and 7.

6 Experiments

Six systems trained on all data available for GALE 2006 evaluation were used in the experiments to demonstrate the performance of all three system combination methods on Arabic and Chinese to English MT tasks. Three systems were phrase-based (A, C and E), two hierarchical (B and D) and one syntax-based (F). The phrase-based systems used different sets of features and re-ordering approaches. The hierarchical systems used different rule sets. All systems were tuned on NIST MT02 evaluation sets with four references. Systems A and B were tuned to minimize TER, the other systems were tuned to maximize BLEU.

As discussed in Section 2, the system combination tuning metric was chosen so that gains were observed in both TER and BLEU on development test sets. NIST MT05 comprising only newswire data (1056 Arabic and 1082 Chinese sentences) with four reference translations and the newsgroup portion of the GALE 2006 dry-run (203 Arabic and 126 Chinese sentences) with one reference translation were used as the test sets. It was found that minimizing TER on Arabic also resulted in higher BLEU scores compared to the best single system. However,

Chinese	Newswire		Newsgroups	
	TER	BLEU	TER	BLEU
system A	56.57	29.63	68.61	13.20
system B	56.30	29.62	69.87	12.33
system C	59.48	31.32	69.37	13.91
system D	58.32	33.77	67.61	16.86
system E	58.46	32.40	69.08	15.08
system F	56.79	35.30	68.08	16.31
phrcomb	56.50	35.33	68.48	15.88
sentcomb	56.71	36.24	69.50	16.11
no weights 6	53.80	36.17	66.87	15.90
BLEU 6	54.34	36.44	66.50	16.44
BLEU 8	54.86	36.90	66.45	17.32

Table 2: Mixed-case TER and BLEU scores on Chinese NIST MT05 (newswire) and the newsgroups portion of the GALE 2006 dry-run data.

minimizing TER on Chinese resulted in significantly lower BLEU. So, TER was used in tuning the combination weights on Arabic and BLEU on Chinese.

The sentence and phrase-level combination weights were tuned on NIST MT03 evaluation sets. On the tuning sets, both methods yield about 0.5%-1.0% gain in TER and BLEU. The mixed-case TER and BLEU scores on both test sets are shown in Table 1 for Arabic and Table 2 for Chinese (`phrcomb` represents phrase and `sentcomb` sentence-level combination). The phrase-level combination seems to outperform the sentence-level combination in terms of both metrics on Arabic although gains over the best single system are modest, if any. On Chinese, the sentence-level combination yields higher BLEU scores than the phrase-level combination. The combination BLEU scores on the newsgroup data are not higher than the best system, though.

The word-level combination was evaluated in three settings. First, simple confusion network decoding with six systems without system weights was performed (`no weights 6` in the tables). Second, system weights were trained for combining six systems (`TER/BLEU 6` in the tables). Finally, all six system outputs as well as the sentence and phrase-level combination outputs were combined with system weights (`TER/BLEU 8` in the tables). The 6-way combination weights were tuned on merged NIST MT03 and MT04 evaluation sets and the 8-way combination weights were tuned only on NIST MT04 since the sentence and phrase-level combination methods were already tuned on NIST MT03. The word-level combination yields about 2.0%-3.0% gain in TER and 2.0%-4.0% gain in BLEU on the tuning sets. The test set results show that the simple confusion network decoding without system weights yields very good scores, mostly better than either sentence or phrase-level combination. The system weights seem to yield even higher BLEU scores but not always lower TER scores on both languages. Despite slightly hurting the TER score on Arabic, the `TER 8` combination result was considered the best due to the highest BLEU and significantly lower TER compared to any single system. Similarly, the `BLEU 8` was considered the best combination result on Chinese. Internal HTER experiments showed that `BLEU 8` yielded lower scores after post-editing even though `no weights 6` had lower automatic TER score.

7 Conclusions

Three methods for machine translation system combination were presented in this paper. The sentence-level combination was based on re-ranking a merged N-best list using generalized linear models with features derived from each system's output. The combination yields slight gains on the tuning set. However, the gains were very small, if any, on the test sets. The re-ranked N-best lists were used successfully in the word-level combination method as new system outputs. Various other features may be explored in this framework although the tuning may be limited by the chosen optimization method in the higher dimensional parameter space.

The phrase-level combination was based on deriving a new phrase translation table from the alignments to source provided in all system outputs. The phrase translation scores were based on the level of agreement between the system outputs and sentence posterior estimates. A standard phrasal decoder with the new phrase table was used to produce the final combination output. The handling of the alignments from non-phrasal decoders may not be optimal, though. The phrase-level combination yields fairly good gains on the tuning sets. However, the performance does not seem to generalize to the test sets used in this work. As usual, the phrasal decoder can generate N-best lists which were used successfully in the word-level combination method as new system outputs.

The word-level combination method based on consensus network decoding seems to be very robust and yield good gains over the best single system even without any tunable weights. The decision of the skeleton is crucial. Minimum Bayes Risk decoding under translation edit rate was used to select the skeleton. Compared to the best possible skeleton decision – according to an oracle experiment – further gains might be obtained by using better decision approach. Also, the alignment may be improved by taking the target-to-source alignments into account and allowing synonyms to align. The confusion network decoding at the word level does not necessarily retain coherent phrases as no language model constraints are taken into account. LM re-scoring might alleviate this problem.

This paper has provided evidence that outputs from six very different MT systems, tuned for two different evaluation metrics, may be combined to yield better outputs in terms of different evaluation metrics. The focus of the future work will be to address the individual issues in the combination methods mentioned above. It would also be interesting to investigate how much different systems contribute to the overall gain achieved via system combination.

Acknowledgments

This work was supported by DARPA/IPTO Contract No. HR0011-06-C-0022 under the GALE program (approved for public release, distribution unlimited). The authors would like to thank ISI and University of Edinburgh for sharing their MT system outputs.

References

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proc. ASRU*, pages 351–354.

Richard P. Brent. 1973. *Algorithms for Minimization Without Derivatives*. Prentice-Hall.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL*, pages 263–270.

Jonathan G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. ASRU*, pages 347–354.

Robert Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *Proc. ANLP*, pages 95–100.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inferences and training of context-rich syntax translation models. In *Proc. COLING/ACL*, pages 961–968.

Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. EAMT*.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proc. AMTA*.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. EACL*, pages 33–40.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

Khe Chai Sim, 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. ICASSP*.

Manhung Siu and Herbert Gish. 1999. Evaluation of word confidence for speech recognition systems. *Computer Speech and Language*, 13(4):299–319.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciula, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. AMTA*.