Improved Word-Level System Combination for Machine Translation

Antti-Veikko I. Rosti and **Spyros Matsoukas** and **Richard Schwartz**
BBN Technologies, 10 Moulton Street
Cambridge, MA 02138
{arosti,smatsouk,schwartz}@bbn.com

Abstract

Recently, confusion network decoding has been applied in machine translation system combination. Due to errors in the hypothesis alignment, decoding may result in ungrammatical combination outputs. This paper describes an improved confusion network based method to combine outputs from multiple MT systems. In this approach, arbitrary features may be added log-linearly into the objective function, thus allowing language model expansion and re-scoring. Also, a novel method to automatically select the hypothesis which other hypotheses are aligned against is proposed. A generic weight tuning algorithm may be used to optimize various automatic evaluation metrics including TER, BLEU and METEOR. The experiments using the 2005 Arabic to English and Chinese to English NIST MT evaluation tasks show significant improvements in BLEU scores compared to earlier confusion network decoding based methods.

1 Introduction

System combination has been shown to improve classification performance in various tasks. There are several approaches for combining classifiers. In ensemble learning, a collection of simple classifiers is used to yield better performance than any single classifier; for example boosting (Schapire, 1990). Another approach is to combine outputs from a few highly specialized classifiers. The classifiers may be based on the same basic modeling techniques but differ by, for example, alternative feature representations. Combination of speech recognition outputs is an example of this approach (Fiscus, 1997). In speech recognition, confusion network decoding (Mangu et al., 2000) has become widely used in system combination.

Unlike speech recognition, current statistical machine translation (MT) systems are based on various different paradigms; for example phrasal, hierarchical and syntax-based systems. The idea of combining outputs from different MT systems to produce consensus translations in the hope of generating better translations has been around for a while (Frederking and Nirenburg, 1994). Recently, confusion network decoding for MT system combination has been proposed (Bangalore et al., 2001). To generate confusion networks, hypotheses have to be aligned against each other. In (Bangalore et al., 2001), Levenshtein alignment was used to generate the network. As opposed to speech recognition, the word order between two correct MT outputs may be different and the Levenshtein alignment may not be able to align shifted words in the hypotheses. In (Matusov et al., 2006), different word orderings are taken into account by training alignment models by considering all hypothesis pairs as a parallel corpus using GIZA++ (Och and Ney, 2003). The size of the test set may influence the quality of these alignments. Thus, system outputs from development sets may have to be added to improve the GIZA++ alignments. A modified Levenshtein alignment allowing shifts as in computation of the translation edit rate (TER) (Snover et al., 2006) was used to align hy-

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 312–319,
Prague, Czech Republic, June 2007.

potheses in (Sim et al., 2007). The alignments from TER are consistent as they do not depend on the test set size. Also, a more heuristic alignment method has been proposed in a different system combination approach (Jayaraman and Lavie, 2005). A full comparison of different alignment methods would be difficult as many approaches require a significant amount of engineering.

Confusion networks are generated by choosing one hypothesis as the "skeleton", and other hypotheses are aligned against it. The skeleton defines the word order of the combination output. Minimum Bayes risk (MBR) was used to choose the skeleton in (Sim et al., 2007). The average TER score was computed between each system's 1-best hypothesis and all other hypotheses. The MBR hypothesis is the one with the minimum average TER and thus, may be viewed as the closest to all other hypotheses in terms of TER. This work was extended in (Rosti et al., 2007) by introducing system weights for word confidences. However, the system weights did not influence the skeleton selection, so a hypothesis from a system with zero weight might have been chosen as the skeleton. In this work, confusion networks are generated by using the 1-best output from each system as the skeleton, and prior probabilities for each network are estimated from the average TER scores between the skeleton and other hypotheses. All resulting confusion networks are connected in parallel into a joint lattice where the prior probabilities are also multiplied by the system weights.

The combination outputs from confusion network decoding may be ungrammatical due to alignment errors. Also the word-level decoding may break coherent phrases produced by the individual systems. In this work, log-posterior probabilities are estimated for each confusion network arc instead of using votes or simple word confidences. This allows a log-linear addition of arbitrary features such as language model (LM) scores. The LM scores should increase the total log-posterior of more grammatical hypotheses. Powell's method (Brent, 1973) is used to tune the system and feature weights simultaneously so as to optimize various automatic evaluation metrics on a development set. Tuning is fully automatic, as opposed to (Matusov et al., 2006) where global system weights were set manually.

This paper is organized as follows. Three evaluation metrics used in weights tuning and reporting the test set results are reviewed in Section 2. Section 3 describes confusion network decoding for MT system combination. The extensions to add features log-linearly and improve the skeleton selection are presented in Sections 4 and 5, respectively. Section 6 details the weights optimization algorithm and the experimental results are reported in Section 7. Conclusions and future work are discussed in Section 8.

2 Evaluation Metrics

Currently, the most widely used automatic MT evaluation metric is the NIST BLEU-4 (Papineni et al., 2002). It is computed as the geometric mean of n-gram precisions up to 4-grams between the hypothesis E and reference E_r as follows

$$\begin{aligned}\mathrm{BLEU}(E, E_r) = & \qquad\qquad (1)\\ \exp\Big(\frac{1}{4}\sum_{n=1}^{4}\log p_n(E, E_r)\Big)\gamma(E, E_r)\end{aligned}$$

where $\gamma(E, E_r) \le 1$ is the brevity penalty and $p_n(E, E_r)$ are the n-gram precisions. When multiple references are provided, the n-gram counts against all references are accumulated to compute the precisions. Similarly, full test set scores are obtained by accumulating counts over all hypothesis and reference pairs. The BLEU scores are between 0 and 1, higher being better. Often BLEU scores are reported as percentages and "one BLEU point gain" usually means a BLEU increase of 0.01.

Other evaluation metrics have been proposed to replace BLEU. It has been argued that METEOR correlates better with human judgment due to higher weight on recall than precision (Banerjee and Lavie, 2005). METEOR is based on the weighted harmonic mean of the precision and recall measured on unigram matches as follows

$$\mathrm{MTR}(E, E_r) = \frac{10m}{N_h + 9N_r}\Big(1 - 0.5(c/m)^3\Big) \quad (2)$$

where m is the total number of unigram matches, N_h is the hypothesis length, N_r is the reference length and c is the minimum number of n-gram matches that covers the alignment. The second term is a fragmentation penalty which penalizes the harmonic mean by a factor of up to 0.5 when $c = m$; i.e., there are no matching n-grams higher than $n = 1$. By default, METEOR script counts the words that match exactly, and words that match after a simple Porter stemmer. Additional matching modules including WordNet stemming and synonymy may also be used. When multiple references are provided, the lowest score is reported. Full test set scores are obtained by accumulating statistics over all test sentences. The METEOR scores are also between 0 and 1, higher being better. The scores in the results section are reported as percentages.

Translation edit rate (TER) (Snover et al., 2006) has been proposed as more intuitive evaluation metric since it is based on the rate of edits required to transform the hypothesis into the reference. The TER score is computed as follows

$$\mathrm{TER}(E, E_r) = \frac{\mathrm{Ins} + \mathrm{Del} + \mathrm{Sub} + \mathrm{Shft}}{N_r} \qquad (3)$$

where N_r is the reference length. The only difference to word error rate is that the TER allows shifts. A shift of a sequence of words is counted as a single edit. The minimum translation edit alignment is usually found through a beam search. When multiple references are provided, the edits from the closest reference are divided by the average reference length. Full test set scores are obtained by accumulating the edits and the average reference lengths. The perfect TER score is 0, and otherwise higher than zero. The TER score may also be higher than 1 due to insertions. Also TER is reported as a percentage in the results section.

3 Confusion Network Decoding

Confusion network decoding in MT has to pick one hypothesis as the skeleton which determines the word order of the combination. The other hypotheses are aligned against the skeleton. Either votes or some form of confidences are assigned to each word in the network. For example using "cat sat the mat" as the skeleton, aligning "cat sitting on the mat" and "hat on a mat" against it might yield the following alignments:

cat	sat	ε	the	mat
cat	sitting	on	the	mat
hat	ε	on	a	mat

where ε represents a NULL word. In graphical form, the resulting confusion network is shown in Figure 1. Each arc represents an alternative word at that position in the sentence and the number of votes for each word is marked in parentheses. Confusion network decoding usually requires finding the path with the highest confidence in the network. Based on vote counts, there are three alternatives in the example: "cat sat on the mat", "cat on the mat" and "cat sitting on the mat", each having accumulated 10 votes. The alignment procedure plays an important role, as by switching the position of the word 'sat' and the following NULL in the skeleton, there would be a single highest scoring path through the network; that is, "cat on the mat".

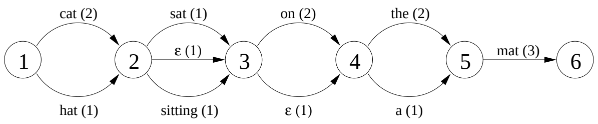

Figure 1: Example consensus network with votes on word arcs.

Different alignment methods yield different confusion networks. The modified Levenshtein alignment as used in TER is more natural than simple edit distance such as word error rate since machine translation hypotheses may have different word orders while having the same meaning. As the skeleton determines the word order, the quality of the combination output also depends on which hypothesis is chosen as the skeleton. Since the modified Levenshtein alignment produces TER scores between the skeleton and the other hypotheses, a natural choice for selecting the skeleton is the minimum average TER score. The hypothesis resulting in the lowest average TER score when aligned against all other hypotheses is chosen as the skeleton E_s as follows

$$E_s = \operatorname*{arg\,min}_{E \in E_i} \sum_{j=1}^{N_s} \mathrm{TER}(E_j, E_i) \qquad (4)$$

where N_s is the number of systems. This is equivalent to minimum Bayes risk decoding with uniform posterior probabilities (Sim et al., 2007). Other evaluation metrics may also be used as the MBR loss function. For BLEU and METEOR, the loss function would be $1 - \mathrm{BLEU}(E_j, E_i)$ and $1 - \mathrm{MTR}(E_j, E_i)$.

It has been found that multiple hypotheses from each system may be used to improve the quality of the combination output (Sim et al., 2007). When using N-best lists from each system, the words may be assigned a different score based on the rank of the hypothesis. In (Rosti et al., 2007), simple $1/(1+k)$ score was assigned to the word coming from the kth-best hypothesis. Due to the computational burden of the TER alignment, only 1-best hypotheses were considered as possible skeletons, and $n = 10$ hypotheses per system were aligned. Similar approach to estimate word posteriors is adopted in this work.

System weights may be used to assign a system specific confidence on each word in the network. The weights may be based on the systems' relative performance on a separate development set or they may be automatically tuned to optimize some evaluation metric on the development set. In (Rosti et al., 2007), the total confidence of the nth best confusion network hypothesis $E_{j,n}$, including NULL words, given the jth source sentence F_j was given by

$$c(E_{j,n}|F_j) = \sum_{i=1}^{N_j-1} \sum_{l=1}^{N_s} \lambda_l c_{wli} + \mu N_{nulls}(E_{j,n}) \tag{5}$$

where N_j is the number of nodes in the confusion network for the source sentence F_j, N_s is the number of translation systems, λ_l is the lth system weight, c_{wli} is the accumulated confidence for word w produced by system l between nodes i and $i+1$, and μ is a weight for the number of NULL links along the hypothesis $N_{nulls}(E_{j,n})$. The word confidences c_{wli} were increased by $1/(1+k)$ if the word w aligns between nodes i and $i+1$ in the network. If no word aligns between nodes i and $i+1$, the NULL word confidence at that position was increased by $1/(1+k)$. The last term controls the number of NULL words generated in the output and may be viewed as an insertion penalty. Each arc in the confusion network carries the word label w and N_s scores c_{wli}. The decoder outputs the hypothesis with the highest $c(E_{j,n}|F_j)$ given the current set of weights.

3.1 Discussion

There are several problems with the previous confusion network decoding approaches. First, the decoding can generate ungrammatical hypotheses due to alignment errors and phrases broken by the word-level decoding. For example, two synonymous words may be aligned to other words not already aligned, which may result in repetitive output. Second, the additive confidence scores in Equation 5 have no probabilistic meaning and cannot therefore be combined with language model scores. Language model expansion and re-scoring may help by increasing the probability of more grammatical hypotheses in decoding. Third, the system weights are independent of the skeleton selection. Therefore, a hypothesis from a system with a low or zero weight may be chosen as the skeleton.

4 Log-Linear Combination with Arbitrary Features

To address the issue with ungrammatical hypotheses and allow language model expansion and re-scoring, the hypothesis confidence computation is modified. Instead of summing arbitrary confidence scores as in Equation 5, word posterior probabilities are used as follows

$$\log p(E_{j,n}|F_j) = \sum_{i=1}^{N_j-1} \log\Big(\sum_{l=1}^{N_s} \lambda_l p(w|l,i)\Big) + \nu L(E_{j,n}) + \mu N_{nulls}(E_{j,n}) + \xi N_{words}(E_{j,n}) \tag{6}$$

where ν is the language model weight, $L(E_{j,n})$ is the LM log-probability and $N_{words}(E_{j,n})$ is the number of words in the hypothesis $E_{j,n}$. The word posteriors $p(w|l,i)$ are estimated by scaling the confidences c_{wli} to sum to one for each system l over all words w in between nodes i and $i+1$. The system weights are also constrained to sum to one. Equation 6 may be viewed as a log-linear sum of sentence-level features. The first feature is the sum of word log-posteriors, the second is the LM log-probability, the third is the log-NULL score and the last is the log-length score. The last two terms are not completely independent but seem to help based on experimental results.

The number of paths through a confusion network grows exponentially with the number of nodes. Therefore expanding a network with an n-gram language model may result in huge lattices if n is high. Instead of high order n-grams with heavy pruning, a bi-gram may first be used to expand the lattice. After optimizing one set of weights for the expanded confusion network, a second set of weights for N-best list re-scoring with a higher order n-gram model may be optimized. On a test set, the first set of weights is used to generate an N-best list from the bi-gram expanded lattice. This N-best list is then re-scored with the higher order n-gram. The second set of weights is used to find the final 1-best from the re-scored N-best list.

5 Multiple Confusion Network Decoding

As discussed in Section 3, there is a disconnect between the skeleton selection and confidence estimation. To prevent the 1-best from a system with a low or zero weight being selected as the skeleton, confusion networks are generated for each system and the average TER score in Equation 4 is used to estimate a prior probability for the corresponding network. All N_s confusion networks are connected to a single start node with NULL arcs which contain the prior probability from the system used as the skeleton for that network. All confusion network are connected to a common end node with NULL arcs. The final arcs have a probability of one. The prior probabilities in the arcs leaving the first node will be multiplied by the corresponding system weights which guarantees that a path through a network generated around a 1-best from a system with a zero weight will not be chosen.

The prior probabilities are estimated by viewing the negative average TER scores between the skeleton and other hypotheses as log-probabilities. These log-probabilities are scaled so that the priors sum to one. There is a concern that the prior probabilities estimated this way may be inaccurate. Therefore, the priors may have to be smoothed by a tunable exponent. However, the optimization experiments showed that the best performance was obtained by having a smoothing factor of 1 which is equivalent to the original priors. Thus, no smoothing was used in the experiments presented later in this paper.

An example joint network with the priors is shown in Figure 2. This example has three confusion networks with priors 0.5, 0.2 and 0.3. The total number of nodes in the network is represented by N_a. Similar combination of multiple confusion networks was presented in (Matusov et al., 2006). However, this approach did not include sentence specific prior estimates, word posterior estimates, and did not allow joint optimization of the system and feature weights.

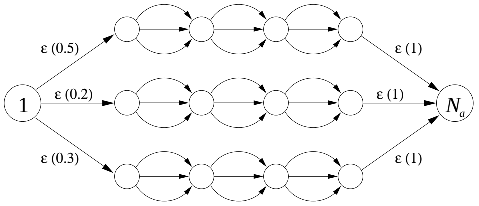

Figure 2: Three confusion networks with prior probabilities.

6 Weights Optimization

The optimization of the system and feature weights may be carried out using N-best lists as in (Ostendorf et al., 1991). A confusion network may be represented by a word lattice and standard tools may be used to generate N-best hypothesis lists including word confidence scores, language model scores and other features. The N-best list may be re-ordered using the sentence-level posteriors $p(E_{j,n}|F_j)$ from Equation 6 for the jth source sentence F_j and the corresponding nth hypothesis $E_{j,n}$. The current 1-best hypothesis $\hat{E}_j$ given a set of weights $\theta = \{\lambda_1, \ldots, \lambda_{N_s}, \nu, \mu, \xi\}$ may be represented as follows

$$\hat{E}_j(F_j|\theta) = \arg\max_{E \in E_{j,n}} p(E_{j,n}|F_j) \tag{7}$$

The objective is to optimize the 1-best score on a development set given a set of reference translations. For example, estimating weights which minimize TER between a set of 1-best hypothesis $\hat{S}$ and reference translations S_r can be written as

$$\hat{\theta} = \arg\min_{\theta} \mathrm{TER}(\hat{S}, S_r) \tag{8}$$

This objective function is very complicated, so gradient-based optimization methods may not be used. In this work, modified Powell's method as proposed by (Brent, 1973) is used. The algorithm explores better weights iteratively starting from a set of initial weights. First, each dimension is optimized using a grid-based line minimization algorithm. Then, a new direction based on the changes in the objective function is estimated to speed up the search. To improve the chances of finding a global optimum, 19 random perturbations of the initial weights are used in parallel optimization runs. Since the N-best list represents only a small portion of all hypotheses in the confusion network, the optimized weights from one iteration may be used to generate a new N-best list from the lattice for the next iteration. Similarly, weights which maximize BLEU or METEOR may be optimized.

The same Powell's method has been used to estimate feature weights of a standard feature-based phrasal MT decoder in (Och, 2003). A more efficient algorithm for log-linear models was also proposed. In this work, both the system and feature weights are jointly optimized, so the efficient algorithm for the log-linear models cannot be used.

7 Results

The improved system combination method was compared to a simple confusion network decoding without system weights and the method proposed in (Rosti et al., 2007) on the Arabic to English and Chinese to English NIST MT05 tasks. Six MT systems were combined: three (A,C,E) were phrase-based similar to (Koehn, 2004), two (B,D) were hierarchical similar to (Chiang, 2005) and one (F) was syntax-based similar to (Galley et al., 2006). All systems were trained on the same data and the outputs used the same tokenization. The decoder weights for systems A and B were tuned to optimize TER, and others were tuned to optimize BLEU. All decoder weight tuning was done on the NIST MT02 task.

The joint confusion network was expanded with a bi-gram language model and a 300-best list was generated from the lattice for each tuning iteration. The system and feature weights were tuned on the union of NIST MT03 and MT04 tasks. All four reference translations available for the tuning and test sets were used. A first set of weights with the bi-gram LM was optimized with three iterations. A second set of weights was tuned for 5-gram N-best list re-scoring. The bi-gram and 5-gram English language models were trained on about 7 billion words. The final combination outputs were detokenized and cased before scoring.

The tuning set results on the Arabic to English NIST MT03+MT04 task are shown in Table 1. The best score on each metric is shown in bold face fonts. The row labeled as `no weights` corresponds to Equation 5 with uniform system weights λ_l and zero NULL weight. The `baseline` corresponds to Equation 5 with TER tuned weights. The following three rows correspond to the improved confusion network decoding with different optimization metrics. As expected, the scores on the metric used in tuning are the best on that metric. Also, the combination results are better than any single system on all metrics in the case of TER and BLEU tuning. However, the METEOR tuning yields extremely high TER and low BLEU scores. This must be due to the higher weight on the recall compared to precision in the harmonic mean used to compute the METEOR

Arabic tuning	TER	BLEU	MTR
system A	44.93	45.71	66.09
system B	46.41	43.07	64.79
system C	46.10	46.41	65.33
system D	44.36	46.83	66.91
system E	45.35	45.44	65.69
system F	47.10	44.52	65.28
no weights	42.35	48.91	67.76
baseline	42.19	49.86	68.34
TER tuned	**41.88**	51.45	68.62
BLEU tuned	42.12	**51.72**	68.59
MTR tuned	54.08	38.93	**71.42**

Table 1: Mixed-case TER and BLEU, and lower-case METEOR scores on Arabic NIST MT03+MT04.

Arabic test	TER	BLEU	MTR
system A	42.98	49.58	69.86
system B	43.79	47.06	68.62
system C	43.92	47.87	66.97
system D	40.75	52.09	71.23
system E	42.19	50.86	70.02
system F	44.30	50.15	69.75
no weights	39.33	53.66	71.61
baseline	39.29	54.51	72.20
TER tuned	**39.10**	55.30	72.53
BLEU tuned	39.13	**55.48**	72.81
MTR tuned	51.56	41.73	**74.79**

Table 2: Mixed-case TER and BLEU, and lower-case METEOR scores on Arabic NIST MT05.

Chinese tuning	TER	BLEU	MTR
system A	56.56	29.39	54.54
system B	55.88	30.45	54.36
system C	58.35	32.88	56.72
system D	57.09	36.18	57.11
system E	57.69	33.85	58.28
system F	56.11	36.64	58.90
no weights	53.11	37.77	59.19
baseline	53.40	38.52	59.56
TER tuned	**52.13**	36.87	57.30
BLEU tuned	53.03	**39.99**	58.97
MTR tuned	70.27	28.60	**63.10**

Table 3: Mixed-case TER and BLEU, and lower-case METEOR scores on Chinese NIST MT03+MT04.

Chinese test	TER	BLEU	MTR
system A	56.57	29.63	56.63
system B	56.30	29.62	55.61
system C	59.48	31.32	57.71
system D	58.32	33.77	57.92
system E	58.46	32.40	59.75
system F	56.79	35.30	60.82
no weights	53.80	36.17	60.75
baseline	54.34	36.44	61.05
TER tuned	**52.90**	35.76	58.60
BLEU tuned	54.05	**37.91**	60.31
MTR tuned	72.59	26.96	**64.35**

Table 4: Mixed-case TER and BLEU, and lower-case METEOR scores on Chinese NIST MT05.

score. Even though METEOR has been shown to be a good metric on a given MT output, tuning to optimize METEOR results in a high insertion rate and low precision. The Arabic test set results are shown in Table 2. The TER and BLEU optimized combination results beat all single system scores on all metrics. The best results on a given metric are again obtained by the combination optimized for the corresponding metric. It should be noted that the TER optimized combination has significantly higher BLEU score than the TER optimized baseline. Compared to the baseline system which is also optimized for TER, the BLEU score is improved by 0.97 points. Also, the METEOR score using the METEOR optimized weights is very high. However, the other scores are worse in common with the tuning set results.

The tuning set results on the Chinese to English NIST MT03+MT04 task are shown in Table 3. The baseline combination weights were tuned to optimize BLEU. Again, the best scores on each metric are obtained by the combination tuned for that metric. Only the METEOR score of the TER tuned combination is worse than the METEOR scores of systems E and F - other combinations are better than any single system on all metrics apart from the METEOR tuned combinations. The test set results follow clearly the tuning results again - the TER tuned combination is the best in terms of TER, the BLEU tuned in terms of BLEU, and the METEOR tuned in terms of METEOR. Compared to the baseline, the BLEU score of the BLEU tuned combination is improved by 1.47 points. Again, the METEOR tuned weights hurt the other metrics significantly.

8 Conclusions

An improved confusion network decoding method combining the word posteriors with arbitrary features was presented. This allows the addition of language model scores by expanding the lattices or re-scoring N-best lists. The LM integration should result in more grammatical combination outputs. Also, confusion networks generated by using the 1-best hypothesis from all systems as the skeleton were used with prior probabilities derived from the average TER scores. This guarantees that the best path will not be found from a network generated for a system with zero weight. Compared to the earlier system combination approaches, this method is fully automatic and requires very little additional information on top of the development set outputs from the individual systems to tune the weights.

The new method was evaluated on the Arabic to English and Chinese to English NIST MT05 tasks. Compared to the baseline from (Rosti et al., 2007), the new method improves the BLEU scores significantly. The combination weights were tuned to optimize three automatic evaluation metrics: TER, BLEU and METEOR. The TER tuning seems to yield very good results on Arabic - the BLEU tuning seems to be better on Chinese. It also seems like

METEOR should not be used in tuning due to high insertion rate and low precision. It would be interesting to know which tuning metric results in the best translations in terms of human judgment. However, this would require time consuming evaluations such as human mediated TER post-editing (Snover et al., 2006).

The improved confusion network decoding approach allows arbitrary features to be used in the combination. New features may be added in the future. Hypothesis alignment is also very important in confusion network generation. Better alignment methods which take synonymy into account should be investigated. This method could also benefit from more sophisticated word posterior estimation.

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

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

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. Multiengine machine translation guided by explicit word matching. In *Proc. EAMT*, pages 143–152.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proc. AMTA*, pages 115–124.

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.

Mari Ostendorf, Ashvin Kannan, Steve Austin, Owen Kimball, Richard Schwartz, and Jan Robin Rohlicek. 1991. Integration of diverse recognition methodologies through reevaluation of N-best sentence hypotheses. In *Proc. HLT*, pages 83–87.

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.

Antti-Veikko I. Rosti, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Necip Fazil Ayan, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In *Proc. NAACL-HLT 2007*, pages 228–235.

Robert E. Schapire. 1990. The strength of weak learnability. *Machine Learning*, 5(2):197–227.

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*, volume 4, pages 105–108.

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*, pages 223–231.