The RWTH System Combination System for WMT 2011

Gregor Leusch, Markus Freitag, and Hermann Ney
RWTH Aachen University
Aachen, Germany
{leusch,freitag,ney}@cs.rwth-aachen.de

Abstract

RWTH participated in the System Combination task of the Sixth Workshop on Statistical Machine Translation (WMT 2011).

For three language pairs, we combined 6 to 14 systems into a single consensus translation. A three-level meta-combination scheme combining six different system combination setups with three different engines was applied on the French–English language pair. Depending on the language pair, improvements versus the best single system are in the range of $+1.9\%$ and $+2.5\%$ abs. on BLEU, and between -1.8% and -2.4% abs. on TER. Novel techniques compared with RWTH's submission to WMT 2010 include two additional system combination engines, an additional word alignment technique, meta combination, and additional optimization techniques.

1 Introduction

RWTH's main approach to System Combination (SC) for Machine Translation (MT) is a refined version of the ROVER approach in Automatic Speech Recognition (ASR) (Fiscus, 1997), with additional steps to cope with reordering between different hypotheses, and to use true casing information from the input hypotheses. The basic concept of the approach has been described by Matusov et al. (2006). Several improvements have been added later (Matusov et al., 2008). This approach includes an enhanced alignment and reordering framework. In contrast to existing approaches (Jayaraman and Lavie, 2005; Rosti et al., 2007b), the context of the whole corpus rather than a single sentence is considered in this iterative, unsupervised procedure, yielding a more reliable alignment. Majority voting on the generated lattice is performed using prior weights for each system as well as other statistical models such as a special n-gram language model. True casing is considered a separate step in RWTH's approach, which also takes the input hypotheses into account. The pipeline, and consequently the description of the main pipeline given in this paper, is based on our pipeline for WMT 2010 (Leusch and Ney, 2010), with extensions as described. When necessary, we denote this pipeline as *Align-to-Lattice*, or *A2L* .

For the French–English task, we used two additional system combination engines for the first time: The first one uses the same alignments as A2L, but generates lattices in the OpenFST framework (Allauzen et al., 2007). The OpenFST decoder (`fstshortestpath`) is then used to find the best path (consensus translation) in this lattice. Analogously, we call this engine *A2FST* . The second additional engine, which we call *SCUNC,* uses a TER-based alignment, similar to the approach by Rosti et al. (2007b). Instead of a lattice rescoring, finding the consensus translation is considered a per-node classification problem: For each slot, which one is the "correct" one (i.e. will give the "best" output)? This approach is inspired by iROVER (Hillard et al., 2007). Consensus translations from different settings of these approaches could then be combined again by an additional application of system combination – which we refer to as *meta combination* (Rosti et al., 2007a). These three approaches are described in more detail in Section 2. In Section 3 we describe how we tuned the parameters and decisions of our system combination approaches for WMT 2011. Section 4 then lists our experimental setup as well as the experimental results we obtained on the WMT 2011 system combination track. We conclude this paper in Section 5.

2 System Combination Algorithm (A2L)

In this section we present the details of our main system combination method, A2L. The upper part of Figure 1 gives an overview of the system combination architecture described in this section. After preprocessing the MT hypotheses, pairwise align-

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 152–158,
Edinburgh, Scotland, UK, July 30–31, 2011.

ments between the hypotheses are calculated. The hypotheses are then reordered to match the word order of a selected *primary* (*skeleton*) hypothesis. From this, we create a confusion network (CN) which we then rescore using system prior weights and a language model (LM). The single best path in this CN then constitutes the consensus translation. The consensus translation is then true cased and post processed.

2.1 Word Alignment

The main proposed alignment approach is a statistical one. It takes advantage of multiple translations for a whole corpus to compute a consensus translation for each sentence in this corpus. It also takes advantage of the fact that the sentences to be aligned are in the same language.

For each of the K source sentences in the test corpus, we select one of its N translations from different MT systems $E, m = 1, \ldots, N$, as the *primary* hypothesis. Then we align the *secondary* hypotheses $E_n (n = 1, \ldots, ; n \neq m)$ with E_n to match the word order in E_n. Since it is not clear which hypothesis should be primary, i. e. has the "best" word order, we let several or all hypothesis play the role of the primary translation, and align all pairs of hypotheses $(E_n, E_m);\ n \neq m$.

The word alignment is *trained* in analogy to the alignment training procedure in statistical MT. The difference is that the two sentences that have to be aligned are in the same language. We use the IBM Model 1 (Brown et al., 1993) and the Hidden Markov Model (HMM, (Vogel et al., 1996)) to estimate the alignment model.

The alignment training corpus is created from a test corpus of effectively $N \cdot (N-1) \cdot K$ sentences translated by the involved MT engines. Model parameters are trained iteratively using the GIZA++ toolkit (Och and Ney, 2003). The training is performed in the directions $E_m \rightarrow E_n$ and $E_n \rightarrow E_m$. The final alignments are determined using a cost matrix C for each sentence pair (E_m, E_n). Elements of this matrix are the local costs $C(j, i)$ of aligning a word $e_{m,j}$ from E_m to a word $e_{n,i}$ from E_n. Following Matusov et al. (2004), we compute these local costs by interpolating the negated logarithms of the state occupation probabilities from the "source-to-target" and "target-to-source" training of the HMM model.

A different approach that has e.g. been proposed by Rosti et al. (2007b) is the utilization of a TER alignment (Snover et al., 2006) for this purpose. Because the original TER is insensitive to small changes in spellings, synonyms etc., it has been proposed to use more complex variants, e.g. TERp. For our purposes, we utilized "poor-man's-stemming", i.e. shortening each word to its first four characters when calculating the TER alignment. Since a TER alignment already implies a reordering between the primary and the secondary hypothesis, an explicit reordering step is not necessary.

2.2 Word Reordering and Confusion Network Generation

After reordering each secondary hypothesis E_m and the rows of the corresponding alignment cost matrix, we determine $N-1$ monotone *one-to-one* alignments between E_n as the primary translation and $E_m, m = 1, \ldots, N; m \neq n$. We then construct the confusion network.

We consider words without a correspondence to the primary translation (and vice versa) to have a null alignment with the empty word ε, which will be transformed to an ε-arc in the corresponding confusion network.

The $N-1$ monotone one-to-one alignments can then be transformed into a confusion network, as described by Matusov et al. (2008).

2.3 Voting in the Confusion Network (A2L, A2FST)

Instead of choosing a fixed sentence to define the word order for the consensus translation, we generate confusion networks for N possible hypotheses as primary, and unite them into a single lattice. In our experience, this approach is advantageous in terms of translation quality compared to a minimum Bayes risk primary (Rosti et al., 2007b).

Weighted majority voting on a single confusion network is straightforward and analogous to ROVER (Fiscus, 1997). We sum up the probabilities of the arcs which are labeled with the same word and have the same start state and the same end state.

Compared to A2L, our new A2FST engine allows for a higher number of features for each arc. Consequently, we add a binary system feature for each system in addition to the logarithm of the sum of system weights, as before. The advantage of these features is that the weights are linear within a log-linear model, as opposed to be part of a logarithmic sum. Consequently they can later be optimized using techniques designed for linear feature weights, such as MERT, or MIRA.

2.4 Language Models

The lattice representing a union of several confusion networks can then be directly rescored with an n-gram language model (LM). When regarding

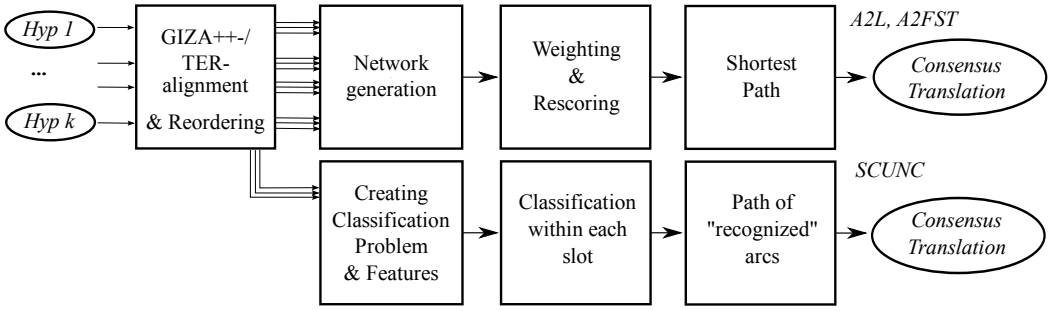


Figure 1: The system combination architecture.

the lattice as a weighted Finite State Transducer (FST), this can be regarded (and implemented) as composition with a LM FST.

In our approach, we train a trigram LM on the outputs of the systems involved in system combination. For LM training, we take the system hypotheses for the same test corpus for which the consensus translations are to be produced. Using this "adapted" LM for lattice rescoring thus gives bonus to n-grams from the original system hypotheses, in most cases from the original phrases. Presumably, many of these phrases have a correct word order. Previous experimental results show that using this LM in rescoring together with a word penalty notably improves translation quality. This even results in better translations than using a "classical" LM trained on a monolingual training corpus. We attribute this to the fact that most of the systems we combine already include such general LMs. Nevertheless, one of the SC systems we use for the French–English task (`IV` in Section 4.1) uses a filtered fourgram LM trained on GigaWord and other constrained training data sets for this WMT tasks as an additional LM.

2.5 Extracting Consensus Translations

To generate our consensus translation, we extract the single-best path from the rescored lattice, using "classical" decoding as in MT. In A2L, this is implemented as shortest-path decoder on a pruned lattice. In A2FST, we use the OpenFST `fstshortestpath` decoder, which does not require a pruning step for lattices of the size and density produced here.

2.6 Classification in the Confusion Network (SCUNC)

Instead of considering the selection of the consensus problem as a shortest-path problem in a rescored confusion network, we can treat it instead as a classification problem: For each slot (set of outgoing arcs from one node in a CN), we consider one or more arcs to be "correct", and train a classifier to identify these certain arcs. This is the idea of the iROVER approach in ASR (Hillard et al., 2007). We call our implementation *System Combination Using N-gram Classifiers*, or *SCUNC*.

For the WMT evaluation, we used the ICSI-Boost framework (Favre et al., 2007) as classifier (in binary mode, i.e. giving a yes/no-decision for each single arc). We generated 109 features from 8 families: Pairwise equality of words from different systems, Number of votes for a word, word that would win a simple majority voting, empty word (also in previous two arcs), position at beginning or end of sentence, cross-BLEU-S score of hypothesis, equality of system with system of last slot, and SRILM uni- to trigram scores. As this approach requires strict CN instead of lattices, a union of CNs for different primary hypotheses was no longer possible. We decided to select a fixed single primary system; other approaches would have been to train an additional classifier for this purpose, or to select a minimum-Bayes-risk (MBR) skeleton.

2.7 Consensus True Casing

Previous approaches to achieve true cased output in system combination operated on true-cased lattices, used a separate input-independent true caser, or used a general true-cased LM to differentiate between alternative arcs in the lattice, as described by Leusch et al. (2009). For WMT 2011, we use per-sentence information from the input systems to determine the consensus case of each output word. Lattice generation, rescoring, and reranking are performed on lower-cased input, with a lower-cased consensus hypothesis as their result. For each word in this hypothesis, we count how often each casing variant occurs in the input hypotheses for this sentence. We then use the variant with the highest support for the final consensus output.

Table 1: Corpus and Task statistics.

	avg. # words			#sys
	TUNE	DEV	TEST	
FR–EN	15670	11410	49832	25
DE–EN	15508	10878	49395	24
ES–EN	15989	11234	50612	15
# sent	609	394	2000	

3 Tuning

3.1 Feature weights

For lattice rescoring, we selected a linear combination of BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) as optimization criterion, $\hat{\Theta} := \text{argmax}_{\Theta} \{BLEU - TER\}$ for the A2L engine, based on previous experience (Mauser et al., 2008). To achieve more stable results, we use the case-insensitive variants for both measures, despite the explicit use of case information in the pipeline. System weights were tuned to this criterion using the Downhill Simplex method.

In the A2FST setup, we were able to generate full lattices, with separate costs for each individual feature on all arcs (Power Semiring). This allowed us to run Lattice MERT (Macherey et al., 2008) on the full lattice, with no need for pruning (and thus additional outer iterations for re-generating lattices). We tried different strategies – random lines vs axis-parallel lines, regularization, random restarts, etc, and selected the most stable results on TUNE and DEV for this engine. Optimization criterion here was BLEU.

3.2 Training a classifier for SCUNC

In MT system combination, even with given reference translations, there is no simple way to identify the "correct" arc in a slot. This renders a classifier-based approach even more difficult than iROVER in ASR. The problem is even aggravated because both the alignment of words, and their order, can be incorrect already in the CN. We thus consider an arc to be "correct" within this task exactly if it gives us the best possible total BLEU-S score.[1] These "correct" arcs, which lie on such an "oracle path" for BLEU-S, were therefore used as reference classes when training the classifier.

3.3 System Selection

With the large numbers of input systems – e.g., 25 for FR–EN – and their large spread in translation quality – e.g. from 22.2 to 31.4% in BLEU – not all systems should participate in the system combination process. This is especially the case since several of these e.g. 25 systems are often only small variants of each other (contrastive vs. primary submissions), which leads to a low variability of these translations. We considered several variants of the set of input systems, often starting from the top, and either replacing some of the systems very similar to others with systems further down the list, or not considering those as primary, adding further systems as additional secondaries. Depending on the engine we were using, we selected between 6 and 14 different systems as input.

[1] We are looking at the sentence level, so we use BLEU-S (Lin and Och, 2004) instead of BLEU

4 Experimental Results

Each language pair in WMT 2011 had its own set of systems, so we selected and tuned separately for each language pair . Due to time constraints, we only participated in tasks with English as the target language. In preliminary experiments, it turned out that System Combination was not able to get a better result than the best single system on the Czech–English task. Consequently, we focused on the language pairs French–English, German–English, and Spanish–English.

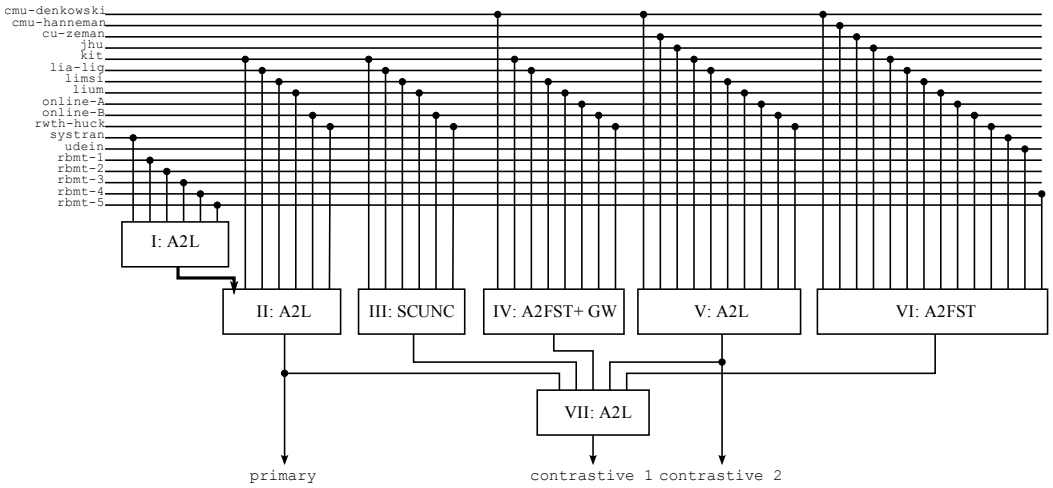
We split the available tuning data documentwise into a 609-line TUNE set (for tuning), and a 394-line DEV set (to verify tuning results). More statistics on these sets can be found in Table 1.

Unfortunately, late in the evaluation campaign it turned out that the quality of several reference sentences used in TUNE and DEV was rather low: Many reference sentences contained spelling errors, a few dozen lines even contained French phrases or sentences within or after the English text. We corrected many of these errors manually in the references. In total 101 of 690 lines (16.6%) in TUNE and 58 of 394 lines (14.7%) in DEV were affected by this. While it was too late to rerun all of the optimization runs, we re-optimized at least a few final systems. All scores within this section were calculated on the corrected reference translations.

4.1 FR–EN

For French–English, we built in total seven different system combination setups to generate a single consensus translation and two contrastive translations. Figure 2 shows the structure and the data flow of our setup for FR–EN. Table 2 lists more details about the individual engines.

Our primary submission was focused on our experience that while rule-based MT systems (such as `RBMT-1..5` and `systran`) tend to have lower BLEU scores than statistical (SMT) systems, they usually give considerable improve-



Bold arrows denote a system that is always considered as skeleton.
Note that there are two variants of setup `II`, see text.

Figure 2: System combination pipelines for FR–EN

Table 2: Engines and input systems for FR–EN.

	Engine	# Input	submitted?
I	A2L	6 RBMT	
II	A2L	I + 6	`primary`
II'	A2L	fix I + 6	*for VII*
III	SCUNC	6	
IV	A2FST	GW, 8	
V	A2L	10	`contrastive-2`
VI	A2FST	14	
VII	A2L	II'–VI	`contrastive-1`

"GW" means a 4-gram LM trained on GigaWord.
`II` uses all skeletons, `II'` uses `I` as fixed skeleton.

Table 3: Results for FR–EN.

	TUNE		DEV	
	BLEU	TER	BLEU	TER
`kit`	31.56	50.15	30.25	52.88
`systran`	28.18	53.32	26.50	56.07
I	27.37	54.73	26.72	57.73
II	33.69	48.47	32.45	51.09
II'	33.39	48.77	31.81	51.57
III	32.74	48.06	31.88	50.87
IV	34.16	48.31	31.95	51.64
V	33.17	48.95	32.60	51.14
VI	33.86	48.69	31.56	52.25
VII	34.41	48.20	32.15	51.49

`kit` *is the best single system.*
`systran` *is the best single rule-based system.*
All scores are case insensitive, and were calculated on the corrected reference translations.

ments to the latter in a SC setup. Here, though, the number of such systems was too high to simply add them to a reasonable set of SMT systems. Consequently, we first built a SC system (`I`) combining all RBMT/Systran systems, and then a second SC system (`II`) which combines the output of `I`, and 6 SMT systems. As further experiments showed, allowing all hypotheses as primary (or skeleton) gave significantly better scores than forcing SC to use the output of `I` as primary only. But vice versa, when looking at the meta combination scheme, `VII`, using `I` as primary only (a setup which we will now denote as `II'`) gave measurable improvements in the overall translation quality. We assume this is due to the similarity of the output of `II` with that of the other setups.

Setup `III` is a SCUNC setup, that is, we built a single CN for each sentence using poor-man's-stemming-TER, with `rwth-huck` as primary hypothesis. We then generated a large number of features for each arc, and trained an ICSIBoost classifier to recognize the arc (or system) that gave the best BLEU-S score. This then gave us the consensus translation.

For `IV`, we built an OpenFST lattice out of eight systems, and rescored it with both the Hypothesis LM (3-gram), and a 4-gram LM trained on GigaWord and other WMT constrained training data for this task. The log-linear weights were trained using lattice MERT for BLEU. Setup `V` is a classical A2L setup, using ten different input systems. This setup was tuned on BLEU – TER using the Downhill-Simplex algorithm. In setup `VI`, again the A2FST engine was used, this time using the Hyp LM only, without an additional LM. Tuning

Table 4: Results for DE–EN.

	TUNE		DEV	
	BLEU	TER	BLEU	TER
online-B	23.13	60.15	26.20	57.20
Primary	24.57	58.51	28.11	54.83
4 best sys	23.85	58.22	27.47	54.96
6 best sys	24.46	57.74	27.82	54.50

online-B *is the best single system.*

Table 5: Results for ES–EN.

	TUNE		DEV	
	BLEU	TER	BLEU	TER
online-A	30.58	51.69	30.77	51.95
Primary	34.29	48.47	33.41	49.71
Contrastive	34.23	48.27	33.30	49.51

online-A *is the best single system.*

was also performed using lattice MERT towards BLEU. And finally, setup VII combines the output of II' to IV using the A2L engine again.

All the results of system combination on TUNE and DEV are listed in Table 3. It turns out that with the exception of I, all system combination approaches were able to achieve a significant improvement of at least $+1.8\%$ abs. in BLEU compared to the best input system. For I, we need to keep in mind that all other systems were several BLEU points worse than the best one – a scenario where we can expect system combination, which is based on the *consensus* translation after all, to underperform. We also see that both A2FST and SCUNC, with their large number of features, show a tendency to overfitting – we see large improvements on TUNE, but significantly smaller improvements on DEV. This tendency is, unfortunately, also the case for meta combination: While we see an additional $+0.3\%$ abs. in BLEU over the best first-level system combination on TUNE, this improvement does not reflect in the scores on DEV: While we still see a $+0.2\%$ abs. improvement in BLEU over the setup that performed best on TUNE, there is even a small deterioration of -0.4% in BLEU over the setup that performed best on DEV. Because of this effect, we decided to submit our meta combination output only as first contrastive, and the output that performed well both on TUNE and DEV as our primary submission for WMT. As second contrastive submission, we selected the setup that performed best on DEV.

4.2 DE–EN

24 systems were available in the German–English language pair, but incorporating only 7 of them turned out to deliver optimal results on DEV. We ran experiments on several settings of systems, but only in our tried and tested A2L framework. We settled for a combination of seven systems (online-B,cmu-dyer,dfki-xu,limsi, online-A,rwth-wuebker,kit) as primary submission. Table 4 also lists two different settings. One setting consists of the four best systems (online-B,cmu-dyer,rwth-wuebker, kit) and the other setting contains the six best systems (online-B,cmu-dyer,dfki-xu, rwth-wuebker,online-A,kit). When we added more systems to system combination, we lost performance in both TUNE and DEV.

4.3 ES–EN

For Spanish–English, we tried several settings of systems. We sticked to our tried and tested A2L framework. We settled for a combination of six systems (alacant,koc,online-A, online-B,rbmt-1,systran) as contrastive submission, and a combination of ten systems (+rbmt-2,rbmt-3,rbmt-4,udein) as primary submission. Table 5 lists the results for this task. The difference between our primary setup (10 systems) and our contrastive setup (6 systems) is rather small, less than 0.1% abs. in BLEU. Nevertheless, we see significant improvements over the best single system of $+2.4\%$ abs. in BLEU, and -2.2% in TER.

5 Conclusions

We have shown that our system combination approach leads to significant improvements over single best MT output where a significant number of comparably good translations is available on a single language pair. A meta combination can give additional improvement, but can be sensitive to overfitting; so in some cases, using one of its input system combination hypothesis may be a better choice. In any way, both of our new engines have shown that they can compete with our present approach, so we hope to make good use of the new possibilities they may offer.

Acknowledgments

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. This work was partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

References

C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proc. of the Twelfth International Conference on Implementation and Application of Automata (CIAA)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

B. Favre, D. Hakkani-Tür, and S. Cuendet. 2007. Icsiboost. `http://code.google.come/p/icsiboost`.

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

D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schlüter, and H. Ney. 2007. iROVER: improving system combination with classification. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, NAACL-Short '07, pages 65–68. Association for Computational Linguistics.

S. Jayaraman and A. 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)*, pages 143–152, Budapest, Hungary, May.

G. Leusch and H. Ney. 2010. The rwth system combination system for wmt 2010. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 315–320, Uppsala, Sweden, July.

G. Leusch, E. Matusov, and H. Ney. 2009. The RWTH system combination system for WMT 2009. In *Fourth Workshop on Statistical Machine Translation*, pages 56–60, Athens, Greece, March. Association for Computational Linguistics.

C. Y. Lin and F. J. Och. 2004. Orange: a method for evaluation automatic evaluation metrics for machine translation. In *Proc. COLING 2004*, pages 501–507, Geneva, Switzerland, August.

W. Macherey, F. Och, I. Thayer, and J. Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–734. Association for Computational Linguistics.

E. Matusov, R. Zens, and H. Ney. 2004. Symmetric word alignments for statistical machine translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 219–225, Geneva, Switzerland, August.

E. Matusov, N. Ueffing, and H. Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 33–40, Trento, Italy, April.

E. Matusov, G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y. S. Lee, J. B. Marino, M. Paulik, S. Roukos, H. Schwenk, and H. Ney. 2008. System combination for machine translation of spoken and written language. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1222–1237, September.

A. Mauser, S. Hasan, and H. Ney. 2008. Automatic evaluation measures for statistical machine translation system optimization. In *International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May.

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

K. Papineni, S. Roukos, T. Ward, and W. J. 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)*, pages 311–318, Philadelphia, PA, July.

A. V. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. M. Schwartz, and B. J. Dorr. 2007a. Combining outputs from multiple machine translation systems. In *HLT-NAACL'07*, pages 228–235.

A. V. Rosti, S. Matsoukas, and R. Schwartz. 2007b. Improved word-level system combination for machine translation. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 312–319, Prague, Czech Republic, June.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A Study of Translation Error Rate with Targeted Human Annotation. In *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231, Boston, MA, August.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, August.