The RWTH System Combination System for WMT 2009

Gregor Leusch, Evgeny Matusov, and Hermann Ney
RWTH Aachen University
Aachen, Germany

Abstract

RWTH participated in the System Combination task of the Fourth Workshop on Statistical Machine Translation (WMT 2009). Hypotheses from 9 German→English MT systems were combined into a consensus translation. This consensus translation scored 2.1% better in BLEU and 2.3% better in TER (abs.) than the best single system. In addition, cross-lingual output from 10 French, German, and Spanish→English systems was combined into a consensus translation, which gave an improvement of 2.0% in BLEU/3.5% in TER (abs.) over the best single system.

1 Introduction

The RWTH approach to MT system combination is a refined version of the ROVER approach in 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., 2007), 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 the prior probabilities for each system as well as other statistical models such as a special n-gram language model.

2 System Combination Algorithm

In this section we present the details of our system combination method. Figure 1 gives an overview of the system combination architecture described in this section. After preprocessing the MT hypotheses, pairwise alignments between the hypotheses are calculated. The hypotheses are then reordered to match the word order of a selected primary hypothesis. From this, we create a confusion network (CN), which we then rescore using
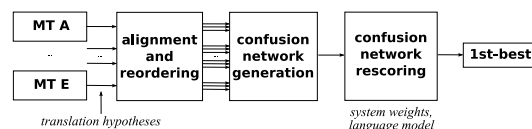

Figure 1: The system combination architecture.

system prior weights and a language model (LM). The single best path in this CN then constitutes the consensus translation.

2.1 Word Alignment

The 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 source sentence F in the test corpus, we select one of its translations $E_n, n = 1, \ldots, M$, as the *primary* hypothesis. Then we align the *secondary* hypotheses $E_m (m = 1, \ldots, M; n \neq m)$ 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 every 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[1] of effectively $M \cdot (M - 1) \cdot N$ sentences translated by the involved MT engines. The single-word based lexicon probabilities $p(e|e')$ are initialized from normalized lexicon counts collected over the sentence pairs (E_m, E_n) on this corpus. Since all of the hypotheses are in the same language, we count co-occurring identical words, i.e. whether $e_{m,j}$ is the same word as $e_{n,i}$ for some i and j. In addition, we add a fraction of a count for words with identical prefixes.

---

[1] A test corpus can be used directly because the alignment training is unsupervised and only automatically produced translations are considered.

The 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 $E_n \rightarrow E_m$. After each iteration, the updated lexicon tables from the two directions are interpolated. 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. Two different alignments are computed using the cost matrix C: the alignment $\widetilde{a}$ used for reordering each secondary translation E_m, and the alignment $\bar{a}$ used to build the confusion network.

In addition to the GIZA++ alignments, we have also conducted preliminary experiments following He et al. (2008) to exploit character-based similarity, as well as estimating $p(e|e') := \sum_f p(e|f)p(f|e')$ directly from a bilingual lexicon. But we were not able to find improvements over the GIZA++ alignments so far.

2.2 Word Reordering and Confusion Network Generation

After reordering each secondary hypothesis E_m and the rows of the corresponding alignment cost matrix according to $\widetilde{a}$, we determine $M-1$ monotone *one-to-one* alignments between E_n as the primary translation and $E_m, m = 1, \ldots, M; m \neq n$. We then construct the confusion network. In case of many-to-one connections in $\widetilde{a}$ of words in E_m to a single word from E_n, we only keep the connection with the lowest alignment costs.

The use of the one-to-one alignment $\bar{a}$ implies that some words in the secondary translation will not have a correspondence in the primary translation and vice versa. We consider these words to have a null alignment with the empty word ε. In the corresponding confusion network, the empty word will be transformed to an ε-arc.

$M-1$ monotone one-to-one alignments can then be transformed into a confusion network. We follow the approach of Bangalore et al. (2001) with some extensions. Multiple insertions with regard to the primary hypothesis are sub-aligned to each other, as described by Matusov et al. (2008). Figure 2 gives an example for the alignment.

2.3 Voting in the confusion network

Instead of choosing a fixed sentence to define the word order for the consensus translation, we generate confusion networks for all hypotheses as primary, and unite them into a single lattice. In our experience, this approach is advantageous in terms of translation quality, e.g. by 0.7% in BLEU compared to a minimum Bayes risk primary (Rosti et al., 2007). 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. To exploit the true casing abilities of the input MT systems, we sum up the scores of arcs bearing the same word but in different cases. Here, we leave the decision about upper or lower case to the language model.

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). A transformation of the lattice is required, since LM history has to be memorized.

We train a trigram LM on the outputs of the systems involved in system combination. For LM training, we took 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, since they are extracted from the training data. Previous experimental results show that using this LM in rescoring together with a word penalty (to counteract any bias towards short sentences) 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 are phrase-based systems, which already include such general LMs. Since we are using a true-cased LM trained on the hypotheses, we can exploit true casing information from the input systems by using this LM to disambiguate between the separate arcs generated for the variants (see Section 2.3).

After LM rescoring, we add the probabilities of identical partial paths to improve the estimation of the score for the best hypothesis. This is done through determinization of the lattice.

2.5 Extracting Consensus Translations

To generate our consensus translation, we extract the single-best path within the rescored confusion network. With our approach, we could also extract N-best hypotheses. In a subsequent step, these N-best lists could be rescored with additional statistical models (Matusov et al., 2008). But as we did not have the resources in the WMT 2009 evaluation, this step was dropped for our submission.

3 Tuning system weights

System weights, LM factor, and word penalty need to be tuned to produce good consensus translations. We optimize these parameters using the

system hypotheses	**0.25 would your like coffee or tea** 0.35 have you tea or Coffee 0.10 would like your coffee or 0.30 I have some coffee tea would you like
alignment and reordering	have|**would** you|**your** $|**like** Coffee|**coffee** or|**or** tea|**tea** would|**would** your|**your** like|**like** coffee|**coffee** or|**or** $\|**tea** I\|$ would|**would** you|**your** like|**like** have|$ some|$ coffee|**coffee** $|**or** tea|**tea**

confusion network

$	would	your	like	$	$	coffee	or	tea
$	have	you	$	$	$	Coffee	or	tea
$	would	your	like	$	$	coffee	or	$
I	would	you	like	have	some	coffee	$	tea

voting (normalized)

$	would	you	$	$	$	coffee	or	tea
0.7	0.65	0.65	0.35	0.7	0.7	0.5	0.7	0.9
I	have	your	**like**	have	some	**Coffee**	$	$
0.3	0.35	0.35	0.65	0.3	0.3	0.5	0.3	0.1

consensus translation	would you like coffee or tea

Figure 2: Example of creating a confusion network from monotone one-to-one word alignments (denoted with symbol |). The words of the primary hypothesis are printed in bold. The symbol $ denotes a null alignment or an ε-arc in the corresponding part of the confusion network.

Table 1: Systems combined for the WMT 2009 task. Systems written in oblique were also used in the Cross Lingual task (`rbmt3` for FR→EN).

DE→EN *google*, `liu`, *rbmt3*, `rwth`, `stuttgart`, `systran`, `uedin`, *uka*, `umd`

ES→EN *google*, `nict`, *rbmt4*, *rwth*, `talp-upc`, `uedin`

FR→EN `dcu`, *google*, `jhu`, `limsi`, *liumsystran*, *rbmt4*, `rwth`, *uedin*, `uka`

publicly available CONDOR optimization toolkit (Berghen and Bersini, 2005). For the WMT 2009 Workshop, we selected a linear combination of BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) as optimization criterion, $\hat{\Theta} := \mathrm{argmax}_\Theta \{(2 \cdot \mathrm{BLEU}) - \mathrm{TER}\}$, based on previous experience (Mauser et al., 2008). We used the whole dev set as a tuning set. For more stable results, we used the case-insensitive variants for both measures, despite the explicit use of case information in our approach.

4 Experimental results

Due to the large number of submissions (71 in total for the language pairs DE→EN, ES→EN, FR→EN), we had to select a reasonable number of systems to be able to tune the parameters in a reliable way. Based on previous experience, we manually selected the systems with the best BLEU/TER score, and tried different variations of this selection, e.g. by removing systems which had low weights after optimization, or by adding promising systems, like rule based systems.

Table 1 lists the systems which made it into our final submission. In our experience, if a large number of systems is available, using n-best translations does not give better results than using single best translations, but raises optimization time significantly. Consequently, we only used single best translations from all systems.

The results also confirm another observation: Even though rule-based systems by itself may have significantly lower automatic evaluation scores (e.g. by 2% or more in BLEU on DE→EN),

they are often very important in system combination, and can improve the consensus translation e.g. by 0.5% in BLEU.

Having submitted our translations to the WMT workshop, we calculated scores on the WMT 2009 test set, to verify the results on the tuning data. Both the results on the tuning set and on the test set can be found in the following tables.

4.1 The Google Problem

One particular thing we noticed is that in the language pairs of FR→EN and ES→EN, the translations from one provided single system (Google) were much better in terms of BLEU and TER than those of all other systems – in the former case by more than 4% in BLEU. In our experience, our system combination approach requires at least three "comparably good" systems to be able to achieve significant improvements. This was confirmed in the WMT 2009 task as well: Neither in FR→EN nor in ES→EN we were able to achieve an improvement over the Google system. For this reason, we did not submit consensus translations for these two language pairs. On the other hand, we would have achieved significant improvements over all (remaining) systems leaving out Google.

4.2 German→English (DE→EN)

Table 2 lists the scores on the tuning and test set for the DE→EN task. We can see that the best systems are rather close to each other in terms of BLEU. Also, the rule-based translation system (RBMT), here SYSTRAN, scores rather well. As a consequence, we find a large improvement using system combination: 2.9%/2.7% abs. on the tuning set, and still 2.1%/2.3% on test, which means that system combination generalizes well here.

4.3 Spanish→English (ES→EN), French→English (FR→EN)

In Table 3, we see that on the ES→EN and FR→EN tasks, a single system – Google – scores significantly better on the TUNE set than any other

Table 2: German→English task: case-insensitive scores. Best single system was Google, second best UKA, best RBMT Systran. SC stands for system combination output.

German→English	TUNE		TEST	
	BLEU	TER	BLEU	TER
Best single	23.2	59.5	21.3	61.3
Second best single	23.0	58.8	21.0	61.7
Best RBMT	21.3	61.3	18.9	63.7
SC (9 systems)	**26.1**	**56.8**	**23.4**	**59.0**
w/o RBMT	24.5	57.3	22.5	59.2
w/o Google	24.9	57.4	23.0	59.1

Table 3: Spanish→English and French→English task: scores on the tuning set after system combination weight tuning (case-insensitive). Best single system was Google, second best was Uedin (Spanish) and UKA (French). No results on TEST were generated.

Spanish→English	ES→EN		FR→EN	
	BLEU	TER	BLEU	TER
Best single	**29.5**	**53.6**	**32.2**	**50.1**
Second best single	26.9	56.1	28.0	54.6
SC (6/9 systems)	28.7	53.6	30.7	52.5
w/o Google	27.5	55.6	30.0	52.8

system, namely by 2.6%/4.2% resp. in BLEU. As a result, a combination of these systems scores better than any other system, even when leaving out the Google system. But it gives worse scores than the single best system. This is explainable, because system combination is trying to find a *consensus* translation. For example, in one case, the majority of the systems leave the French term *"wagon-lit"* untranslated; spurious translations include "baggage car", "sleeping car", and "alive". As a result, the consensus translation also contains "wagon-lit", not the correct translation "sleeper" which only the Google system provides. Even tuning all other system weights to zero would not result in pure Google translations, as these weights neither affect the LM nor the selection of the primary hypothesis in our approach.

4.4 Cross-Lingual→English (XX→EN)

Finally, we have conducted experiments on cross-lingual system combination, namely combining the output from DE→EN, ES→EN, and FR→EN systems to a single English consensus translation. Some interesting results can be found in Table 4. We see that this consensus translation scores 2.0%/3.5% better than the best single system, and 4.4%/5.6% better than the second best single system. While this is only 0.8%/2.5% better than the combination of only the three Google systems, the combination of the non-Google sys-

Table 4: Cross-lingual task: combination of German→English, Spanish→English, and French→English. Case-insensitive scores. Best single system was Google for all language pairs.

Cross-lingual → English	TUNE		TEST	
	BLEU	TER	BLEU	TER
Best single German	23.2	59.5	21.3	61.3
Best single Spanish	29.5	53.6	28.7	53.8
Best single French	32.2	50.1	31.1	51.7
SC (10 systems)	**35.5**	**46.4**	**33.1**	**48.2**
w/o RBMT	35.1	46.5	32.7	48.3
w/o Google	32.3	48.8	29.9	50.5
3 Google systems	34.2	48.0	32.3	49.2
w/o German	34.0	49.3	31.5	50.9
w/o Spanish	33.4	49.8	31.0	51.9
w/o French	30.5	51.4	28.6	52.3

tems leads to translations that could compete with the FR→EN Google system. Again, we see that RBMT systems lead to a small improvement of 0.4% in BLEU, although their scores are significantly worse than those of the competing SMT systems.

Regarding languages, we see that despite the large differences in the quality of the systems (10 points between DE→EN and FR→EN), all languages seem to provide significant information to the consensus translation: While FR→EN certainly has the largest influence (−4.5% in BLEU when left out), even DE→EN "contributes" 1.6 BLEU points to the final submission.

5 Conclusions

We have shown that our system combination system can lead to significant improvements over single best MT output where a significant number of comparably good translations is available on a single language pair. For cross-lingual system combination, we observe even larger improvements, even if the quality in terms of BLEU or TER between the systems of different language pairs varies significantly. While the input of high-quality SMT systems has the largest weight for the consensus translation quality, we find that RBMT systems can give important additional information leading to better translations.

Acknowledgments

References

S. Bangalore, G. Bordel, and G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *IEEE Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Italy, December.

F. V. Berghen and H. Bersini. 2005. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175.

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.

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*.

X. He, M. Yang, J. Gao, P. Nguyen, and R. Moore. 2008. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 98–107, Honolulu, Hawaii, October.

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.

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, S. Matsoukas, and R. Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings 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.