

# SEQUENTIAL SYSTEM COMBINATION FOR MACHINE TRANSLATION OF SPEECH

*Damianos Karakos and Sanjeev Khudanpur*

Center for Language and Speech Processing  
and Department of Electrical and Computer Engineering  
Johns Hopkins University, Baltimore, MD 21218  
email: {damianos, khudanpur}@jhu.edu

## ABSTRACT

System combination is a technique which has been shown to yield significant gains in speech recognition and machine translation. Most combination schemes perform an *alignment* between different system outputs in order to produce lattices (or *confusion networks*), from which a composite hypothesis is chosen, possibly with the help of a large language model. The benefit of this approach is two-fold: (i) whenever many systems agree with each other on a set of words, the combination output contains these words with high confidence; and (ii) whenever the systems disagree, the language model resolves the ambiguity based on the (probably correct) agreed-upon context. The case of machine translation system combination is more challenging because of the different word orders of the translations: the alignment has to incorporate computationally expensive movements of word blocks. In this paper, we show how one can combine translation outputs efficiently, extending the incremental alignment procedure of [1]. A comparison between different system combination design choices is performed on an Arabic speech translation task.

**Index Terms**— Machine Translation, System Combination, Confusion Networks, Alignments with reordering

## 1. INTRODUCTION

System combination has had a long tradition of being a very useful technique for improving ASR performance. Two approaches, recognizer output voting error reduction (ROVER) [2] and confusion network combination (CNC) [3], are very common in ASR systems, as they are able to combine the strengths of individual systems, or complementary versions of the same system [4]. In both schemes, an alignment of the hypotheses/confusion networks needs to be performed. Since there is no efficient way of doing *multi-string* alignment that optimizes a combined edit metric<sup>1</sup>, various heuristics have been developed for combining more than two sys-

tems together. For instance, in ROVER, a system output is used as the “skeleton”, and all other outputs are aligned with it in a greedy manner, minimizing the average edit distance (between confusion network and new string) at each step. Once all system outputs have been aligned together into a single confusion network, the path with the maximum posterior (with respect to the number of systems outputting each of its words, and the probability assigned by a large language model) is finally selected as the output of the combination.

The above procedures use the output of *monotone* alignments, which are appropriate in speech recognition. On the other hand, *non-monotone* alignments, where blocks of words (or confusion network bins) are allowed to move arbitrarily, are needed in machine translation (MT): it is very common for different MT systems to output translations with different word orders, and monotone alignments, such as the ones minimizing regular edit distance, may not produce meaningful results.

One approach for generating non-monotone alignments is to try to minimize the number of insertions, deletions, substitutions and *block moves* needed to convert one system output to another. The minimum number of such operations is called Translation Error Rate (TER), and its computation is an NP-hard problem [6]. A number of algorithms (e.g., *tercom* [7], or *invWER* [8]) try to approximate TER, but it is unknown if the approximation error can be bounded. *Tercom* is currently used by NIST as the official program for computing TER. Its computational efficiency is based on the constraint that a block of words can be moved only if it does not have a perfect match in its original position, but it has a perfect match in its new position. *invWER*'s complexity is of the order of  $O(k^6)$  (where  $k$  is the length of the longest string to be aligned) and it only allows *nested* block moves; this restriction corresponds to a synchronous parse tree under a simple ITG [9] that has one nonterminal and whose terminal symbols allow insertion, deletion, and substitution. A comparison between *tercom* and *invWER* was done in [10].

System combination with non-monotone alignments has been shown to be a very effective method for improving the quality of machine translation [11, 12, 13, 1, 14], both in

This work was partially supported by the DARPA GALE program (Contract No HR0011-06-2-0001)

<sup>1</sup>Multi-string alignment heuristics are very common in biology for aligning together gene sequences; [5] describes an algorithm that does that.

terms of TER and BLEU<sup>2</sup>. The pairwise alignment algorithm of BBN [13] has been used extensively under the GALE program, as well as in the 2008 NIST MT evaluation. The same BBN group published a more recent *incremental* alignment combination algorithm [1] which has given significant gains compared to the pairwise algorithm. Here, we extend the incremental alignment work of [1] in three directions: (i) we show that it is not necessary to use a word reordering procedure (such as *tercom*) in order to align together the  $N$ -best list of a system; edit distance can be equally effective; (ii) we use a “soft” cost function for confusion bins, instead of the 0/1 cost used in [1]; and (iii) we show how to combine several confusion networks generated from *all* alignment permutations, and select a path from the ensemble.

The paper is organized as follows: Section 2 presents the basic incremental algorithm of [1]; a description of our extensions to that algorithm appears in Section 3; experimental results from Arabic speech translation are presented in Section 4 and concluding remarks are given in Section 5.

## 2. INCREMENTAL CONFUSION NETWORK GENERATION FOR MT

This section describes the algorithm of [1] for confusion network generation. The basic steps of this algorithm are as follows:

1. One system output (usually the 1-best of a system with the best word order or performance) is chosen as the “skeleton”, whose words are used as anchors for aligning all other machine translation outputs together. An initial “confusion network” is thus created, where each “bin” has only one arc with a translation output word as its label.
2. One-by-one, the system outputs get aligned with respect to the confusion network. The cost of aligning a confusion bin with a system output word is 0 if the bin has at least one arc with that word as its label, and 1 otherwise. The alignment is done using *tercom*, which allows the system output words to get re-ordered with respect to the confusion network. The 0/1 cost function is appropriate for the *tercom* constraints on block movements, which include that a block should match perfectly in its new location; here, a match between a word string and “bin string” is defined as a match between the word string and *any* path in the “bin string”.
3. If a word gets *inserted* into the confusion network, a new bin is created with two arcs: one with the inserted word for its label, and one with the special token “NULL” (epsilon/empty transition). If a bin gets “deleted”, then a “NULL” arc is simply inserted into the bin. All other words get inserted into the bins they are aligned to.
4. The cost of each arc is set equal to the negative log (posterior) probability of its label in the bin. Words which come from an  $N$ -best list get discounted as  $1/(m+1)$ , where  $m$  is the rank of the hypothesis contributing the word.
5. The final confusion network is then *rescored* with a 5-gram language model; this results in a re-assignment of arc costs according to a linear combination of the costs (as computed above) and the probability assigned by the language model. The weights in the linear combination are determined through a development set.

One detail of the above procedure which needs to be determined is the order with which the different systems get aligned together. According to [1], the order did not make a big difference in their experiments, but the best results are reported with an order that corresponds to increasing TER on the development set.

## 3. OUR EXTENSIONS

We extended the algorithm of [1] in three ways. Specifically,

1. Since the alignment step using *tercom* is computationally expensive, especially when each system contributes a large enough  $N$ -best list, we defer its use until all  $N$ -best hypotheses of the same system have been aligned together using regular edit distance (Levenshtein cost). One would expect such a scheme not to affect performance significantly, because, usually, the hypotheses in an  $N$ -best list do not differ dramatically from each other,<sup>3</sup> especially the neighboring ones in the list. After the per-system confusion networks are created, they get aligned together, using one of them as the skeleton.
2. Instead of fixing the order with which the different systems get aligned together, as was done in [1], we consider *all* possible permutations of the  $s$  systems and we thus generate  $s!$  confusion networks. Each one of the confusion networks is rescored and weighted appropriately (based on performance on a development set), and the path with the lowest cost *in the ensemble* is finally chosen.
3. Instead of the 0/1 cost function used in [1], that favors alignments of confusion network bins even when they only share one common word, we use the following “soft” cost function

$$c(b_1, b_2) = \frac{1}{|b_1||b_2|} \sum_{w_1 \in b_1} \sum_{w_2 \in b_2} \mathbf{1}(w_1 \neq w_2),$$

which calculates the probability that a word from bin  $b_1$  is not included in bin  $b_2$ .

<sup>2</sup>See [15] for the definition of BLEU.

<sup>3</sup>Note that we only use an  $N$  (e.g., 50) which is typically a small fraction of the total (exponentially large) number of hypotheses at the output of an MT system.

#### 4. EXPERIMENTAL SETUP AND RESULTS

We tested our algorithms for system combination on the development corpora of Arabic speech that were made available under Phase 2 of the GALE program (DEV-07), and we used MT06 broadcast conversations for tuning parameters. Three systems generated translations; a fourth system was also considered for the combination, but its performance was significantly lower than that of the other three, and degraded the combination (on the tuning set) when it was combined with them. For this reason, it was discarded. These systems were used by the Rosetta team in the GALE program.

An ASR system generated the automatic transcriptions that were subsequently translated by the three systems. The ASR system had a cross-adapted architecture between unvoiced and vowelized speaker-adaptive trained (SAT) acoustic models. The distinction between the two comes from the explicit modeling of short vowels, which are pronounced in Arabic but almost never transcribed. Both sets of models were trained discriminatively on approximately 500 hours of supervised data and 2000 hours of unsupervised data. More details about the training of the Arabic models can be found in [16]. The ASR decoder that was used to generate the lattices is described in [17].

The language model used in the rescoring of the confusion networks was a 5-gram modified Kneser-Ney, trained on roughly 180 million words of the English side of the parallel data used in GALE for training the Arabic systems, resulting in roughly 50 million distinct  $n$ -grams. The rescoring also involved a small “deletion” penalty, expressed as an extra cost (usually between 0 and 1) added to the cost of the “NULL” transitions in the confusion networks. This was used in order to control the amount of deleted words in the output.

Two sets of experiments were conducted: (i) In order to understand the impact of various system combination design choices, such as 0/1 vs. “soft” bin cost function, and  $N$ -best list alignment method (Levenshtein vs. *tercom*) for various  $N$ , a number of confusion networks were created, each with the same alignment order to the skeleton. The order was in terms of increasing (TER-BLEU)/2; the system names used throughout the paper (“system 1”, “system 2” and “system 3”) reflect that order. (ii) To see whether the order (in which systems get aligned together) plays a significant role, all system permutations were considered when aligning the  $N$ -best lists together. The resulting confusion networks were then combined according to the following scheme: for each sentence, the id of the permutation that resulted in the best performance was recorded, and the normalized costs (that is, cost divided by the number of words) of all permutation outputs were used to create a feature vector for that segment. Then, a discriminative scheme, Quadratic Discriminative Analysis, was used to learn a model that predicts the id of the permutation based on the features. The model was trained on the tuning set and applied on DEV-07.

	BC		BN	
System/method	TER	BLEU	TER	BLEU
System 1	53.93	25.13	48.54	30.39
System 2	55.33	26.04	49.70	31.31
System 3	54.96	24.60	50.51	29.67
Alignment of 10-best using <i>tercom</i>				
0/1 cost	52.59	25.85	47.23	31.28
“soft” cost	52.93	26.40	47.35	31.59
Alignment of 10-best using Levenshtein dist.				
0/1 cost	52.49	26.02	47.22	31.44
“soft” cost	52.84	26.42	47.33	31.79
Alignment of 20-best using <i>tercom</i>				
0/1 cost	52.48	25.79	47.12	31.40
“soft” cost	52.68	25.73	47.17	31.27
Alignment of 20-best using Levenshtein dist.				
0/1 cost	52.46	26.41	47.28	31.86
“soft” cost	52.46	26.00	<b>46.94</b>	31.53
Alignment of 50-best using <i>tercom</i>				
0/1 cost	52.86	25.48	47.12	31.34
“soft” cost	52.49	26.60	47.37	31.88
Alignment of 50-best using Levenshtein dist.				
0/1 cost	52.49	25.61	47.02	31.66
“soft” cost	52.45	25.97	46.98	31.52
“all-permutations”	<b>52.39</b>	<b>26.86</b>	47.18	<b>32.02</b>
oracle permutation	51.96	27.26	47.16	32.42

**Table 1.** Combination results for Arabic broadcast news and broadcast conversations. The best non-oracle result in each column (minimum for TER and maximum for BLEU) is shown in **bold**.

Table 1 shows the average TER, BLEU of each individual system, as well as of the output of the combinations that resulted with the different competing methods. As can be seen from this table, the “soft” cost function appears more effective than the 0/1 cost for both genres, while *tercom* and Levenshtein alignments appear equally effective. The second-to-last line in the table shows the performance of the permutation experiment described above, while the last line shows the performance of the oracle alignment permutation.

Finally, a number of other experiments, which involve tuning the shift costs, calculating a letter-based similarity between words in the bin cost function, and discounting “NULL” arcs when merging two bins, did not result in significant improvements.

#### 5. CONCLUDING REMARKS

System combination for MT is definitely an area that needs to be explored further, as it consistently offers gains on top of state-of-the-art performance. Confusion network generation using monolingual (target) resources is currently being

used by an increasing number of researchers, in order to combine the strengths of complementary systems. As we demonstrated on an Arabic speech translation task, consistent gains (both in terms of TER and BLEU) can be obtained, as long as the system combined are of comparable quality. We are currently investigating alternative techniques for determining, automatically, which systems are worth combining on a per-segment basis. A cascaded approach, where the systems are first ranked using a diverse set of features, and then the top-ranked ones are selected for confusion network generation and rescoring, is currently underway.

## Acknowledgments

We would like to thank Zhifei Li for his help with making modifications to *tercom*, and Jason Smith for many stimulating discussions. Useful discussions with Lidia Mangu and Antti-Veikko Rosti are gratefully acknowledged. Many thanks go to the Center of Excellence in Human Language Technology at Johns Hopkins University for the availability of their computer cluster.

## 6. REFERENCES

- [1] A.-V.I.Rosti, B. Zhang, S. Matsoukas, and R. Schwartz, "Incremental hypothesis alignment for building confusion networks with application to machine translation system combination," in *Proceedings of the ACL Workshop on Statistical Machine Translation*, Columbus, Ohio, 2008, pp. 183–186.
- [2] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings of IEEE ASRU Workshop*, 1997, pp. 347–352.
- [3] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Proceedings of EuroSpeech*, 1999.
- [4] R. Sinha, M.J.F.Gales, D.Y.Kim, X. Liu, K.C. Sim, and P. Woodland, "The CU-HTK mandarin broadcast news transcription system," in *Proceedings of ICASSP*, Toulouse, 2006.
- [5] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "Clustalw: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–4680, 1994.
- [6] D. Shapira and J. A. Storer, "Edit distance with move operations," in *Proceedings of the 13th Annual Symposium on Combinatorial Pattern Matching*, Fukuoka, Japan, July 2002, vol. 2373/2002, pp. 85–98.
- [7] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, Cambridge, MA, August 2006.
- [8] G. Leusch, N. Ueffing, and H. Ney, "A novel string-to-string distance measure with applications to machine translation evaluation," in *Proceedings of the Machine Translation Summit 2003*, September 2003, pp. 240–247.
- [9] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, September 1997.
- [10] D. Karakos, J. Eisner, S. Khudanpur, and M. Dreyer, "Machine translation system combination using ITG-based alignments," in *Proceedings of the ACL*, Columbus, Ohio, 2008.
- [11] S. Bangalore, G. Bordel, and G. Riccardi, "Computing consensus translation from multiple machine translation systems," in *Proceedings of ASRU*, 2001, pp. 351–354.
- [12] E. Matusov, N. Ueffing, and H. Ney, "Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment," in *Proceedings of EACL*, 2006, pp. 33–40.
- [13] A.-V.I. Rosti, S. Matsoukas, and R. Schwartz, "Improved word-level system combination for machine translation," in *Proceedings of the ACL*, June 2007, pp. 312–319.
- [14] X. He, M. Yang, J. Gao, P. Nguyen, and R. Moore, "Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, Waikiki, Hawaii, October 2008.
- [15] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of 40th Annual Meeting of the ACL (ACL-02)*, 2002, pp. 311–318.
- [16] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, "The IBM 2006 GALE Arabic ASR system," in *Proceedings of ICASSP*, 2007.
- [17] G. Saon, D. Povey, and G. Zweig, "Anatomy of an extremely fast LVCSR decoder," in *Proceedings of Interspeech*, 2005.