Machine Translation System Combination using ITG-based Alignments*

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, Markus Dreyer
Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD 21218
{damianos,eisner,khudanpur,dreyer}@jhu.edu

Abstract

Given several systems' automatic translations of the same sentence, we show how to combine them into a confusion network, whose various paths represent composite translations that could be considered in a subsequent rescoring step. We build our confusion networks using the method of Rosti et al. (2007), but, instead of forming alignments using the *tercom* script (Snover et al., 2006), we create alignments that minimize invWER (Leusch et al., 2003), a form of edit distance that permits properly nested block movements of substrings. Oracle experiments with Chinese newswire and weblog translations show that our confusion networks contain paths which are significantly better (in terms of BLEU and TER) than those in *tercom*-based confusion networks.

1 Introduction

Large improvements in machine translation (MT) may result from combining different approaches to MT with mutually complementary strengths. System-level combination of translation outputs is a promising path towards such improvements. Yet there are some significant hurdles in this path. One must somehow align the multiple outputs—to identify where different hypotheses reinforce each other and where they offer alternatives. One must then use this alignment to hypothesize a set of new, composite translations, and select the *best* composite hypothesis from this set. The alignment step is difficult because different MT approaches usually reorder the translated words differently. Training the selection step is difficult because identifying the best hypothesis (relative to a known reference translation) means scoring all the composite hypotheses, of which there may be exponentially many.

Most MT combination methods do create an exponentially large hypothesis set, representing it as a *confusion network* of strings in the target language (e.g., English). (A confusion network is a lattice where every node is on every path; i.e., each time step presents an *independent* choice among several phrases. Note that our contributions in this paper could be applied to arbitrary lattice topologies.) For example, Bangalore et al. (2001) show how to build a confusion network following a *multistring* alignment procedure of several MT outputs. The procedure (used primarily in biology, (Thompson et al., 1994)) yields monotone alignments that minimize the number of insertions, deletions, and substitutions. Unfortunately, monotone alignments are often poor, since machine translations (particularly from different models) can vary significantly in their word order. Thus, when Matusov et al. (2006) use this procedure, they deterministically reorder each translation prior to the monotone alignment.

The procedure described by Rosti et al. (2007) has been shown to yield significant improvements in translation quality, and uses an estimate of *Translation Error Rate* (TER) to guide the alignment. (TER is defined as the *minimum* number of inser-

*This work was partially supported by the DARPA GALE program (Contract No HR0011-06-2-0001). Also, we would like to thank the IBM Rosetta team for the availability of several MT system outputs.

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 81–84,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

tions, deletions, substitutions and *block shifts* between two strings.) A remarkable feature of that procedure is that it performs the alignment of the output translations (i) without any knowledge of the translation model used to generate the translations, and (ii) without any knowledge of how the target words in each translation align back to the source words. In fact, it only requires a procedure for creating pairwise alignments of translations that allow appropriate re-orderings. For this, Rosti et al. (2007) use the *tercom* script (Snover et al., 2006), which uses a number of heuristics (as well as dynamic programming) for finding a sequence of edits (insertions, deletions, substitutions and block shifts) that convert an input string to another. In this paper, we show that one can build *better* confusion networks (in terms of the *best* translation possible from the confusion network) when the pairwise alignments are computed not by *tercom*, which *approximately* minimizes TER, but instead by an *exact* minimization of *invWER* (Leusch et al., 2003), which is a restricted version of TER that permits only properly nested sets of block shifts, and can be computed in polynomial time.

The paper is organized as follows: a summary of TER, *tercom*, and invWER, is presented in Section 2. The system combination procedure is summarized in Section 3, while experimental (oracle) results are presented in Section 4. Conclusions are given in Section 5.

2 Comparing *tercom* and invWER

The *tercom* script was created mainly in order to measure translation quality based on TER. As is proved by Shapira and Storer (2002), computation of TER is an NP-complete problem. For this reason, *tercom* uses some heuristics in order to compute *an approximation to TER* in polynomial time. In the rest of the paper, we will denote this approximation as *tercomTER*, to distinguish it from (the intractable) TER. The block shifts which are allowed in *tercom* have to adhere to the following constraints: (i) A block that has an exact match cannot be moved, and (ii) for a block to be moved, it should have an *exact* match in its new position. However, this sometimes leads to counter-intuitive sequences of edits; for instance, for the sentence pair

“thomas jefferson says eat your vegetables”
“eat your cereal thomas edison says”,

tercom finds an edit sequence of cost 5, instead of the optimum 3. Furthermore, the block selection is done in a greedy manner, and the final outcome is dependent on the shift order, even when the above constraints are imposed.

An alternative to *tercom*, considered in this paper, is to use the Inversion Transduction Grammar (ITG) formalism (Wu, 1997) which allows one to view the problem of alignment as a problem of bilingual parsing. Specifically, ITGs can be used to find the optimal edit sequence under the restriction that block moves must be properly nested, like parentheses. That is, if an edit sequence swaps adjacent substrings A and B of the original string, then any other block move that affects A (or B) must stay completely within A (or B). An edit sequence with this restriction corresponds to a synchronous parse tree under a simple ITG that has one nonterminal and whose terminal symbols allow insertion, deletion, and substitution.

The minimum-cost ITG tree can be found by dynamic programming. This leads to *invWER* (Leusch et al., 2003), which is defined as the minimum number of edits (insertions, deletions, substitutions and block shifts allowed by the ITG) needed to convert one string to another. In this paper, the minimum-invWER alignments are used for generating confusion networks. The alignments are found with a 11-rule Dyna program (Dyna is an environment that facilitates the development of dynamic programs—see (Eisner et al., 2005) for more details). This program was further sped up (by about a factor of 2) with an A^* search heuristic computed by additional code. Specifically, our admissible outside heuristic for aligning two substrings estimated the cost of aligning the words *outside* those substrings as if reordering those words were free. This was complicated somewhat by type/token issues and by the fact that we were aligning (possibly weighted) lattices. Moreover, the *same* Dyna program was used for the computation of the minimum invWER path in these confusion networks (oracle path), without having to invoke *tercom* numerous times to compute the best sentence in an N-best list.

The two competing alignment procedures were

Lang. / Genre	tercomTER	invWER
Arabic NW	15.1%	14.9%
Arabic WB	26.0%	25.8%
Chinese NW	26.1%	25.6%
Chinese WB	30.9%	30.4%

Table 1: Comparison of average per-document tercomTER with invWER on the EVAL07 GALE Newswire ("NW") and Weblogs ("WB") data sets.

Genre	*CNs with tercom*	CNs with ITG
NW	50.1% (27.7%)	**48.8% (28.3%)**
WB	51.0% (25.5%)	**50.5% (26.0%)**

Table 2: TercomTERs of invWER-oracles and (in parentheses) oracle BLEU scores of confusion networks generated with *tercom* and ITG alignments. The best results per row are shown in bold.

used to *estimate* the TER between machine translation system outputs and reference translations. Table 1 shows the TER estimates using *tercom* and invWER. These were computed on the translations submitted by a system to NIST for the GALE evaluation in June 2007. The references used are the post-edited translations for that system (i.e., these are "HTER" approximations). As can be seen from the table, in *all* language and genre conditions, invWER gives a *better* approximation to TER than tercomTER. In fact, out of the roughly 2000 total segments in all languages/genres, tercomTER gives a lower number of edits in only 8 cases! This is a clear indication that ITGs can explore the space of string permutations more effectively than *tercom*.

3 The System Combination Approach

ITG-based alignments and *tercom*-based alignments were also compared in oracle experiments involving confusion networks created through the algorithm of Rosti et al. (2007). The algorithm entails the following steps:

- Computation of all pairwise alignments between system hypotheses (either using ITGs or *tercom*); for each pair, one of the hypotheses plays the role of the "reference".
- Selection of a system output as the "skeleton" of the confusion network, whose words are used as anchors for aligning all other machine translation outputs together. Each arc has a translation output word as its label, with the special token "NULL" used to denote an insertion/deletion between the skeleton and another system output.
- Multiple consecutive words which are inserted relative to the skeleton form a phrase that gets aligned with an *epsilon* arc of the confusion network.
- Setting the weight of each arc equal to the negative log (posterior) probability of its label; this probability is proportional to the number of systems which output the word that gets aligned in that location. Note that the algorithm of Rosti et al. (2007) used N-best lists in the combination. Instead, we used the single-best output of each system; this was done because not all systems were providing N-best lists, and an unbalanced inclusion would favor some systems much more than others. Furthermore, for each genre, one of our MT systems was significantly better than the others in terms of word order, and it was chosen as the skeleton.

4 Experimental Results

Table 2 shows tercomTERs of invWER-oracles (as computed by the aforementioned Dyna program) and oracle BLEU scores of the confusion networks. The confusion networks were generated using 9 MT systems applied to the Chinese GALE 2007 Dev set, which consists of roughly 550 Newswire segments, and 650 Weblog segments. The confusion networks which were generated with the ITG-based alignments gave significantly better oracle tercomTERs (significance tested with a Fisher sign test, $p - 0.02$) and better oracle BLEU scores. The BLEU oracle sentences were found using the dynamic-programming algorithm given in Dreyer et al. (2007) and measured using Philipp Koehn's evaluation script. On the other hand, a comparison between the 1-best paths did not reveal significant differences that would favor one approach or the other (either in terms of tercomTER or BLEU).

We also tried to understand which alignment method gives higher probability to paths “close” to the corresponding oracle. To do that, we computed the probability that a random path from a confusion network is within x edits from its oracle. This computation was done efficiently using finite-state-machine operations, and did not involve any randomization. Preliminary experiments with the invWER-oracles show that the probability of all paths which are within $x = 3$ edits from the oracle is roughly the same for ITG-based and tercom-based confusion networks. We plan to report our findings for a whole range of x-values in future work. Finally, a runtime comparison of the two techniques shows that ITGs are much more computationally intensive: on average, ITG-based alignments took 1.5 hours/sentence (owing to their $O(n^6)$ complexity), while *tercom*-based alignments only took 0.4 sec/sentence.

5 Concluding Remarks

We compared alignments obtained using the widely used program *tercom* with alignments obtained with ITGs and we established that the ITG alignments are superior in two ways. Specifically: (a) we showed that invWER (computed using the ITG alignments) gives a better approximation to TER between machine translation outputs and human references than *tercom*; and (b) in an oracle system combination experiment, we found that confusion networks generated with ITG alignments contain better oracles, both in terms of tercomTER and in terms of BLEU.

Future work will include rescoring results with a language model, as well as exploration of heuristics (e.g., allowing only “short” block moves) that can reduce the ITG alignment complexity to $O(n^4)$.

References

S. Bangalore, G. Bordel, and G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proceedings of ASRU*, pages 351–354.

M. Dreyer, K. Hall, and S. Khudanpur. 2007. Comparing reordering constraints for smt using efficient bleu oracle computation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 103–110, Rochester, New York, April. Association for Computational Linguistics.

Jason Eisner, Eric Goldlust, and Noah A. Smith. 2005. Compiling comp ling: Weighted dynamic programming and the Dyna language. In *Proceedings of HLT-EMNLP*, pages 281–290. Association for Computational Linguistics, October.

G. Leusch, N. Ueffing, and H. Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proceedings of the Machine Translation Summit 2003,* pages 240–247, September.

E. Matusov, N. Ueffing, and H. Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proceedings of EACL*, pages 33–40.

A.-V.I. Rosti, S. Matsoukas, and R. Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of the ACL*, pages 312–319, June.

D. Shapira and J. A. Storer. 2002. Edit distance with move operations. In *Proceedings of the 13th Annual Symposium on Combinatorial Pattern Matching*, volume 2373/2002, pages 85–98, Fukuoka, Japan, July.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, Cambridge, MA, August.

J. D. Thompson, D. G. Higgins, and T. J. Gibson. 1994. Clustalw: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, September.