Combining Many Alignments for Speech to Speech Translation

Sameer R. Maskey, Steven J. Rennie, Bowen Zhou

IBM T. J. Watson Research Center, Yorktown Heights, NY

{smaskey, sjrennie, zhou}@us.ibm.com

Abstract

Alignment combination (symmetrization) has been shown to be useful for improving Machine Translation (MT) models. Most existing alignment combination techniques are based on heuristics, and can combine only two sets of alignments at a time. Recently in [1], we proposed a power mean based algorithm that can be optimized to combine an arbitrary number alignment tables simultaneously. In this paper we present an empirical investigation of the merits of the approach for combining a large number of alignments (more than 200 in total before pruning). The results of the study suggest that the algorithm can often improve the performance of speech to speech translation systems for low resource languages.

1. Introduction

One of the crucial steps involved in building a Machine Translation (MT) system is obtaining alignments between source and target words of bi-text parallel corpus. A typical alignment algorithm finds links between source and target words using some variation of Expectation Maximization (EM) algorithm. There have been numerous such algorithms proposed, including [2], [3], [4], [5]. The estimated alignment pairs are used to build the core models of the MT engine, such as phrase tables [6], hierarchical rules [7], or tree-to-string mappings [8]. Thus, it is crucial that the estimated alignment links are as accurate as possible.

One common technique for improving alignment accuracy is to estimate (one-to-many) alignment tables in both the source-to-target ($E2F$) and target-to-source ($F2E$) directions, and then combine them [3]. Such methods involve taking two sets of alignment tables A_1 and A_2 for the same sentence pair, $E - F$, and producing a new set A_o. How to find the optimal A_o is the key question. The intersection of these tables, $A_\cup = A_1 \cup A_2$, for example, has high precision but low recall and produces fewer alignments, while their intersection, $A_\cap = A_1 \cap A_2$, has high recall but low precision. Various heuristic methods for estimating A_o have been proposed ([3], [6]). The method presented in [3], for example, interpolates between the intersection and union of two asymmetric alignment tables by adding links that are adjacent to intersection links, and connect at least one previously unaligned word. Another example is the method in [6], which adds links to the intersection of two alignment tables that are the diagonal neighbors of existing links, optionally requiring that any added links connect two previously unaligned words.

Other methods for improving alignment accuracy have focused on using bi-directional information during the alignment training process. In [9], asymmetric models are jointly trained to maximize the similarity of their alignments by iteratively optimizing an objective function based on agreement heuristics. In [10], the authors present a technique for combining alignments based on various linguistic resources such as parts of speech, dependency parses, or bilingual dictionaries, and use machine learning techniques to do alignment combination.

Recently, [11] presented a method for combining two alignment tables that is effective and relies minimally on heuristics during the combination process. [12] extended this algorithm by integrating confidence scores into the framework of [11], and further showed that combining more than 2 alignments can be useful. Recently, [1] introduced a power mean based algorithm for alignment combination. The method not only avoids the use of heuristics, but can also simultaneously combine an arbitrary number of alignment tables, and has parameters that can be used to optimize any chosen objective function. The power mean is defined by equation (1) below, where p is a real number in $(-\infty, \infty)$.

$$S_p(a_1, a_2, ..., a_n) = (\frac{1}{n}\sum_{k=1}^{n} a_k^p)^{\frac{1}{p}} \quad (1)$$

[1] showed that as $p \to 0$, the combination process is equivalent to the logical intersection of the input alignments when the alignments are represented as binary variable tables, and as $p \to \infty$, the combination result is equivalent to the union of the input alignments. The power mean is therefore a principled way to interpolate between these extremes. In this paper, we empirically investigate the merit of generating large numbers of different alignments and combining them using the power mean algorithm as presented by [1].

2. Data

We performed all of our experiments on English-Pashto data provided by DARPA for TRANSTAC (Spoken Language Communication and Translation System for Tactical Use) Evaluation of 2010. The TRANSTAC task was designed to evaluate speech-to-speech translation systems, so all training sentences are conversational in nature. Hence, the data consists of labels that mark each sentence as E2F, F2E, EIF or FIE where E2F is Pashto translated from English speaker and FIE is translation of Interpreter's Pashto speech. 2026 sentences were randomly sampled from this training data to prepare held out development set (dev). The heldout test set (test1) consisted of 1019 parallel sentences with 1 reference. Our second held out test set (test2) consisted of 564 sentences with 4 references. We also had 150 sentences manually aligned sentences which were used to tune the parameters of power mean algorithm.

3. Experiments

The alignment between source and target sentence pairs that are estimated during training naturally depends on several factors including: (i) the statistical training algorithm, (ii) the parameters of the trainer, (iii) the heuristics used for directional combination, and (iv) any data preprocessing. Different systems typically produce different alignments, and there are an endless number of such systems, so as usual we must resort to an approximate search for the system that outputs the “best” set of alignments. In this paper we investigate the merit of interpolating over many alignments to, in essence, approximate the expected value of the alignments when averaged over all possible alignment generation systems, using the power mean algorithm [1].

3.1. Generating sets of alignments

The first task is to generate a large set of alignments by varying the properties of the aligner, as alluded to above. In the following we describe how we generated a large number of alignments (200) as candidate alignment“exemplars” by varying the following properties of the aligner: (i) The Alignment algorithm, (ii) Data Preprocessing, (iii) Symmetrization algorithm. How these properties were varied is described in the following three subsections.

3.2. Alignment algorithms

Alignments were generated using GIZA++ algorithm [13] and an HMM aligner similar to the one proposed in [4]. The GIZA++ uses an EM algorithm based on the IBM Models [2] to generate alignments that maximize the log likelihood of $\hat{\theta} = \arg\max_{\theta} \prod_{s=1}^{S} \sum_{a} p_{\theta}(f_s, a|e_s)$, where the form of the model is further restricted so that the most likely alignment a under the model can be determined using dynamic programming. The HMM aligner [4] uses a different model topology than GIZA++. In this work we will prefix alignments generated by these systems with a b (GIZA++) and h (HMM), respectively. In addition, we distinguish between alignments that require the final word to be covered (1) or uncovered (2). For example an alignment with prefix $b2$ is generated using Giza++ with final word covered.

3.3. Data Preprocessing

Pashto is morphologically rich language with many prefixes and suffixes. In lack of a morphological segmenter it has been suggested that keeping only first ‘n’ characters of a word can effectively reduce the vocabulary size and may produce better alignments [14]. We trained such alignments using using GIZA++ on parallel data with partial words for Pashto sentences. In fact, we produced many partial word alignments based on the length of partial words we kept in preprocessing stage i.e we built alignments based partial words (PA) of lengths $n = 2, 3, 4, 5, 6, 7$. In this work, alignments generated using partial words are tagged with fwn, where n is PA length. For example, alignments tagged with $fw3$ are built using foreign partial words of length 3.

3.4. Direction based Symmetrization

Symmetrized alignments, based on combining alignments that were generated in the two E2F and F2E directions have been widely used. Besides the basic methods of Intersection (I) and Union (U), several heuristic methods have been proposed and some are widely used [15, 16]. The method presented in [11] relies less on heuristics and has been shown to perform well. Here we consider all of the aforementioned symmetrization methods,and tag alignments generated with them with the following acronyms: Intersection (I), Union (U), Grow Diagonal Final (GDF) [16], refined Heuristics (H) [15], optimal phrase pair heuristics (O) [11].

3.5. Summary of alignment naming convention

In summary, the naming convention we use for each of these alignment is of the form: **<algorithm type><final word uncovered>[.data preprocessing].<symmetrization method>**. For example, b1.fw4.H means that the alignment was generated from the baseline alignment algorithm (GIZA++) with final words uncovered, foreign partial words of length four, and refined heuristics for direction combination.

Precision	Recall	F-measure	Alignment Type
0.8519	0.6298	0.7242	b1.fw6.H
0.8604	0.6222	0.7222	b1.fw4.H
0.8453	0.6261	0.7194	b1.fw7.H
...			
0.8349	0.3751	0.5176	h1.I
0.8651	0.3556	0.5041	b1.fw2.I
0.8071	0.3328	0.4712	h2.I

Table 1: E2F : F-measure Base Alignments

3.6. Alignment prepruning

By varying the alignment generation parameters discussed above, 200 different alignments were generated. Tables 1 and 2 summarize the F-scores of the top 3 and bottom 3 alignments w.r.t. 150 human annotated sentences, for the F2E and E2F directions (obtained by bidirectional combination), respectively.

Precision	Recall	F-measure	Alignment Type
0.8603	0.6157	0.7177	b1.fw4.H
0.8043	0.6436	0.7150	b1.fw4.GDF
0.7919	0.6461	0.7116	b1.fw4.U
...			
0.8256	0.3719	0.5128	h1.I
0.8729	0.3442	0.4937	b1.fw2.I
0.7920	0.3223	0.4582	h2.I

Table 2: F2E : F-measure Base Alignments

We observe in Table 2 that in the F2E direction, the best F-measures result from using the baseline aligner with foreign partial words. In contrast, the HMM aligner with final word covered and direction combination based on intersection produced the worst F-measure. In general, HMM-based alignments seem to do worse on our data set. For efficiency, in this work these alignments were pre-pruned down to a small number (20 or less) based on their F-measure w.r.t. annotated alignment data before combining them.

For E2F weighted data we again observe that the partial word based alignments perform the best. Unlike F2E direction, partial words with $n = 6$ seems to do better. HMM-based alignments again seem to do worse overall in F2E direction as well. After getting all these individual alignments we picked the best set of alignments and combined them using [1] algorithm. Before we combined these individual alignments we were also able to obtain the combined alignments as produced by [12]. [12] combines four different alignments based on segmentation, partial words with $n = 5$, verb reordered sentences, and baseline alignment produced by GIZA++, using the technique described in [12]. We also used their verb reordered alignment as another set of individual alignment that we can combine with. There are no restrictions on the number of alignments or quality of the alignments when combining them using the power mean algorithm [1].

3.7. Alignment combination using the power mean

To investigate the performance of the power mean algorithm as a function of number of alignments combined, alignments were chosen from Table 1 and 2 based on their F-measure scores, and their diversity relative to already chosen alignments, in greedy fashion. Better scoring alignments were naturally added first.

Combination	Fmeasure
comb.11	0.7545
comb.10	0.7543
comb.5	0.7534
comb.7	0.7532
comb.3	0.7522
comb.5	0.7504
comb.4	0.7502
comb.4b	0.7497
comb.2	0.7451
comb.10b	0.7451
comb.10c	0.7451
comb.3b	0.7444
comb.2b	0.7444
comb.17	0.7439
comb.3c	0.7435

Table 3: E2F : F-measure, Combined Alignments

Using the power mean algorithm, which is fully described in [1], each set of alignments was combined to maximize the F-measure of the output alignment on 150 sentences of hand aligned development data. As in [1], while doing so we also optimized table weights $W_q \in (0,\, 1), \sum_q W_q = 1$, which were applied to the alignment tables before combining them using the power mean. The W_q allow the algorithm to weight alignments differently. As in [1] we found that the F-measure function had many local minima so the simplex algorithm was initialized at several values of p and $\{W_q\}$ to find the globally optimal F-measure.

3.8. F-Score Results

F-score results as a function of the number of alignments included in the combination process are shown in Table 3 for the E2F direction. "comb.Number" in the table stands for the number of alignments that were combined where each alignment was already combined for direction. If there are more than two sets with same of number of alignment we identify it by adding letters 'a, b or c' to the "comb.Number." The "comb.11" result, which combined 11 different alignments, performed the best with an F-measure of 0.7545 which is better by 3.08% than the best individual alignment. Similarly in the F2E direction, the best combined alignment was better by 3.43% than the best individual alignment set. Significant improvements on both E2F and F2E direction show that combining multiple alignments instead of combining alignments based only on direction produces better alignments. From the results we can see that using more alignments during combination does not necessarily translate to higher F-measure. This could be due to overfitting of the data as more parameters are added to the combination process, and/or the identification of locally optimal parameters w.r.t. to the objective function (F-measure), which was observed to have many local minima. Regularization of the table weights with a sparseness term in the objective function in particular will probably produce more consistent improvements.

3.9. BLEU Score Results

In the previous section alignment F-measure was optimized using the power mean algorithm in the hope that increases in F-measure would lead to some improvement in overall MT quality with respect to BLEU scores. However, how well alignment F-measures actually correlate with BLEU scores is an open question, as explained in [17]. While there is no mathematical problem with optimizing the parameters of the presented PM-based combination algorithm w.r.t. BLEU scores, computationally it is not practical to do so because each iteration would require a complete training phase. To further evaluate the quality of the alignments and the combination method, we built several MT models based on them and compared the resulting BLEU scores.

Combination	Fmeasure
comb.11	0.752
comb.5	0.7516
comb.14	0.7516
comb.11	0.7513
comb.3	0.7496
comb.3b	0.7496
comb.4	0.7496
comb.7	0.7496
comb.16	0.7488
comb.2	0.7486
comb.2b	0.7486

Table 4: F2E : F-measure, Combined Alignments

Method	Dev	Test1	Test 2
alignComb.17	0.1585	0.1542	0.2769
alignComb.10	0.1580	0.1507	0.2713
alignComb.2	0.1559	0.1537	0.2708
alignComb.10b	0.1570	0.1548	0.2671
alignComb.4	0.1535	0.1530	0.2660
alignComb.11	0.1519	0.1528	0.2657
alignComb.3	0.1549	0.1504	0.2655
alignComb.3b	0.1553	0.1527	0.2642
alignComb.3c	0.1527	0.1575	0.2639
alignComb.5	0.1537	0.1514	0.2600
O68	0.1507	0.1461	0.2551
alignComb.7	0.1365	0.1341	0.2537
alignComb.10c	0.1534	0.1517	0.2523
alignComb.5	0.1351	0.1352	0.2486
alignComb.4b	0.1437	0.1373	0.2463

Table 5: E2F : BLEU Scores for Combined Alignments

The MT models we built were trained on the bi-text corpora described in Section 2. We built a phrase based translation system with a phrase length of 6 for English and 8 for Pashto. We trained the lexicalized reordering model that produced distortion costs based on the number of words that are skipped on the target side, in a manner similar to [18]. We had significant amount of out of domain English sentences (1.4 million) that we interpolated with in-domain data to produce an English language model. For the Pashto LM, we simply used the Pashto side of bilingual corpus. MT models were trained using minimum error rate training [19] with a stack based decoder that uses an A^* search.

We can see in Table 6 that "comb.17", which combines 17 different alignments, has the best BLEU scores with respect to dev set, test1 and test2. For our baseline (O68), we combined alignments based on the combination algorithm proposed by [11]. We observe that comb.17 is better than baseline method by 2.18 BLEU score on test2 with 4 references. It is also better on test1 by 0.81 and on dev set by 0.77 BLEU. We can see from Table 5 that many of the alignments combined with 3 or more tables are better than the baseline without the combination using power mean algorithm. We should note, however, that the highest F-measure combined alignment did not rank the highest in BLEU scores. There could be several reasons for such results. First, trying to optimize the weights and p-value of too many alignments with only 150 sentences of human labeled data could be leading to data overfitting. Second, the F-measure objective may have more local optima w.r.t. the parameters of the power mean algorithm when more alignments are combined, since there are more parameters to optimize.

Method	Dev	Test1	Test 2
alignComb.4	0.1786	0.1795	0.3375
alignComb.2	0.1806	0.1796	0.3359
O68	0.1821	0.1797	0.3346
alignComb.2b	0.1800	0.1797	0.3346
alignComb.5	0.1783	0.1797	0.3313
alignComb.14	0.1793	0.1805	0.3310
alignComb.11	0.1790	0.1797	0.3303
alignComb.3b	0.1767	0.1744	0.3294
alignComb.11	0.1788	0.1790	0.3280
alignComb.16	0.1781	0.1811	0.3273
alignComb.7	0.1803	0.1801	0.3243
alignComb.3	0.1744	0.1733	0.3171

Table 6: F2E : BLEU Scores for Combined Alignments

Although we saw significant gains on E2F direction we did not similar gains on F2E direction similar to observations made by [1]. One possible explanation for such results is that the Pashto LM for the E2F direction is trained on sentences from available training bi-text corpus while English LM for F2E direction was trained on 1.4 million sentences. Therefore the English LM, which is trained on significantly more data, is probably more robust to errors made by the translation and reordering models.

We also observed that combining many alignments not only produced better alignments but also in general reduces phrase table size. Intersection that has the least number of alignments tend to produce the largest phrase table while Union tends to produce the least number of phrases because phrase extraction algorithm has more constraints to satisfy. The PT size produced by PM$_n$ for our 17 different alignments combined is between I and U and is 19.2% smaller than the baseline model suggesting that we are improving the MT model but reducing its size.

4. Conclusion and Future Work

In this paper we presented empirical results that suggest that combining many alignments with the power mean produces better MT models, particularly for low resource languages, which are difficult to build strong language models for. In particular, we showed that existing techniques for generating alignments, symmetrizing alignments, and preprocessing data can be used to generate a large number of alignments for subsequent combination using the power mean algorithm. The combined alignments resulted in significant gains in terms of both BLEU and F-measure for English to Pashto speech-to-speech translation.

5. Acknowledgement

This work is partially supported by the Darpa Transtac program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Darpa.

6. References

[1] S. Maskey, S. Rennie, and B. Zhou, “Power mean based algorithm for combining multiple alignment tables,” in *Proceedings of COLING*, 2010.

[2] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer, “The mathematics of statistical machine translation: parameter estimation,” *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[3] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[4] S. Vogel, H. Ney, and C. Tillmann, “Hmm-based word alignment in statistical translation,” in *COLING 96: The 16th Int. Conf. on Computational Linguistics*, 1996.

[5] E. Matusov, R. Zens, and H. Ney, “Symmetric word alignments for statistical machine translation,” in *Proceedings of COLING*, Morristown, NJ, USA, 2004, p. 219.

[6] P. Koehn, F. J. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proceedings of HLT/NAACL*, 2003.

[7] D. Chiang, “A hierarchical phrase-based model for statistical machine translation,” in *Proceedings of ACL*, 2005.

[8] K. Yamada and K. Knight, “A syntax-based statistical translation model,” in *Proceedings of ACL*. Toulouse, France: ACL, July 2001, pp. 523–530.

[9] P. Liang, B. Taskar, and D. Klein, “Alignment by agreement,” in *Proceedings of ACL*, 2006.

[10] N. Ayan, B. J. Dorr, and N. Habash, “Multi-align: Combining linguistic and statistical techniques to improve alignments for adaptable mt,” in *Proceedings of AMTA*, 2004.

[11] Y. Deng and B. Zhou, “Optimizing word alignment combination for phrase table training,” in *Proceedings of ACL, Short Papers*, 2009.

[12] B. Xiang, Y. Deng, and B. Zhou, “Diversify and combine: Improving word alignment for machine translation on low-resource languages,” in *Proceedings of ACL*, 2010.

[13] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[14] D. Chiang, K. Knight, and S. Echihabi, in *Presentation at NIST MT 2009 Workshop, August.*, 2009.

[15] F. J. Och and D. Marcu, “Statistical phrase-based translation,” in *HLT*, 2003, pp. 127–133.

[16] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proc. NAACL/HLT*, 2003.

[17] A. Fraser and D. Marcu, “Measuring word alignment quality for statistical machine translation,” *Comput. Linguist.*, vol. 33, no. 3, pp. 293–303, 2007.

[18] Y. Al-Onaizan and K. Papineni, “Distortion models for statistical machine translation,” in *Proceedings of ACL*, 2006.

[19] F. J. Och, “Minimum error rate training in statistical machine,” in *Proceedings of ACL*, 2003.