A Maximum Entropy Approach to Combining Word Alignments

Necip Fazil Ayan and **Bonnie J. Dorr**
Institute of Advanced Computer Studies (UMIACS)
University of Maryland
College Park, MD 20742
{nfa,bonnie}@umiacs.umd.edu

Abstract

This paper presents a new approach to combining outputs of existing word alignment systems. Each alignment link is represented with a set of feature functions extracted from linguistic features and input alignments. These features are used as the basis of alignment decisions made by a maximum entropy approach. The learning method has been evaluated on three language pairs, yielding significant improvements over input alignments and three heuristic combination methods. The impact of word alignment on MT quality is investigated, using a phrase-based MT system.

1 Introduction

Word alignment—detection of corresponding words between two sentences that are translations of each other—is usually an intermediate step of statistical machine translation (MT) (Brown et al., 1993; Och and Ney, 2003; Koehn et al., 2003), but also has been shown useful for other applications such as construction of bilingual lexicons, word-sense disambiguation, projection of resources, and cross-language information retrieval.

Maximum entropy (ME) models have been used in bilingual sense disambiguation, word reordering, and sentence segmentation (Berger et al., 1996), parsing, POS tagging and PP attachment (Ratnaparkhi, 1998), machine translation (Och and Ney, 2002), and FrameNet classification (Fleischman et al., 2003). They have also been used to solve the word alignment problem (Garcia-Varea et al., 2002; Ittycheriah and Roukos, 2005; Liu et al., 2005), but a sentence-level approach to combining knowledge sources is used rather than a word-level approach.

This paper describes an approach to combining evidence from alignments generated by existing systems to obtain an alignment that is closer to the true alignment than the individual alignments. The alignment-combination approach (called *ACME*) operates at the level of alignment links, rather than at the sentence level (as in previous ME approaches). ACME uses ME to decide whether to include/exclude a particular alignment link based on feature functions that are extracted from the input alignments and linguistic features of the words. Since alignment combination relies on evidence from existing alignments, we focus on alignment links that exist in at least one input alignment. An important challenge in this approach is the selection of appropriate links when two aligners make different alignment choices.

We show that ACME yields a significant relative error reduction over the input alignment systems and heuristic-based combinations on three different language pairs. Using a higher number of input alignments and partitioning the training data into disjoint subsets yield further error-rate reductions.

The next section briefly overviews ME models. Section 3 presents a new ME approach to combining existing word alignment systems. Section 4 describes the evaluation data, input alignments, and evaluation metrics. Section 5 presents experiments on three language pairs, upper bounds for alignment error rate in alignment combination, and MT evaluation on English-Chinese and English-Arabic. Section 6 describes previous work on alignment combination and ME models on word alignment.

2 Maximum Entropy (ME) Models

In a statistical classification problem, the goal is to estimate the probability of a class y in a given context x, i.e., $p(y|x)$. In an ideal scenario, if the training data contain evidence for all pairs of (y, x), it is

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 96–103,
New York, June 2006.

trivial to compute the probability distribution p. Unfortunately, due to training-data sparsity, p is generally modeled using only the available evidence.

Given a collection of facts, ME chooses a model consistent with all the facts, but otherwise as uniform as possible (Berger et al., 1996). Formally, the evidence is represented as feature functions, i.e., binary valued functions that map a class y and a context x to either 0 or 1, i.e., $h_m : \mathcal{Y} \times \mathcal{X} \rightarrow \{0, 1\}$, where $\mathcal{Y}$ is the set of all classes and $\mathcal{X}$ is the set of all facts. The biggest advantage of maximum entropy models is that they are able to focus on the selection of feature functions rather than on how such functions are used. Any context can be used to define feature functions without concern for the independence of the feature functions from each other or the relevance of the feature functions to the final decision (Ratnaparkhi, 1998).

Each feature function h_m is associated with a model parameter λ_m. Given a set of M feature functions $h_1, \ldots, h_M$, the probability of class y given a context x is equal to:

$$p(y|x) = \frac{1}{Z_x} exp\left(\sum_{m=1}^{M} \lambda_m h_m(y, x)\right)$$

where Z_x is a normalization constant. The contribution of each feature function to the final decision, i.e., λ_m, can be automatically computed using Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff, 1972). The final classification for a given instance is the class y that maximizes $p(y|x)$.

3 Alignment Combination: ACME

Let $\mathbf{e} = e_1, \ldots, e_I$ and $\mathbf{f} = f_1, \ldots, f_J$ be two sentences in two different languages. An alignment link (i, j) corresponds to a translational equivalence between words e_i and f_j. Let A_k be an alignment between sentences $\mathbf{e}$ and $\mathbf{f}$, where each element $a \in A_k$ is an alignment link (i, j). Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be a set of alignments between $\mathbf{e}$ and $\mathbf{f}$. We refer to the true alignment as T, where each $a \in T$ is of the form (i, j). The goal of ACME is to combine the information in $\mathcal{A}$ such that the combined alignment A_C is closer to T. A straightforward solution is to take the intersection or union of the individual alignments. In this paper, an additional model is learned to combine outputs of $A_1, \ldots, A_n$.

In our combination framework, first, n different word-alignment systems, $A_1, \ldots, A_n$, generate word alignments between a given English sentence and a foreign-language (FL) sentence. Then a *Feature Extractor* takes the output of these alignment systems and the parallel corpus (which might be enriched with linguistic features) and extracts a set of feature functions based on linguistic properties of the words and the input alignments. Each feature function h_m is associated with a model parameter λ_m. Next, an *Alignment Combiner* decides whether to include or exclude an alignment link based on the extracted feature functions and the model parameters associated with them.

For each possible alignment link a set of features is extracted from the input alignments and linguistic properties of words. The features that are used for representing an alignment link (i, j) are as follows:

1. **Part-of-speech tags (*posE, posF, prevposE, prevposF, nextpostE, nextposF*):** POS tags for the previous, current, and the next English and FL words.
2. **Outputs of input aligners (*out*):** Whether (i, j) exists in a given input alignment A_k.
3. **Neighbors (*neigh*):** A *neighborhood* of an alignment link (i, j)—denoted by $N(i, j)$—consists of 8 possible alignment links in a 3×3 window with (i, j) in the center of the window. Each element of $N(i, j)$ is called a *neighboring link* of (i, j). Neighbor features include: (1) Whether a particular neighbor of (i, j) exists in a given input alignment A_k; and (2) Total number of neighbors of (i, j) in a given input alignment A_k.
4. **Fertilities (*fertE, fertF*):** The number of words that e_i (or f_j) is aligned to in a given input alignment A_k.
5. **Monotonicity (*mon*):** The absolute difference between i and j.

Our combination approach employs feature functions derived from a subset of the features above. Assuming $\mathcal{Y} = \{\text{yes}, \text{no}\}$ represents the set of classes, where each class denotes the existence or absence of a link in the combined alignment, and $\mathcal{X}$ is the set of features above, we generate various feature functions $h(y, x)$, where $y \in \mathcal{Y}$ and x are instantiations of one or more features in $\mathcal{X}$. Table 1 lists the feature sets with an example feature func-

Features	Example Feature Function	
$posE$	$h('\text{yes}', i, j) = 1$	if $(i,j) \in A_C$ and $pos(e_i) = Noun$
$posF$	$h('\text{no}', i, j) = 1$	if $(i,j) \notin A_C$ and $pos(f_j) = Verb$
out	$h('\text{yes}', i, j, k) = 1$	if $(i,j) \in A_C$ and $(i,j) \in A_k$
$out, neigh$	$h('\text{yes}', i, j, k) = 1$	if $(i,j) \in A_C$ and $(i-1, j+1) \in A_k$
	$h('\text{yes}', i, j, k) = 1$	if $(i,j) \in A_C$ and $\|NC\| = 2$ where $NC = \{n \| n \in N(i,j), n \in A_k\}$
$out, fertE$	$h('\text{no}', i, j, k) = 1$	if $(i,j) \notin A_C$ and $\|FT\| = 0$ where $FT = \{t \| (i,t) \in A_k\}$
$out, fertF$	$h('\text{no}', i, j, k) = 1$	if $(i,j) \notin A_C$ and $\|FT\| = 1$ where $FT = \{t \| (t,j) \in A_k\}$
mon	$h('\text{yes}', i, j) = 1$	if $(i,j) \in A_C$ and $\|i - j\| = 2$

Table 1: Feature Functions.

tion for each.[1] For example, the feature function in the fifth row has a value of 1 if there are 2 neighboring links to (i, j) that exist in the input alignment A_k and the alignment link (i, j) exists in A_C.

In combining evidence from different alignments, it is assumed that, when an alignment link is left out by all aligners, that particular link should not be included in the final output. Since the majority of all possible word pairs are unaligned in real data, the inclusion of all possible word pairs in the training data leads to skewed results, where the learning algorithm is biased toward labeling the links as invalid. To offset this problem, our training data includes only alignment links that appear in at least one input alignment.

Once the feature functions are extracted, we learn the model parameters using the YASMET ME package (Och, 2002), which is an efficient implementation of the GIS algorithm.

4 Experiment Data, Alignment Inputs, and Metrics

The alignment combination techniques are evaluated in this paper using data from three language pairs, as shown in Table 2.

Lang Pair	# of Sent's	# Words (en/fl)	Source
en-ch	491	13K/13K	NIST MTEval '02[2]
en-ar	450	11K/13K	NIST MTEval '03[3]
en-ro	248	5.5K/5.5K	HLT Workshop '03[4]

Table 2: Data Used for Combination Experiments.

Input alignments are generated using two existing word alignment systems: GIZA++ (Och, 2000) and SAHMM (Lopez and Resnik, 2005). Both systems are run in two different directions with default configurations. We indicate the two directions using the notation $Aligner(en \rightarrow fl)$ and $Aligner(fl \rightarrow en)$, where en is English, fl is either Chinese (*ch*), Arabic (*ar*), or Romanian (*ro*).

To train both systems, additional data was used for the three language pairs: 107K English-Chinese sentence pairs (4.1M/3.3M English/Chinese words); 44K English-Arabic sentence pairs (1.4M/1M English/Arabic words); 48K English-Romanian sentence pairs (1M/1M English/Romanian words).[5]

POS tags were generated using the MXPOST tagger (Ratnaparkhi, 1998). POS tagger for English was trained on Sections 0-18 of the Penn Treebank Wall Street Journal corpus. On the FL side, we used POS tagger for only Chinese and it was trained on Sections 16-299 of Chinese Treebank.

For comparison purposes, three additional heuristically-induced alignments are generated for each system: (1) Intersection of both directions ($Aligner$(int)); (2) Union of both directions ($Aligner$(union)); and (3) The previously best-known heuristic combination approach called *grow-diag-final* (Koehn et al., 2003) ($Aligner$(gdf)).

In our evaluation, we take A to be the set of alignment links for a set of sentences, S to be the set of sure alignment links, and P be the set of probable alignment links (in the gold standard). Precision (Pr), recall (Rc) and alignment error rate (AER) are defined as follows:[6]

$$Pr = \frac{|A \cap P|}{|A|} \quad Rc = \frac{|A \cap S|}{|S|}$$

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

[1] In Table 1, *NC* corresponds to the set of (i, j)'s neighbors that exist in the alignment A_k, and *FT* represents the set of words that e_i (or f_j) is aligned to.

[2] From (Ayan et al., 2005).

[3] From (Ittycheriah and Roukos, 2005).

[4] From (Mihalcea and Pedersen, 2003).

[5] Note that both GIZA++ and SAHMM are unsupervised learning systems. Sentence-aligned parallel texts are the only required input.

[6] Note that AER= 1 - F-score when there is no distinction between probable and sure alignment links.

Our gold standard for each language pair is a manually aligned corpus. English-Chinese annotations distinguish between sure and probable alignment links (i.e., $S \subset P$), but there is no such distinction for the other two language pairs (i.e., $P = S$).

Because of the availability of limited manually annotated data, evaluations are performed using 5-fold cross validation. Once the alignments are generated for each fold (using one as the test set and the other 4 folds as training set), the results are concatenated to compute precision, recall and error rate on the entire set of sentence pairs for each data set.[7]

5 Experiments and Results

This section presents several experiments and results comparing AER of ACME to those of standard alignment approaches on English-Chinese data. We also present experiments on additional languages, analyses based on precision and recall, an upper-bound oracle analysis, and MT evaluations.

5.1 English-Chinese Experiments

The experiments below test the effects of input alignments, feature set, data partitioning, number of inputs, and size of training data on the performance of ACME.

2 Input alignments: Table 3 shows the AER for GIZA++ and SAHMM (in each direction), three heuristic-based combinations and ACME using 2 uni-directional alignments as input and all features described in Section 3.[8] (We use 'ACME[2]' in this section to refer to ACME applied to two input alignments and ACME[4] in later sections to refer to ACME applied to four input alignments.)

Using 2 GIZA++ uni-directional alignments as input, ACME yields a 22.0% AER—a relative error reduction of 25.9% over GIZA++(gdf). Similarly, using 2 SAHMM uni-directional alignments as input, ACME produces a 20.6% AER—a relative error reduction of 28.0% and 25.4% over SAHMM(gdf) and SAHMM(int), respectively.

[7] Because the NIST MTEval data include sentences that may be related (according to the document in which they appear), the training and test material could potentially be related; however, given the types of features used in our experiments, we do not believe this biases our results.

[8] For ease of readability, in the rest of this paper, we will report precision, recall, and AER in percentages.

Alignments	GIZA++	SAHMM
$Aligner(en \rightarrow fl)$	30.7	26.5
$Aligner(fl \rightarrow en)$	32.2	31.3
$Aligner(int)$	31.2	27.6
$Aligner(union)$	31.6	29.8
$Aligner(gdf)$	29.7	28.6
ACME[2]	22.0	20.6

Table 3: Comparison of GIZA++ and SAHMM to ACME[2] (on English-Chinese).

Feature Set: To examine the effects of each feature on the performance of ACME, we compute the AER under a variety of conditions, removing each feature one at a time. ACME is evaluated using 2 uni-directional GIZA++ alignments as input on English-Chinese data. Using all features, the AER is 22.0%. Our experiments show that there is no significant increase in AER for the removal of features corresponding to monotonicity (22.1%), neighbors (22.8%), POS on English side (22.9%), POS on foreign-language side (22.9%). On the other hand, deleting POS tags on both sides yields an AER of 25.2% and deleting the fertility features increases the AER to 25.9%. This indicates that both POS tags (or fertilities) contribute heavily toward the decision as to whether a particular alignment should be included/excluded.

Partitioning Data: Previous work showed that partitioning the data into disjoint subsets and learning a different model for each partition improves the performance of the alignment systems (Ayan et al., 2005). To test whether this same principle applies to alignment combination with maximum entropy modeling, the training data was partitioned using POS tags for English and the FL, and different weights were learned for each partition.

Alignments	GIZA++	SAHMM
ACME[2]	22.0	20.6
ACME[2]-Part[$posE$]	19.8	18.0
ACME[2]-Part[$posF$]	20.0	18.1
ACME[2]-Part[$posE, posF$]	20.0	18.4

Table 4: Application of ACME[2] on Partitioned Data (on English-Chinese).

Table 4 presents the AER for ACME[2], using either two GIZA++ alignments or two SAHMM alignments, on English-Chinese data. Without any partitioning, ACME achieves an AER of 22.0 (GIZA++) and 20.6 (SAHMM). Using English POS tags for data partitioning results in a significant reduction

in AER: 19.8% (GIZA++) and 18.0% (SAHMM). Interestingly, using foreign-language (FL) tags on their own or together with English POS tags does not provide any improvement. Overall when ACME[2] is applied to partitioned data (using *posE* for partitioning) a relative error reduction of 33–37% over GIZA++(gdf) and SAHMM(gdf) is achieved.

Number of Input Alignments: Table 5 presents the English-Chinese AER for ACME[1] (using either GIZA++ or SAHMM in only one direction), ACME[2] (using either GIZA++ or SAHMM in two directions) and ACME[4] (using GIZA++ and SAHMM, each in two directions).

Regardless of the number of inputs, partitioning the data (using English POS tags) yields lower AER than no partitioning. Using one GIZA++ alignment as input, ACME[1] with partitioning improves the AER to 26.9% and 25.5% for each direction, respectively. Similarly, using one SAHMM alignment as input, ACME[1] with partitioning reduces the AER to 22.9% and 24.7%. ACME[2] with partitioning reduces the AER to 19.8% and 18.0% for GIZA++ and SAHMM, respectively. Finally, using all four input alignments, ACME[4] with partitioning yields a 15.6% AER—a relative error reduction of 21.2% and 13.3% over each ACME[2] case.

Alignments	GIZA++	SAHMM
ACME[1]($en \rightarrow fl$)	28.1	24.4
ACME[1]-Part[posE]($en \rightarrow fl$)	26.9	22.9
ACME[1]($fl \rightarrow en$)	26.6	26.9
ACME[1]-Part[posE]($fl \rightarrow en$)	25.5	24.7
ACME[2]	22.0	20.6
ACME[2]-Part[posE]	19.8	18.0
ACME[4]	17.8	
ACME[4]-Part[posE]	15.6	

Table 5: Application of ACME to 1, 2 and 4 Input Alignments (on English-Chinese).

Size of Training Data to Obtain Input Alignments: In general, statistical alignment systems improve as the size of the training data increases. We present the AER for GIZA++ and ACME[2] using GIZA++ alignments as input, where GIZA++ is trained on different sizes of data. We started with 20K sentence pairs of FBIS data and increased it to all available FBIS data (241K sentence pairs).

Figure 1 compares the alignment performance of: (1) uni-directional GIZA++ (each direction); (2) GIZA++(gdf); and (3) ACME[2] with all features and English POS partitioning. With only 20K sentence pairs, ACME[2] achieves an AER of 23.7% in contrast to 34.3% AER for GIZA++(gdf). With 241K sentence pairs, ACME[2] yields 18.3% AER in contrast to 27.7% AER for GIZA++(gdf). We should emphasize that ACME[2] on only 20K sentence pairs yields a lower AER than those of all GIZA++ alignments obtained on 241K sentence pairs. Overall ACME[2] achieves a relative error reduction of 31–38% over the input alignments, and a relative error reduction of 31–34% over GIZA++(gdf) for different sizes of training data.

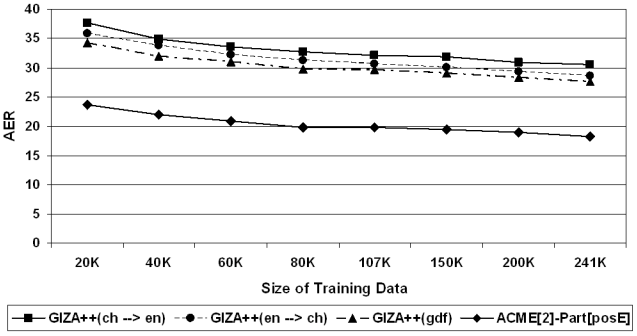

Figure 1: Effects of Training Data Size Used for Initial Alignments on the performance of GIZA++ and ACME[2] (on English-Chinese).

5.2 Expanding to Additional Languages

We also investigated the applicability of ACME to additional language pairs. Table 6 presents the AER for GIZA++ and SAHMM (in each direction), three combination heuristics (gdf, int and union), and ACME[2] and ACME[4] on English-Arabic and English-Romanian data. We should emphasize that no POS tagger on the FL side was used for these experiments.

On English-Arabic data, ACME[2] (with POS partitioning and including all features) yields 21.4% (20.7%) AER—a relative error reduction of 24.6% (13.0%) over the best combination heuristic with GIZA++ (SAHMM) alignments. ACME[4] reduces the AER to 18.1%—a relative error reduction of 36.3% and 23.9% over GIZA++(int) and SAHMM(int), respectively.

On English-Romanian data, ACME[2] (with POS partitioning and including all features) yields 24.7% (26.2%) AER—a relative error reduction of 14.3% (10.6%) over the best combination heuristic with GIZA++ (SAHMM) alignments. ACME[4] re-

	English-Arabic		English-Romanian	
Alignments	**GIZA++**	**SAHMM**	**GIZA++**	**SAHMM**
$Aligner(en \rightarrow fl)$	34.5	27.8	32.7	31.0
$Aligner(fl \rightarrow en)$	27.9	29.5	30.0	29.8
$Aligner(int)$	28.4	23.8	32.7	29.3
$Aligner(union)$	32.8	32.0	30.5	31.2
$Aligner(gdf)$	30.2	30.4	28.8	30.3
ACME[2]	23.2	21.9	25.2	27.0
ACME[2]-Part[$posE$]	21.4	20.7	24.7	26.2
ACME[4]	19.8		24.0	
ACME[4]-Part[$posE$]	18.1		22.3	

Table 6: AER for Input Alignments, Heuristic-based Alignments, and ACME Using 2 and 4 Input Alignments (on English-Arabic and English-Romanian).

duces the AER to 22.3%—a relative error reduction of 22.6% and 23.9% over GIZA++(int) and SAHMM(int), respectively.

5.3 Precision, Recall and Upper-Bound Analysis

We now turn to a precision vs. recall analysis of different alignments to elucidate the nature of the differences between two alignments.

Figure 2 presents precision and recall values for three combined alignments using GIZA++ (int, union, gdf) as well as results for ACME[2] and ACME[4] on three different language pairs. For all three pairs, the ranking of the combined alignments is the same with respect to precision and recall. GIZA++(int) yields the highest precision (nearly 95%) but the lowest recall (53–57%). Both union and gdf methods achieve low precision (56–68%) but high recall (75–83%), and gdf is better than union. By contrast, ACME[2] yields significantly higher precision (nearly 87%) but lower recall (67–75%) with respect to union and gdf. ACME[4] has higher precision and recall than ACME[2]—an absolute increase of 2–3% and 4%, respectively.

Next we compute an oracle upper-bound in AER where mismatched input alignments are assumed to be resolved perfectly within the alignment combination framework (i.e., an *oracle* chooses the correct output in cases where the input aligners make different choices).[9]

Table 7 presents the upper bounds using a generic alignment combiner (denoted *Oracle*) with 2 and 4 input alignments on three language pairs, assuming a perfect resolution of mismatched input alignments. For English-Chinese, the upper bound is 9.4% (using *Oracle*[2]) and 4.7% (using *Oracle*[4]). The English-Arabic data exhibits a slightly higher upper bound of 5.5% for *Oracle*[4]. The upper bounds for AER on English-Romanian data are even higher (up to 17.7%), which indicates that the input alignments are significantly worse than others. This may be one of the main contributing factors to the lower improvement of ACME on English-Romanian in comparison to the other two language pairs.

Alignments	**GIZA++**	**SAHMM**
Oracle[2] (en-ch)	9.4	8.4
Oracle[4] (en-ch)	4.7	
Oracle[2] (en-ar)	9.8	11.1
Oracle[4] (en-ar)	5.5	
Oracle[2] (en-ro)	15.4	17.7
Oracle[4] (en-ro)	11.3	

Table 7: Oracle Upper Bounds on AER for Alignment Combination

5.4 MT Evaluation

To determine the contribution of improved alignment in an external application, we examined the improvement in an off-the-shelf phrase-based MT system Pharaoh (Koehn, 2004) on both Chinese and Arabic data. In these experiments, all components of the MT system were kept the same except for the component that generates a phrase table from a given alignment.

The input alignments were generated using GIZA++ and SAHMM on 107K (44K) sentence pairs for Chinese (Arabic). ACME (with English POS partitioning) combines alignments using model parameters learned from the corresponding manually aligned data. MT output is evaluated using the standard MT evaluation metric BLEU (Papineni et al., 2002).[10] Table 8 presents the BLEU scores on

[9] If the input aligners agree on a particular link, that decision is taken as the final output in computing the upper bound.

[10] We used the NIST script (version 11a) with its default set-

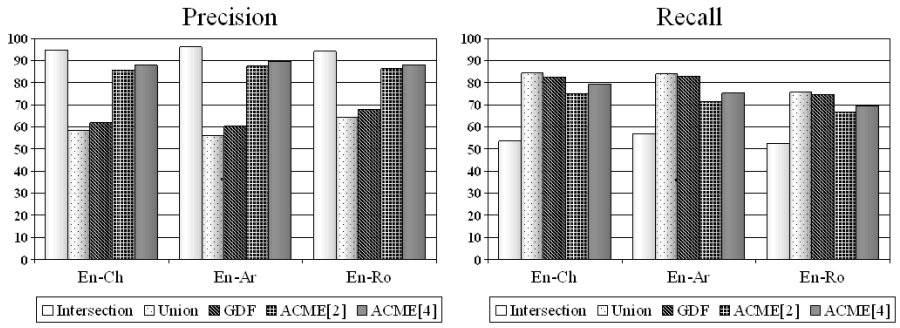

Figure 2: Precision and Recall Scores for GIZA++ and ACME Using 2 and 4 Input Alignments.

MTEval'03 data for 5 different Pharaoh runs, one for each alignment. The parameters of the MT system were optimized on MTEval'02 data using minimum error rate training (Och, 2003).

For the language model, the SRI Language Modeling Toolkit was used to train a trigram model with modified Kneser-Ney smoothing on 155M words of English newswire text, mostly from the Xinhua portion of the Gigaword corpus. During decoding, the number of English phrases per FL phrase was limited to 100 and the distortion of phrases was limited by 4. Based on the observations in (Koehn et al., 2003), we also limited the phrase length to 3 for computational reasons.

Alignment	Chinese	Arabic
GIZA++(union)	22.66	41.72
GIZA++(gdf)	23.79	43.82
GIZA++(int)	23.97	42.76
ACME[2]	25.20	44.94
ACME[4]	**25.59**	**45.54**

Table 8: Evaluation of Pharaoh with Different Initial Alignments using BLEU (in percentages)

For both languages, ACME[2] and ACME[4] outperform the other three alignment combination techniques. ACME[4], for instance, yields the BLEU scores of 25.59% for Chinese and 45.54% for Arabic—an absolute 1.6-1.7% BLEU point increase over the best of the other three alignment combinations. The differences between the BLEU scores for ACME and the other three BLEU scores are statistically significant, using a significance test with bootstrap resampling (Zhang et al., 2004).

6 Related Work

ME models have been previously applied to several NLP problems, including word alignments. For instance, the IBM models (Brown et al., 1993) can be improved by adding more context dependencies into the translation model using a ME framework rather than using only $p(f_j|e_i)$ (Garcia-Varea et al., 2002). In a later study, Och and Ney (2003) present a log-linear combination of the HMM and IBM Model 4 that produces better alignments than either of those. The major advantage of these two methods is that they do not require manually annotated data.

The alignment process can be modeled as a product of a transition model and an observation model, where ME models the observations (Ittycheriah and Roukos, 2005). Significant improvements are reported using this approach but the need for large manually aligned data is a bottleneck. An alternative ME approach models alignment directly as a log-linear combination of feature functions (Liu et al., 2005). Moore (2005) and Taskar et al. (2005) represent alignments with several feature functions that are then combined in a weighted sum to model word alignments. Once a confidence score is assigned to all links, a non-trivial search is invoked to find the best alignment using the scores associated with the links. The major difference between these approaches and that of ACME is that we use the ME model to predict the correct class for each alignment link independently using outputs of existing alignment systems, instead of generating them from scratch at the level of the whole sentence, thus eliminating the need for an exhaustive search over all possible alignments, i.e., previous approaches work globally while ACME is a localized model. A discussion of these two contrasting approaches can be found in (Tillmann and Zhang, 2005).

A recent attempt to combine outputs of different alignments views the combination problem as a classifier ensemble in the neural network framework

tings: case-insensitive matching of n-grams up to $n = 4$, and the shortest reference sentence for the brevity penalty.

(Ayan et al., 2005). However, this method is subject to the unpredictability of random network initialization, whereas ACME is guaranteed to find the model that maximizes the likelihood of training data.

7 Conclusions

We presented a new approach, ACME, to combining the outputs of different word alignment systems by reducing the combination problem to the level of alignment links and using a maximum entropy model to learn whether a particular alignment link is included in the final alignment.

Our results indicate that ACME yields significant relative error reduction over the input alignments and their heuristic-based combinations on three different language pairs. Moreover, ACME provides similar relative improvements for different sizes of training data for the input alignment systems. We have also shown that using a higher number of input alignments, and partitioning the training data into disjoint subsets and learning a different model for each partition yield further improvements.

We have tested impact of the reduced AER on MT and have shown that alignments generated by ACME yield statistically significant improvements in BLEU scores in two different languages, even if we don't employ a POS tagger on the FL side. However, additional studies are needed to investigate why huge improvements in AER result in relatively smaller improvements in BLEU scores.

Because ACME is a supervised learning approach, it requires annotated data; however, our experiments have shown that significant improvements can be obtained using a small set of annotated data.

Acknowledgments This work has been supported, in part, under ONR MURI Contract FCPO.810548265 and the GALE program of the Defense Advanced Research Projects Agency, Contracts No. HR0011-06-2-0001. We also thank anonymous reviewers for their helpful comments.

References

Necip F. Ayan, Bonnie J. Dorr, and Christof Monz. 2005. Neuralign: Combining word alignments using neural networks. In *Proceedings of EMNLP'2005*, pages 65–72.

Adam L. Berger, Stephan A. Della-Pietra, and Vincent J. Della-Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1).

Peter F. Brown, Stephan A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480.

Michael Fleischman, Namhee Kwon, and Eduard Hovy. 2003. Maximum entropy models for framenet classification. In *Proceedings of EMNLP'2003*.

Ismael Garcia-Varea, Franz Josef Och, Hermann Ney, and Francisco Casacuberta. 2002. Improving alignment quality in statistical machine translation using context-dependent maximum entropy models. In *Proceedings of COLING'2002*.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of EMNLP'2005*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL'2003*.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation. In *Proceedings of AMTA'2004*.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of ACL'2005*.

Adam Lopez and Philip Resnik. 2005. Improved HMM alignment models for languages with scarce resources. In *Proceedings of the ACL'2005 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 83–86.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL'2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of EMNLP'2005*.

Franz J. Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL'2002*, pages 295–302.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):9–51, March.

Franz J. Och. 2000. GIZA++: Training of statistical translation models. Technical report, RWTH Aachen, University of Technology.

Franz J. Och. 2002. Yet another maxent toolkit: YASMET. Available at http://www.fjoch.com/YASMET.html.

Franz J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL'2003*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'2002*, pages 311–318.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of EMNLP'2005*.

Christoph Tillmann and Tong Zhang. 2005. A localized prediction model for statistical machine translation. In *Proceedings of ACL'2005*.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of LREC'2004*, pages 2051–2054.