EMDC: A Semi-supervised Approach for Word Alignment

Qin Gao
Language Technologies Institute
Carnegie Mellon University
qing@cs.cmu.edu

Francisco Guzman
Centro de Sistemas Inteligentes
Tecnológico de Monterrey
guzmanhe@gmail.com

Stephan Vogel
Language Technologies Institute
Carnegie Mellon University
stephan.vogel@cs.cmu.edu

Abstract

This paper proposes a novel semi-supervised word alignment technique called EMDC that integrates discriminative and generative methods. A discriminative aligner is used to find high precision partial alignments that serve as constraints for a generative aligner which implements a constrained version of the EM algorithm. Experiments on small-size Chinese and Arabic tasks show consistent improvements on AER. We also experimented with moderate-size Chinese machine translation tasks and got an average of 0.5 point improvement on BLEU scores across five standard NIST test sets and four other test sets.

1 Introduction

Word alignment is a crucial component in statistical machine translation (SMT). From a Machine Learning perspective, the models for word alignment can be roughly categorized as generative models and discriminative models. The widely used word alignment tool, i.e. GIZA++ (Och and Ney, 2003), implements the well-known IBM models (Brown et al., 1993) and the HMM model (Vogel et al., 1996), which are generative models. For language pairs such as Chinese-English, the word alignment quality is often unsatisfactory. There has been increasing interest on using manual alignments in word alignment tasks, which has resulted in several discriminative models. Ittycheriah and Roukos (2005) proposed to use only manual alignment links in a maximum entropy model, which is considered supervised. Also, a number of semi-supervised word aligners have been proposed (Taskar et al., 2005; Liu et al., 2005; Moore, 2005; Blunsom and Cohn, 2006; Niehues and Vogel, 2008). These methods use held-out manual alignments to tune weights for discriminative models, while using the model parameters, model scores or alignment links from unsupervised word aligners as features. Callison-Burch et. al. (2004) proposed a method to interpolate the parameters estimated by sentence-aligned and word-aligned corpus. Also, there are recent attempts to combine multiple alignment sources using alignment confidence measures so as to improve the alignment quality (Huang, 2009).

In this paper, the question we address is whether we can jointly improve discriminative models and generative models by feeding the information we get from the discriminative aligner back into the generative aligner. Examples of this line of research include Model 6 (Och and Ney, 2003) and the EMD training approach proposed by Fraser and Marcu (2006) and its extension called LEAF aligner (Fraser and Marcu, 2007). These approaches use labeled data to tune additional parameters to weight different components of the IBM models such as the lexical translation model, the distortion model and the fertility model. These methods are proven to be effective in improving the quality of alignments. However, the discriminative training in these methods is restricted in using the model components of generative models, in other words, incorporating new features is difficult.

Instead of using discriminative training methods to tune the weights of generative models, in this paper we propose to use a discriminative word aligner to produce reliable constraints for the EM algorithm. We call this new training scheme EMDC (**E**xpectation-**M**aximization-**D**iscrimination-**C**onstraint). The methodology can be viewed as a variation of bootstrapping. It enables the generative models to interact with discriminative models at the data level instead of the model level. Furthermore, with a discriminative

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 349–357,
Beijing, August 2010

word aligner that uses generative word aligner's output as features, we create a feedback loop that can iteratively improve the quality of both aligners. The major contributions of this paper are: 1) The EMDC training scheme, which ties the generative and discriminative aligners together and enables future research on integrating other discriminative aligners. 2) An extended generative aligner based on GIZA++ that allows to perform constrained EM training.

In Section 2, we present the EMDC training scheme. Section 3 provides details of the constrained EM algorithm. In Section 4, we introduce the discriminative aligner and link filtering. Section 5 provides the experiment set-up and the results. Section 6 concludes the paper.

2 EMDC Training Scheme

The EMDC training scheme consists of three parts, namely **EM**, **D**iscrimination, and **C**onstraints. As illustrated in Figure 1, a large unlabeled training set is first aligned with a generative aligner (GIZA++ for the purpose of this paper). The generative aligner outputs the model parameters and the Viterbi alignments for both source-to-target and target-to-source directions. Afterwards, a discriminative aligner (we use the one described in (Niehues and Vogel, 2008)), takes the lexical translation model, fertility model and Viterbi alignments from both directions as features, and is tuned to optimize the AER on a small manually aligned tuning set. Afterwards, the alignment links generated by the discriminative aligner are filtered according to their likelihood, resulting in a subset of links that has high precision and low recall. The next step is to put these high precision alignment links back into the generative aligner as constraints. A conventional generative word aligner does not support this type of constraints. Thus we developed a constrained EM algorithm that can use the links from a partial alignment as constraints and estimate the model parameters by marginalizing likelihoods.

After the constrained EM training is performed, we repeat the procedure and put the *updated* generative models and Viterbi alignment back into the discriminative aligner. We can either fix the number of iterations, or stop the procedure when the gain on AER of a small held-out test set drops below a threshold.

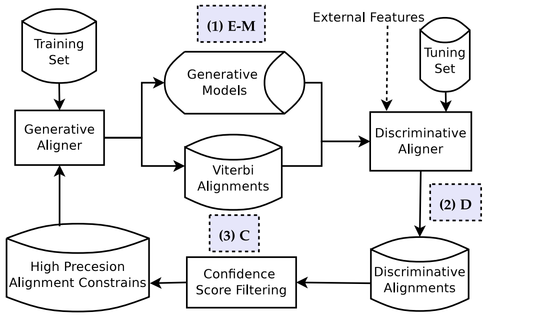


Figure 1: Illustration of EMDC training scheme

The key components for the system are:

1. A generative aligner that can make use of reliable alignment links as constraints and improve the models/alignments.
2. A discriminative aligner that outputs confidence scores for alignment links, which allows to obtain high-precision-low-recall alignments.

While in this paper we derive the reliable links by filtering the alignment generated by a discriminative aligner, such partial alignments may be obtained from other sources as well: manual alignments, specific named entity aligner, noun-phrase aligner, etc.

As we mentioned in Section 1, the discriminative aligner is not restricted to use features parameters of generative models and Viterbi alignments. However, including the features from generative models is required for iterative training, because the improvement on the quality of these features can in turn improve the discriminative aligner. In our experiments, the discriminative aligner makes heavy use of the Viterbi alignment and the model parameters from the generative aligner. Nonetheless, one can easily replace the discriminative aligner or add new features to it without modifying the training scheme. The open-ended property of the training scheme makes it a promising method to integrate different aligners.

In the next two sections, we will describe the key components of this framework in detail.

3 Constrained EM algorithm

In this section we will briefly introduce the constrained EM algorithm we used in the experiment, further details of the algorithm can be found in (Gao et al., 2010).

The IBM Models (Brown et al., 1993) are a series of generative models for word alignment. GIZA++ (Och and Ney, 2003), the most widely used implementation of IBM models and HMM (Vogel et al., 1996), employs EM algorithm to estimate the model parameters. For simpler models such as Model 1 and Model 2, it is possible to obtain sufficient statistics from all possible alignments in the E-step. However, for fertility-based models such as Models 3, 4, and 5, enumerating all possible alignments is NP-complete. To overcome this limitation, GIZA++ adopts a greedy hill-climbing algorithm, which uses simpler models such as HMM or Model 2 to generate a "center alignment" and then tries to find better alignments among its neighbors. The neighbors of an alignment $a_1^J = [a_1, a_2, \cdots, a_J]$ with $a_j \in [0, I]$ are defined as alignments that can be generated from a_1^J by one of the following two operators:

1. The move operator $m_{[i,j]}$, that changes $a_j := i$, i.e. arbitrarily sets word f_j in the target sentence to align to the word e_i in source sentence;
2. The swap operator $s_{[j_1,j_2]}$ that exchanges a_{j_1} and a_{j_2}.

The algorithm will update the center alignment as long as a better alignment can be found, and finally outputs a local optimal alignment. The neighbor alignments of the final center alignment are then used in collecting the counts for the M-Step. Och and Ney (2003) proposed a fast implementation of the hill-climbing algorithm that employs two matrices, i.e. Moving Matrix $M_{I\times J}$ and Swapping Matrix $S_{J\times J}$. Each cell of the matrices stores the value of likelihood difference after applying the corresponding operator.

We define a partial alignment constraint of a sentence pair (f_1^J, e_1^I) as a set of links: $\alpha_I^J = \{(i,j)|0 \le i < I, 0 \le j < J\}$. Given a set of constraints, an alignment $a_1^J = [a_1, a_2, \cdots, a_j]$ on the sentence pair f_1^J, e_1^I, the translation probability of $Pr(f_1^J|e_1^I)$ will be zero if the alignment is inconsistent with the constraints. Constraints $(0, j)$ or $(i, 0)$ are used to explicitly represent that word f_j or e_i is aligned to the empty word.

Under the assumptions of the IBM models, there are two situations that a_1^J is inconsistent with α_I^J:

1. Target word misalignment: The IBM models assume that one target word can only be aligned to one source word. Therefore, if the target word f_j aligns to a source word e_i, while the constraint α_I^J suggests f_j should be aligned to $e_{i'}$, the alignment violates the constraint and thus is considered inconsistent.
2. Source word to empty word misalignment: if a source word is aligned to the empty word, it cannot be aligned to any concrete target word.

However, the partial alignments, which allow n-to-n alignments, may already violate the 1-to-n alignment restriction of the IBM models. In these cases, we relax the condition in situation 1 that if the alignment link a_{j^*} is consistent with any one of the conflicting target-to-source constraints, it will be considered consistent. Also, we arbitrarily assign the source word to empty word constraints higher priorities than other constraints, because unlike situation 1, it does not have the problem of conflicting with other constraints.

3.1 Constrained hill-climbing algorithm

To ensure that resulting center alignment be consistent with the constraints, we need to split the hill-climbing algorithm into two stages: 1) optimize towards the constraints and 2) optimize towards the optimal alignment under the constraints.

From a seed alignment, we first move the alignment towards the constraints by choosing a move or swap operator that:

1. produces the alignment that has the highest likelihood among alignments generated by other operators,
2. eliminates at least one inconsistent link.

We iteratively update the alignment until no other inconsistent link can be removed. The algorithm implies that we force the seed alignment to be closer to the constraints while trying to find the best consistent alignment. Figure 2 demonstrates the idea, given the constraints shown in (a), and the seed alignment shown as solid links in (b), we

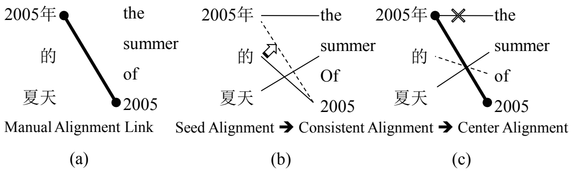


Figure 2: Illustration of Algorithm 1

move the inconsistent link to the dashed link by a move operation.

After we find the consistent alignment, we proceed to optimize towards the optimal alignment under the constraints. The algorithm sets the value of the cells in moving/swapping matrices to negative if the corresponding operators will lead to an inconsistent alignment. The moving matrix needs to be processed only once, whereas the swapping matrix needs to be updated every iteration, since once the alignment is updated, the possible violations will also change.

If a source word e_i is aligned to the empty word, we set $M_{i,j} = -1, \forall j$. The swapping matrix does not need to be modified in this case because the swapping operator will not introduce new links.

Because the cells that can lead to violations are set to negative, the operators will never be picked when updating the center alignments. This ensures the consistency of the final center alignment.

3.2 Count Collection

After finding the center alignment, we need to collect counts from neighbor alignments so that the M-step can normalize the counts to produce the model parameters for the next step. In this stage, we want to make sure all the inconsistent alignments in the neighbor set of the center alignment be ruled out from the sufficient statistics, i.e. have zero probability. Similar to the constrained hill climbing algorithm, we can manipulate the moving/swapping matrices to effectively exclude inconsistent alignments. Since the original count collection algorithm depends only on moving and swapping matrices, we just need to bypass all the cells which hold negative values, i.e. represent inconsistent alignments.

We can also view the algorithm as forcing the posteriors of inconsistent alignments to zero, and therefore increase the posteriors of consistent alignments. When no constraint is given, the algorithm falls back to conventional EM, and when all the alignments are known, the algorithm becomes fully supervised. And if the alignment quality can be improved if high-precision partial alignment links is given as constraints. In (Gao et al., 2010) we experimented with using a dictionary to generate such constraints, and in (Gao and Vogel, 2010) we experimented with manual word alignments from Mechanical Turk. And in this paper we try to use an alternative method that uses a discriminative aligner and link filtering to generate such constraints.

4 Discriminative Aligner and Link Filtering

We employ the CRF-based discriminative word aligner described in (Niehues and Vogel, 2008). The aligner can use a variety of knowledge sources as features, such as: the fertility and lexical translation model parameters from GIZA++, the Viterbi alignment from both source-to-target and target-to-source directions. It can also make use of first-order features which model the dependency between different links, the Parts-of-Speech tagging features, the word form similarity feature and the phrase features. In this paper we use all the features mentioned above except the POS and phrase features.

The aligner is trained using a belief-propagation (BP) algorithm, and can be optimized to maximize likelihood or directly optimize towards AER on a tuning set. The aligner outputs confidence scores for alignment links, which allows us to control the precision and recall rate of the resulting alignment. Guzman et al. (2009) experimented with different alignments produced by adjusting the filtering threshold for the alignment links and showed that they could get high-precision-low-recall alignments by having a higher threshold. Therefore, we replicated the confidence filtering procedures to produce the partial alignment constraints. Afterwards we iterate by putting the partial alignments back to the constrained word alignment algorithm described in section 3.

Although the discriminative aligner performs well in supplying high precision constraints, it does not model the null alignment explicitly.

	Num. of	Num. of Words		Num. of
	Sentences	*Source*	*Target*	Links
Ch-En	21,863	424,683	524,882	687,247
Ar-En	29,876	630,101	821,938	830,349

Table 1: Corpus statistics of the manual aligned corpora

Threshold		P	R	AER
Ch-En	0.6	71.30	58.12	35.96
	0.7	75.24	54.03	37.11
	0.8	85.66	44.19	41.70
	0.9	93.70	37.95	45.98
Ar-En	0.6	72.35	59.87	34.48
	0.7	77.55	55.58	35.25
	0.8	80.07	50.89	37.77
	0.9	83.74	44.16	42.17

Table 2: The qualities of the constraints

Hence we are currently not able to provide source word to empty word alignment constraints which have been proven to be effective in improving the alignment quality in (Gao et al., 2010). Due to space limitation, please refer to: (Niehues and Vogel, 2008; Guzman et al., 2009) for further details of the aligner and link filtering, respectively.

5 Experiments

To validate the proposed training scheme, we performed two sets of experiments. First of all, we experimented with a small manually aligned corpus to evaluate the ability of the algorithm to improve the AER. The experiment was performed on Chinese to English and Arabic to English tasks. Secondly, we experimented with a moderate size corpus and performed translation tasks to observe the effects in translation quality.

5.1 Effects on AER

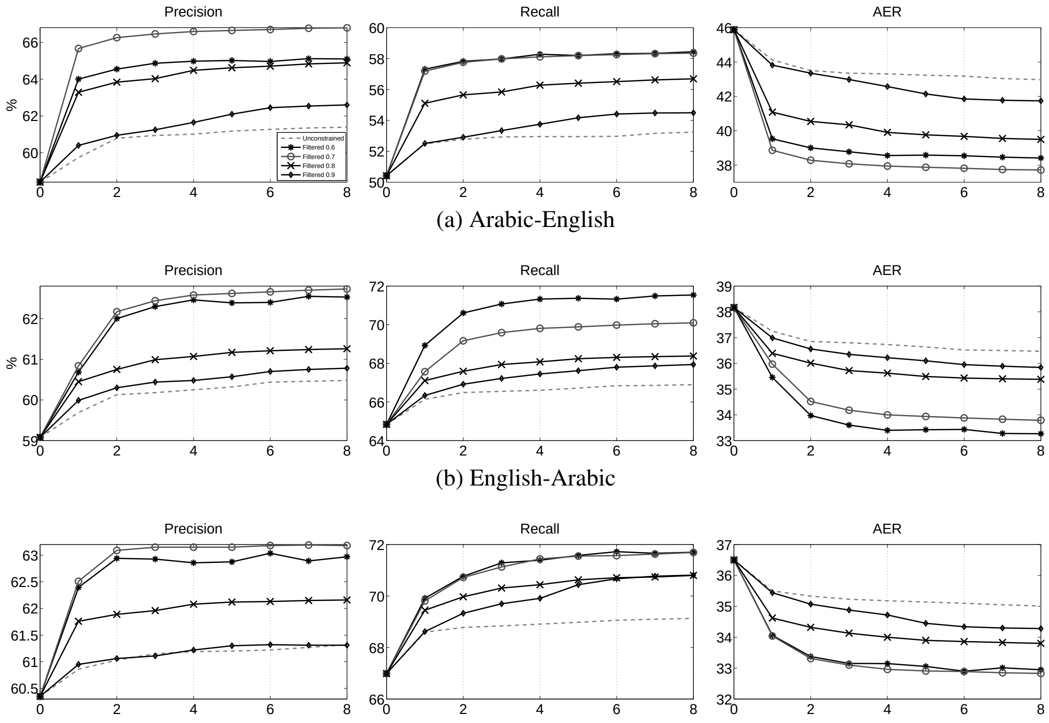
In order to measure the effects of EMDC in alignment quality, we experimented with Chinese-English and Arabic-English manually aligned corpora. The statistics of these sets are shown in Table 1. We split the data into two fragments, the first 100 sentences (Set A) and the remaining (Set B). We trained generative IBM models using the Set B, and tuned the discriminative aligner using the Set A. We evaluated the AER on Set B, but in any of the training steps the manual alignments of Set B were not used.

In each iteration of EDMC, we load the model parameters from the previous step and continue training using the new constraints. Therefore, it is important to compare the performance of continuous training against an unconstrained baseline, because variation in alignment quality could be attributed to either the effect of more training iterations or to the effect of semi-supervised training scheme. In Figures 3 and 4 we show the alignment quality for each iteration. Iteration 0 is the baseline, which comes from standard GIZA++ training[1]. The grey dash curves represent unconstrained Model 4 training, and the curves with start, circle, cross and diamond markers are constrained EM alignments with 0.6, 0.7, 0.8 and 0.9 filtering thresholds respectively. As we can see from the results, when comparing only the mono-directional trainings, the alignment qualities improve over the unconstrained training in all the metrics (precision, recall and AER). From Table 2, we observe that the quality of discriminative aligner also improved. Nonetheless, when we consider the heuristically symmetrized alignment[2], we observe mixed results. For instance, for the Chinese-English case we observe that AER improves over iterations, but this is the result of a increasingly higher recall rate in detriment of precision. Ayan and Dorr (2006) pointed out that *grow-diag-final* symmetrization tends to output alignments with high recall and low precision. However this does not fully explain the tendency we observed between iterations. The characteristics of the alignment modified by EDMC that lead to larger improvements in mono-directional trainings but a precision drop with symmetrization heuristics needs to be addressed in future work.

Another observation is how the filtering thresholds affect the results. As we can see in Table 3, for Chinese to English word alignment, the largest gain on the alignment quality is observed when the threshold was set to 0.8, while for Arabic to English, the threshold of 0.7 or 0.6 works better. Table 2 shows the precision, recall, and AER of the constraint links used in the constrained EM al-

[1] We run 5, 5, 3, 3 iterations of Model 1, HMM, Model 3 and Model 4 respectively.

[2] We used grow-diag-final-and



(a) Arabic-English

(b) English-Arabic

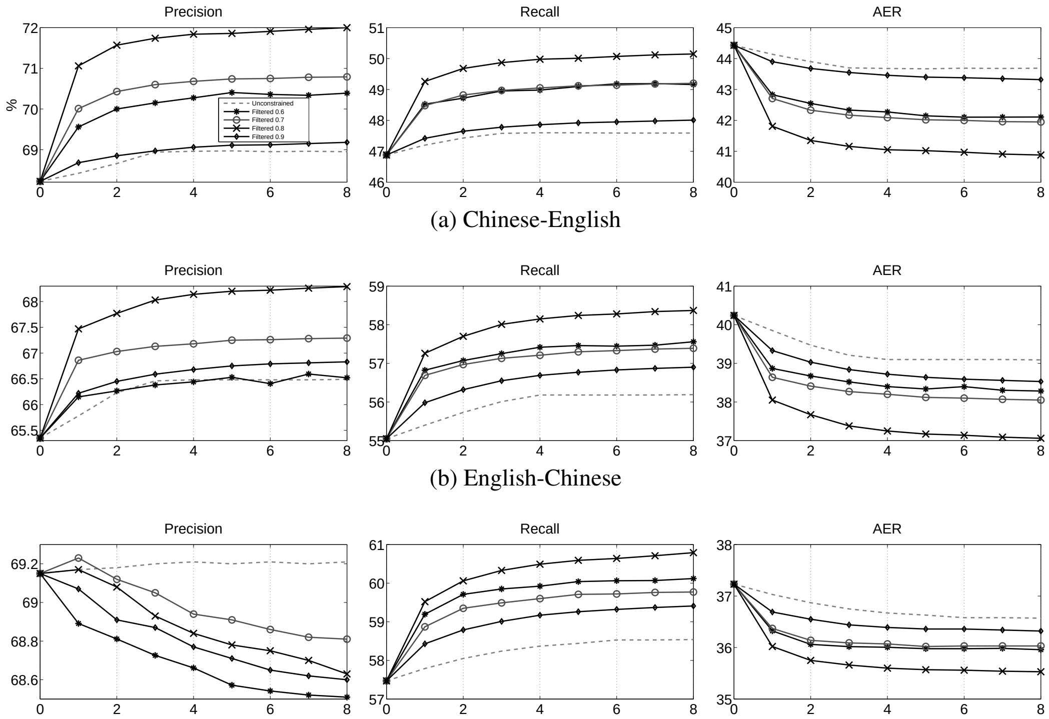
(c) Heuristically-symmetrized

Figure 3: Alignment qualities of each iteration for Arabic-English word alignment task. The grey dash curves represent unconstrained Model 4 training, and the curves with star, circle, cross and diamond markers are constrained EM alignments with 0.6, 0.7, 0.8 and 0.9 filtering thresholds respectively.

		Source-Target			**Target-Source**			**Heuristic**			**Discriminative**		
		P	R	AER	P	R	AER	P	R	AER	P	R	AER
Ch	BL	68.22	46.88	44.43	65.35	55.05	40.25	69.15	57.47	37.23	67.45	59.77	36.62
	NC	+0.73	+0.71	-0.74	+1.14	+1.14	-1.15	**+0.06**	+1.07	-0.66	+0.15	+0.64	-0.42
	0.6	+2.17	+2.28	-2.32	+1.17	+2.51	-1.97	-0.64	+2.65	-1.27	-0.39	+1.89	-0.87
	0.7	+2.57	+2.32	-2.48	+1.94	+2.34	-2.19	*-0.34*	+2.30	-1.20	*-0.28*	+1.60	-0.76
	0.8	**+3.78**	**+3.27**	**-3.55**	**+2.94**	**+3.32**	**-3.18**	*-0.52*	**+3.32**	**-1.70**	**+0.69**	**+0.14**	**-0.89**
	0.9	+0.98	+1.13	-1.11	+1.48	+1.85	-1.71	*-0.55*	+1.94	-0.90	*-0.58*	+1.45	-0.54
Ar	BL	58.41	50.42	45.88	59.08	64.84	38.17	60.35	66.99	36.50	68.93	63.94	33.66
	NC	+2.98	+2.92	-2.96	+1.40	+2.06	-1.70	+0.97	+2.14	-1.49	*-0.87*	+2.37	-0.83
	0.6	+6.69	**+8.02**	-7.47	+3.45	**+6.70**	**-4.90**	+2.62	**+4.71**	-3.55	+0.58	-0.55	+0.03
	0.7	**+8.38**	+7.93	**-8.16**	**+3.65**	+5.26	-4.38	**+2.83**	+4.70	**-3.67**	**+2.46**	*-0.42*	-0.88
	0.8	+6.48	+6.27	-6.39	+2.18	+3.54	-2.80	+1.81	+3.81	-2.70	+1.67	+2.30	-2.01
	0.9	+4.02	+4.07	-4.07	+1.70	+3.10	-2.33	+0.62	+3.82	-2.03	+1.33	**+2.70**	**-2.06**

Table 3: Improvement on word alignment quality on small corpus after 8 iterations. BL stands for baseline, and NC represents unconstrained Model 4 training, and 0.9, 0.8, 0.7, 0.6 are the thresholds used in alignment link filtering.

gorithm, the numbers are averaged across all iterations, the actual numbers of each iteration only have small differences. Although one might expect that the quality of resulting alignment from constrained EM be proportional to the quality of constraints, from the numbers in Table 2 and 3, we are not able to induce a clear relationship between them, and it could be language- or corpus-dependent. However, in practice we nonetheless use a held-out test set to tune this parameter. The



(a) Chinese-English

(b) English-Chinese

(c) Heuristically-symmetrized

Figure 4: Alignment qualities of each iteration for Chinese-English word alignment task. The grey dash curves represent unconstrained Model 4 training, and the curves with star, circle, cross and diamond markers are constrained EM alignments with 0.6, 0.7, 0.8 and 0.9 filtering thresholds respectively.

	Ch-En			En-Ch			Heuristic			Discriminative		
	P	R	AER	P	R	AER	P	R	AER	P	R	AER
BL	73.51	50.14	40.38	68.82	57.66	37.31	72.98	60.23	34.01	72.10	61.63	33.55
NC	73.23	50.38	40.30	68.30	58.00	37.27	72.39	60.99	33.80	72.07	61.81	33.45
0.8	76.27	52.90	37.53	70.26	60.26	35.11	72.75	63.49	32.19	72.64	63.29	32.35

Table 4: Improvement on word alignment quality on moderate-size corpus, where BL and NC represents baseline and non-constrained Model 4 training

relationship between quality of constraints and alignment results is an interesting topic for future research.

5.2 Effects on translation quality

In this experiment we run the whole machine translation pipeline and evaluate the system on BLEU score. We used the corpus LDC2006G05 which contains 25 million words as training set, the same discriminative tuning set as previously used (100 sentence pairs) and the remaining 21,763 sentence pairs from the hand-aligned corpus of the previous experiment are held-out test set for alignment qualities. A 4-gram language model trained from English GigaWord V1 and V2 corpus was used. The AER scores on the held-out test set are also provided for every iteration. Based on the observation in last experiment, we adopt the filtering threshold of 0.8.

Similar to previous experiment, the heuristically symmetrized alignments have lower precisions than their EMDC counterparts, however the gaps are smaller as shown in Table 4. We observe 2.85 and 2.21 absolute AER reduction on two directions, after symmetrization the gain on AER is 1.82. Continuing Model 4 training appears to have minimal effect on AER, and the improve-

I	**M**	**NIST**							**GALE**				
		mt06	**mt02**	**mt03**	**mt04**	**mt05**	**mt08**	*ain*	**db-nw**	**db-wb**	**dd-nw**	**dd-wb**	*aia*
0	G	31.00	31.80	29.89	32.63	29.33	24.24		26.92	24.48	28.44	24.26	
1	D	30.65	31.60	30.04	32.89	29.34	24.52	0.12	27.43	24.72	28.32	24.30	0.14
	G	31.35	31.91	30.35	32.75	29.40	24.16	0.15	27.39	24.50	28.22	24.60	0.15
2	D	**31.61**	32.31	30.40	33.06	29.49	24.11	0.33	**28.17**	24.42	28.58	24.36	0.34
	G	31.14	31.94	30.42	32.86	29.49	24.15	0.20	27.31	24.51	27.50	24.02	0.03
3	D	31.29	**32.39**	30.28	33.19	29.60	24.41	0.43	27.64	**25.32**	28.55	24.71	0.47
	G	30.94	31.95	30.15	32.71	29.38	24.22	0.12	27.63	24.61	28.80	25.05	0.29
4	D	30.80	32.04	30.51	33.24	29.49	24.61	0.46	27.61	25.27	28.72	24.98	0.53
	G	30.68	31.81	30.33	33.05	29.28	24.41	0.26	27.20	24.79	28.43	24.50	0.24
5	D	30.93	31.89	29.96	32.89	29.37	**24.50**	0.17	27.75	24.50	**29.05**	24.90	0.33
	G	31.16	32.28	**30.72**	**33.30**	**29.83**	24.30	**0.51**	27.32	25.05	28.60	**25.44**	**0.54**

Table 5: Improvement on translation alignment quality on moderate-size corpus, The column *ain* shows the average improvement of BLEU scores for all NIST test sets (excluding the tuning set MT06), and column *aia* is the average improvement on all unseen test sets. The column *M* indicates the alignment source, *G* means the alignment comes from generative aligner, and *D* means discriminative aligner respectively. The number of iterations is shown in column *I*.

ment mainly comes from the constraints.

In the experiment, we use the Moses toolkit to extract phrases, tune parameters and decode. We use the NIST MT06 test set as the tuning set, NIST MT02-05 and MT08 as unseen test sets. We also include results for four additional unseen test sets used in GALE evaluations: DEV07-Dev newswire part (dd-nw, 278 sentences) and Weblog part (dd-wb, 345 sentences), Dev07-Blind newswire part (db-nw, 276 sentences and Weblog part (db-wb, 312 sentences). Table 5 presents the average improvement on BLEU scores in each iteration. As we can see from the results, in all iterations we got improvement on BLEU scores, and the largest gain we have gotten is on the fifth iteration, which has 0.51 average improvement on five NIST test sets, and 0.54 average improvement across all nine test sets.

6 Conclusion

In this paper we presented a novel training scheme for word alignment task called EMDC. We also presented an extension of GIZA++ that can perform constrained EM training. By integrating it with a CRF-based discriminative word aligner and alignment link filtering, we can improve the alignment quality of both aligners iteratively. We experimented with small-size Chinese-English and Arabic English and moderate-size Chinese-English word alignment tasks, and observed in all four mono-directional alignments more than 3% absolute reduction on AER, with the largest improvement being 8.16% absolute on Arabic-to-English comparing to the baseline, and 5.90% comparing to Model 4 training with the same numbers of iterations. On a moderate-size Chinese-to-English tasks we also evaluated the impact of the improved alignment on translation quality across nine test sets. The 2% absolute AER reduction resulted in 0.5 average improvement on BLEU score.

Observations on the results raise several interesting questions for future research, such as 1) What is the relationship between the precision of the constraints and the quality of resulting alignments after iterations, 2) The effect of using different discriminative aligners, 3) Using aligners that explicitly model empty words and null alignments to provide additional constraints. We will continue exploration on these directions.

The extended GIZA++ is released to the research community as a branch of MGIZA++ (Gao and Vogel, 2008), which is available online[3].

Acknowledgement

This work is supported by NSF CluE Project (NSF 08-560) and DARPA GALE project.

[3] Accessible on Source Forge, with the URL: `http://sourceforge.net/projects/mgizapp/`

References

Ayan, Necip Fazil and Bonnie J. Dorr. 2006. Going beyond aer: an extensive analysis of word alignments and their impact on mt. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 9–16.

Blunsom, Phil and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72.

Brown, Peter F., Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

Callison-Burch, C., D. Talbot, and M. Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 175–183.

Fraser, Alexander and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 769–776.

Fraser, Alexander and Daniel Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 51–60.

Gao, Qin and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Proceedings of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*, pages 49–57.

Gao, Qin and Stephan Vogel. 2010. Consensus versus expertise : A case study of word alignment with mechanical turk. In *NAACL 2010 Workshop on Creating Speech and Language Data With Mechanical Turk*, pages 30–34.

Gao, Qin, Nguyen Bach, and Stephan Vogel. 2010. A semi-supervised word alignment algorithm with partial manual alignments. In *In Proceedings of the ACL 2010 joint Fifth Workshop on Statistical Machine Translation and Metrics MATR (ACL-2010 WMT)*.

Guzman, Francisco, Qin Gao, and Stephan Vogel. 2009. Reassessment of the role of phrase extraction in pbsmt. In *The twelfth Machine Translation Summit*.

Huang, Fei. 2009. Confidence measure for word alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 932–940.

Ittycheriah, Abraham and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 89–96.

Liu, Yang, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 459–466.

Moore, Robert C. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 81–88.

Niehues, Jan. and Stephan. Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25.

Och, Franz Joseph and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 1:29, pages 19–51.

Taskar, Ben, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM based word alignment in statistical machine translation. In *Proceedings of 16th International Conference on Computational Linguistics)*, pages 836–841.