RC24584 (W0806-056) June 13, 2008
Computer Science

IBM Research Report

A Distortion Model for Arabic to English Maximum Entropy Word Alignment

Abhishek Arun*
School of Informatics
University of Edinburgh
Edinburgh, UK

Abraham Ittycheriah
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

*Work performed while at IBM Research

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

A Distortion Model for Arabic to English maximum entropy word alignment

Abhishek Arun*
School of Informatics
University of Edinburgh
Edinburgh, UK
a.arun@sms.ed.ac.uk

Abraham Ittycheriah
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
abei@us.ibm.com

Abstract

In this work, we present a novel distortion model for an Arabic to English maximum entropy word aligner. In constrast to the distortion model of (Ittycheriah and Roukos, 2005), our model is integrated with the observation model allowing the parameters to be estimated jointly. To have more robust estimates, the distortion model is parametrized using linguistically motivated features such as segmentation and WordNet information. The resulting aligner gives significant improvements over a strong baseline on two different data sets. We also present preliminary work on a self-trained alignment model, which improves alignment quality even further.

1 Introduction

The typical processing pipeline of a phrase-based statistical machine translation (SMT) system consists of two steps. In the first step, word alignments are extracted from a sentence-aligned parallel corpora. In the second step, statistics over the word alignments are used to decode test sentences. In this work, we focus on the first task.

Generally, generative probabilistic models such as the IBM models 1-5 (Brown et al., 1993) are used to produce word alignments with increasing algorithmic complexity and performance. These IBM models and more recent refinements (Moore, 2004) as well as algorithms that bootstrap from these models like the HMM algorithm described in (Vogel et al., 1996) are unsupervised algorithms.

Recently, a flurry of work (Ittycheriah and Roukos, 2005; Taskar et al., 2005; Moore, 2005; Fraser and Marcu, 2006; Moore et al., 2006; Blunsom and Cohn, 2006) has shown that provided the availability of some manually annotated word aligned data as training material, discriminatively trained models can outperform the alignment accuracy of the unsupervised models.

This paper is an extension of the Maximum Entropy (ME) aligner presented in (Ittycheriah and Roukos, 2005). In Section 3, we investigate the use of syntactic features in the alignment model . However, these features do not help improve alignment accuracy. In Section 4, we address a deficiency of the ME aligner by incorporating a distortion model whose parameters are estimated jointly with the observation model. We evaluate our model on the Arabic to English alignment tasks on Sakhr and the MT 03 data sets showing significant improvements on both.

While our labeled training data is relatively small (14.5K sentence pairs), we have access to an unlabeled parallel corpus of almost 600K sentence pairs. In Section 5, we present some self-training experiments, in which the labeled training set is augmented with the unlabeled data to create a larger training set from which a new alignment model is estimated. We show that this semi-supervised model produces alignments of better quality on both test sets.

2 Maximum Entropy aligner

The ME aligner of (Ittycheriah and Roukos, 2005) probabilistically models link decisions between source and target words in a given sentence pair. Figure 1 shows a sentence pair where the top sequence is considered the source sequence and the bottom sequence the target sequence. Each sequence can have auxiliary information such as Arabic segmentation or English WordNet (Miller, 1990) information as shown. Each target word has a link l_i which indicates which source position it links to. The range of l_i is from 0 to K and there are M of these links. The source word position 0 is used to indicate NULL

*This research was conducted during the author's internship at IBM Research

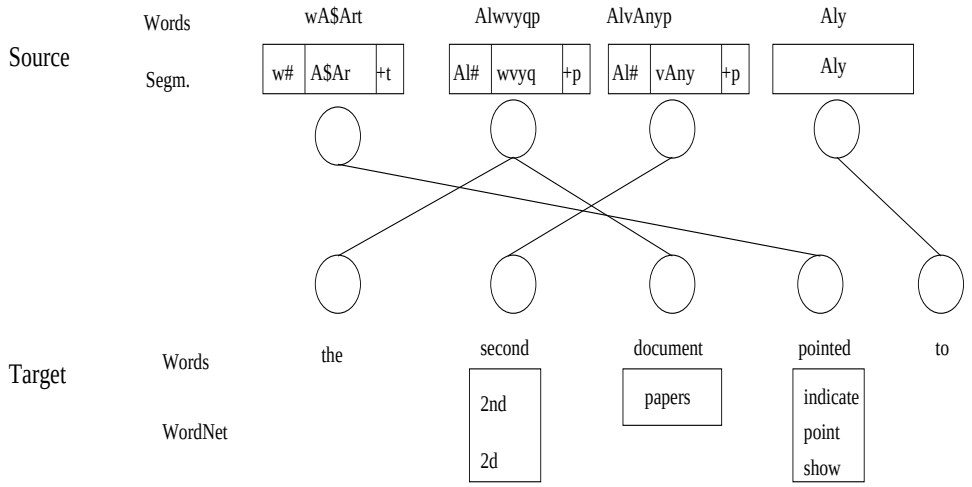


Figure 1: Alignment example.

which we imagine gives rise to unaligned (spontaneous) English words. A valid link configuration has M links.

Define $\mathcal{L}$ to be the set of all possible valid link configurations, and L to be a member of that set. The aim is to maximize the alignment probability by finding the optimum link configuration L_{opt},

$$\begin{aligned} p(L_{\text{opt}}|S,T) &= \arg\max_{L\in\mathcal{L}} p(L|S,T) \\ &= p(l_i^M|t_1^M, s_1^K) \\ &= \prod_{i=0}^{M} p(l_i|t_1^M, s_1^K, l_1^{i-1}). \end{aligned}$$

Note that in the outlined link model, many source words can align to the same target word, but a target word can align to at most one source word (including NULL). Since we are interested in aligning unsegmented Arabic words and typical words have a few affixes to indicate for example pronouns, definiteness, prepositions and conjunctions while in English these are separate words, the unsegmented Arabic words serve as states in the search algorithm with English words being aligned to them.

The link model is factored into a distortion model and an observation model, so that the distortion model computation, which uses information available on the search lattice, is simplified during the search process.

$$p(L|S,T) = \frac{1}{Z}\prod_{i=0}^{M} p(l_i|l_{i-1})^{\alpha} p(l_i|t_1^M, s_1^K, l_1^{i-1})^{1-\alpha}.$$

where Z is the normalizing constant.

2.1 Distortion Model

The distortion model is a fixed distribution that tries to capture the largely locally monotonic nature of word alignments. The model keeps the alignments close together by penalizing alignments in which adjacent words in the target language are attached to distant source words. Also, it penalizes many target words coming from the same Arabic state via a state visit penalty. It has the following parametric form :

$$p(l_i|l_{i-1}) = \frac{1}{Z(l_{i-1})}\left[\frac{1}{\text{dist}(l_i, l_{i-1})} + \frac{1}{ns(l_i)}\right] \quad (1)$$

where $ns(i)$ represents the state visit penalty for state i, $Z(l_{i-1})$ is the normalization constant and

$$\text{dist}(l_i, l_{i-1}) = \min(|l_i - l_{i-1}|, |l_i - f_i|) + a.$$

Here a is a penalty for a zero distance transition and is set to 1 in the experiments below. The min operator chooses the lowest cost transition distance either from the previous state or the frontier state, f_i, which is the right most state that has been visited. This is a language specific criteria and intended to model the adjective noun reversal between English and Arabic. Once the current noun phrase is completed, the next word often aligns to the state just beyond frontier state. As an example, in Figure 1, the verb 'pointed' aligns to the first Arabic word 'wA$Art', and aligning the 'to' to its Arabic counterpart 'Aly' would incur normally a distance of 3 but with the frontier notion it incurs only a penalty of 1 on the hypothesis that aligns the word 'second' to 'AlvAnyp'. In this alignment with the frontier notion, there are only distance 1 transitions, whereas the traditional shapes would incur a penalty of 2 for alignment of 'pointed' and a penalty of 3 for the word 'to'.

The state visit penalty, $ns(i)$ is the distance between the English words aligned to this state times the number of state visits[1]. This penalty controls the fertility of the Arabic words. To determine the English words that aligned to the Arabic position, the search path is traced back for each hypothesis and a sufficiently large beam is maintained so that alignments in the future can correct past alignment decisions. This penalty allows English determiners and prepositions to align to the Arabic content word while penalizing distant words from aligning to the state.

2.2 Observation Model

The observation model measures the linkage of the source and target using a set of feature functions defined on the words and their context. In Figure 1, an event is a single link from an English word to an Arabic state and the event space is the sentence pair. We use the maximum entropy formulation (e.g. (Berger et al., 1996)),

$$f = \psi(l_i)$$
$$h = \left[t_1^{i-1}, s_1^K\right]$$
$$p(f|h) = \frac{1}{Z(h)} \exp \sum_i \lambda_i \phi_i(h, f),$$

where Z(h) is the normalizing constant,

$$Z(h) = \sum_f \exp \sum_i \lambda_i \phi_i(h, f).$$

and $\phi_i(h, f)$ are binary valued feature functions. The function ψ selects the Arabic word at the position being linked or in the case of segmentation features, one of the segmentations of that position. We restrict the history context to select from the current English word and words to the left as well as the current word's WordNet (Miller, 1990) synset as required by the features defined below. As in (Cherry and Lin, 2003), the above functions simplify the conditioning portion, h by utilizing only the words and context involved in the link l_i. Training is done using the IIS technique (Della Pietra et al., 1995) and convergence often occurs in 3-10 iterations. The five types of features which are utilized in the system are described below.

Phrase to phrase (for example, idiomatic phrases) alignments are interpreted as each English word coming from each of the Arabic words.

2.2.1 Features

The features used in the baseline model are :

- Source/target word pair. Since training data is limited, there is a significant out of vocabulary (OOV) issue in the model. All singletons are mapped to an unknown word class in order to explicitly model connecting unknown words.
- Source segmentation features. These features are useful in aligning unknown words since stems might have been seen in the training corpus with other prefixes or suffixes.
- Target WordNet features. English nouns, adjectives, adverbs and verbs are mapped to their WordNet synset id. These features helps combat data sparseness on the English side by clustering words falling into the same synsets and are useful to increase the aligner's recall.
- Spelling features. These features are designed primarily to link unknown names. They are only applied on unknown words and measures the string kernel distance between English and Arabic romanized words.
- Dynamic features. These features are defined over the search lattice and are fired when the previous source and target word pair are linked. For more details, refer to (Ittycheriah and Roukos, 2005)

2.3 Smoothing the Observation Model

Since the annotated training data for word alignment is limited and a much larger parallel corpus is available for other aligners, the observation model is smoothed with an IBM Model 1 estimate,

$$p(l_i|t_1^M, s_1^K) = \frac{1}{Z} p_{\mathrm{ME}}(l_i|t_1^M, s_1^K)^{\beta} p_{\mathrm{M1}}(s|t_i)^{1-\beta}.$$

where β is set to 0.9 in the experiments below. In the equation above, the s represents the Arabic word that is being linked from the English word t_i.

2.4 Search Algorithm

A beam search algorithm is utilized with the English words consumed in sequence and the Arabic word positions serving as states in the search process. In order to take advantage of the transition model described above, a large beam must be maintained. To see this, note that English words often repeat in a sentence and the models will tend to link the word to all Arabic positions which have the same Arabic content. In traditional algorithms, the Markov assumption is made and hypothesis are merged if they have the same history in the previous time step. However, here we maintain all hypotheses and merge only if the paths are same for 30 words which is the average sentence length.

[1]We are overloading the word 'state' to mean Arabic word position.

3 Adding syntax to the model

While the baseline distortion model can capture *local movement*, it ignores the syntactic structure of the source and target sentences. Numerous unsupervised alignment models (Wu, 1995; Cherry and Lin, 2003; Lopez and Resnik, 2005; DeNero and Klein, 2007) have looked at ways of capturing these *structural movements*. Typically, this involves using syntactic knowledge on either the source side or the target side or both. (Lopez and Resnik, 2005) propose a tree distortion model for an HMM-based aligner that, given a dependency parse on the target side, conditions the alignment decision on the tree distance between each pair of target words. However, the tree distortion model fails to improve upon the HMM's surface distortion model. More recently, (DeNero and Klein, 2007) present a generative syntax-sensitive distortion model where the cost of transition between string positions is conditioned on the number of moves needed while walking along the target side constituent tree. This model too is unable to outperform the basic HMM distortion model as measured by Aligned Error Rate (AER) metric.

3.1 Phrasal Cohesion

Our effort to incorporate an element of syntax in the MaxEnt aligner is based around the notion of **phrasal cohesion** (Fox, 2002) i.e the tendency of words in a source side constituent to align to words in the equivalent target side constituent. (Fox, 2002), in an empirical study for the French-English language pair, show that while phrasal cohesion is not systematic, it occurs often enough to justify being taken into account.

Let us illustrate the notion of phrasal cohesion by looking at part of a hand-aligned sentence pair annotated with automatically generated parses on both source and target sides (Figure 2(a)). A striking fact is that the English side parse tree is right branching while the Arabic one is flat. In spite of this difference in annotation, cohesion is quite strong - e.g. the prepositional phrase 'on Tuesday' is aligned to the Arabic temporal noun phrase 'ywm Alvlava'. However, in the system generated alignment this cohesion is broken when the preposition 'on' crosses a bracketing boundary to align to 'Ely' instead of 'ywm'. We would like to discourage these *crossing bracket* alignments. Our proposed solution is in the form of a syntactic feature that is incorporated in the observation model, and whose weight is learned from labeled data.

Formally:

$$\phi_{cb}(l_j, l_{j-1}, E, A) = \begin{cases} 1 & \text{if ParseChunk}(E[j]) = \\ & \text{ParseChunk}(E[j-1]) \\ & \& \\ & \text{ParseChunk}(A[l_j]) = \\ & \text{ParseChunk}(A[l_{j-1}]) \\ 0 & \text{otherwise} \end{cases}$$

where ParseChunk(w) is a pair defining the span of the immediate non-unary constituent w is part of. For example, ParseChunk(*their*) = (21,23) and ParseChunk(*Tuesday*) = (27,28). The feature fires when the words in the proposed link are under the same respective non-terminals as the words in the previous link. This feature is therefore a *dynamic* feature.

While we have human aligned sentence pairs, we do not access to gold standard parses. Instead, the English sentences are parsed using a maximum entropy parser trained on the Penn Treebank while the Arabic parses were created using a statistical parser trained on the Arabic Treebank. (Fox, 2002) show that phrasal cohesion is most often violated in verb phrases - we limit the application of the crossing bracket feature to only NPs and PPs. The feature is conditioned on the Arabic and English side non-terminals as well as the POS tags of the words being linked.

3.2 Results

We trained the MaxEnt aligner on 14500 hand aligned sentence pairs and evaluated it on two different test sets - the first 50 sentences of MT03 Evaluation test set and 200 randomly selected sentences from the Sakhr news site which were manually word aligned by an in-house annotator (Refer to (Ittycheriah and Roukos, 2005) for the annotation guidelines)

In order to measure alignment performance, we use the standard AER measure (Och and Ney, 2000) but consider all links as sure. This measure is then related to the F-measure which can be defined in terms of precision and recall as

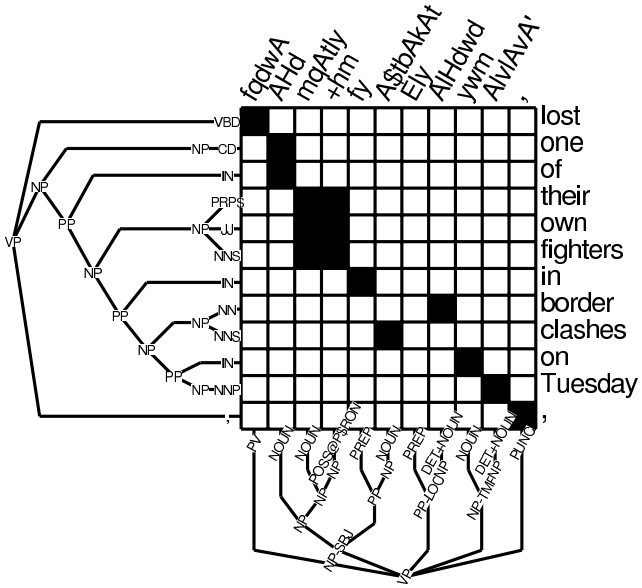
Precision The number of correct word links over the total number of proposed links.

Recall The number of correct word links over the total number of links in the reference.

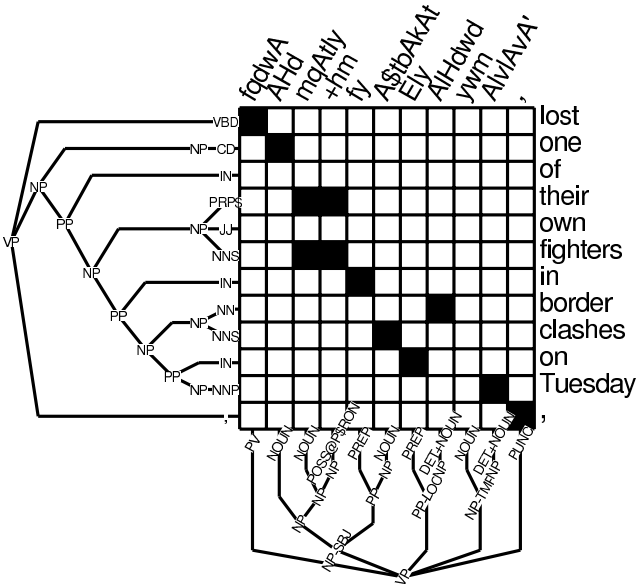
and the usual definition of the F-measure,

$$F = \frac{2PR}{(R+P)}$$

and define the alignment error as AER $= 1 - F$. In this paper, we report our results in terms of F-measure over aligned links. Note that links to the



(a) Manually aligned sentence with parse information



(b) Automatically aligned sentence with parse information

	Sakhr	MT 03
Model	F-Score	F-Score
Baseline	80.2	**87.6**
Crossing bracet feat, cnt ≥ 5	**80.3**	87.3
Crossing bracket feat, all	80.2	87.3

Table 1: F-measure for baseline model, model with crossing bracket features with count cutoff of 5, and model with all crossing bracket features.

NULL state (unaligned English words) are not included in the F-measure.

We compare the baseline MaxEnt aligner to two different syntax-enhanced aligners, one where all crossing bracket features are kept and the other in which we prune out crossing bracket features appearing fewer than 5 times.

The baseline alignment performance on Sakhr(80.2) is much lower than that on MT03. In contrast to MT03 where precision and recall are quite balanced (P=88.2% and R=87.1%), the Sakhr recall(73.6%) is much lower than precision (88.2%). A possible reason for this imbalance is that the Sakhr data is less similar to the training data than the MT03 data, leading to an out of vocabulary (OOV) issue.

Adding the syntactic feature to the model leads to a small but insignificant improvement in the Sakhr dataset, but a deterioration on MT03 performance, indicating that this feature is not very useful. We believe that this is due to two main reasons.

Firstly, the automatic parses are often of poor quality thus limiting their usefulness. Secondly, the feature suffers from data sparseness. For example, the Arabic word 'AlvlAva" is never seen in the training data aligned to an English word in a PP and therefore the feature is unable to correct the wrong alignment in Figure 2(b). In another set of experiments, we dropped the POS tags from the history of the feature, but results were no better.

4 A log-linear distortion model

The fixed distortion model of the baseline aligner is a stand alone distribution with one parameter, the zero distance transition penalty, set by hand to optimize performance on the development set. We would like to incorporate a distortion model in the log-linear framework so that its parameters are learned jointly with the observation model and directly from the training data. This in effect means turning the MaxEnt aligner into a Maximum Entropy Markov Model (MEMM) aligner. A MEMM is an application of the maximum entropy classifier to sequence labeling (Ratnaparkhi, 1996). The structure of the MEMM is inspired by the HMM; each link decision is conditioned on the previous link and the observation.

The distortion model is implemented in form of a feature function such that the mathematical formulation of the model (Eq 2.2) remains the same. The distortion feature, a dynamic feature, predicts a link decision given the previous link, the source sentence and the target sentence. This link decision can be viewed as jointly predicting the source word and the jump (distance from source word aligned to previous target word), given the target word under consideration.

Formally, the feature is a triplet:

$$\phi(f_l(l_j, l_{j-}), f_E(E_j), f_A(A_{l_j}))$$

where f_l is a distortion function, f_E is a function that maps an English word to an alternative representation and f_A a similar mapping over Arabic words.

4.1 Distortion function

The distortion function $f_l(l_j, l_{j-})$ measures the jump given the previous links l_{j-}. Similar to the fixed distortion model, we employ the frontier node concept.

$$f_l(l_j, l_{j-}) = \min(l_j - l_{j-1}, l_j - f_j)$$

In the case of unaligned target words, we map them to a special jump distance (*ToNull*). When the previous target word is unaligned, we experimented with a few options. In one case, we map the jump to a special token (*FromNull*). Another choice was to compute the jump distance from the source word linked to the last aligned target word (*DistLast*). In a third way of parametrizing the function, we compute the jump from the source word aligned to the frontier word (*DistFront*). In order to have less sparse statistics, the jump is binned to a fixed number of buckets. In one set of experiments, we had a bin for every distortion in the range [-5, 5] (*RegBin*). Distortions greater than 5 are grouped in the 5 bucket and similarly jumps less than -5 get grouped in the -5 bucket. In another set of experiments, we had a reduced number of bins with a bin for each distortion in the range [-2,2] (*RedBin*).

4.2 English mapping function

Simply using the English words themselves would lead to over-fitting the training data. We would like our feature to generalize and therefore investigated the use of a number of alternate formulations of the English mapping function f_E.

The function $f_{E1}(e)$ maps the English word e to itself if e is one of the 2500 most frequent words in our training data, and otherwise maps it to its POS tag. The intuition behind this mapping is to cluster infrequent words into equivalence classes that demonstrate similar distortion behavior e.g verbs.

$f_{E2}(e)$ is another mapping whereby e is mapped to its synset ID, or to itself if the synset ID cannot be found. This mapping allows words to be clustered along equivalent semantic classes.

4.3 Arabic mapping function

The distortion feature jointly predicts the source word to link to and the jump. The feature can be made to only predict the jump by using an Arabic mapping function that ignores the identity of the Arabic word $f_{A0}(a)$. In this way, the distortion feature is effectively unlexicalized.

Obviously, we do not expect the unlexicalized feature to be as beneficial as some form of lexicalization. We experimented with two additional mapping functions, $f_{A1}(a)$ - the identity function and $f_{A2}(a)$ which maps the input word to its word segments and to itself.

4.4 Training the distortion feature

4.5 Results

While adding a distortion model to the log-linear model, we also retain the fixed distortion model from the baseline aligner since it is a useful distortion as well as fertility model. The results for the new proposed models are presented in Table 2

Since the distortion feature gets fired for every proposed link, it improves recall significantly across the board. When the feature is unlexicalized, i.e the link decision depends only on the target word being considered (or its POS tag) and not the source word, precision takes a hit such that the F-Score on MT03 drops by almost 1%. On Sakhr, the increase in recall brought about by the feature is large enough to bring about an improvement in F-Score despite the drop in precision.

When the feature is lexicalized, recall drops in comparison to the unlexicalized feature, but is still higher than the baseline, and there is a large increase in precision. The F-Score on MT03 rises from 86.7% to 87.6% (same as baseline) while in Sakhr, F-Score climbs up to 82.2%.

Using fewer bins for distortion is beneficial but otherwise all the other parametrizations yield almost similar results. The main axis of precision improvement is by varying the way distortion is calculated when the previous link is unaligned. On MT03, moving from *FromNull* to *DistLast* gives a significant boost of 0.5% F-Score and a similar improvement is obtained when going from *DistLast* to *DistFront*. Alternating between F_{E1}/F_{E2} and F_{A1}/F_{A2} does not seem to make much difference in the alignment accuracy. The best distortion models give an absolute improvement of 2.3% for Sakhr and 1% on the MT03 with respect to the baseline.

4.6 Training and feature selection

The models presented in Table 2 retained all the distortion features encountered in training. Frequently in a discriminative training set-up, feature selection is performed to have a smaller model which can be trained more quickly and less prone to over-fitting. We performed a series of feature selection experiments whereby we varied the distortion feature count cutoff for the last model in Table 2. Also, we assessed the impact of the number of IIS training iterations on the final alignment model.

	Sakhr			MT 03		
	P	R	F	P	R	F-Score
Baseline	88.2	73.6	80.2	88.2	87.1	87.6
RegBin, FromNull, f_{E1}, f_{A0}	85.0	77.2	80.9*	84.5	89.0	86.7
RegBin, FromNull, f_{E1}, f_{A1}	88.8	76.5	82.2*	86.4	88.8	87.6
RedBin, FromNull, f_{E1}, f_{A1}	89.0	**76.7**	82.4*	86.5	88.7	87.6
RedBin, DistLast, f_{E1}, f_{A1}	88.7	76.5	82.2*	87.2	89.0	88.1
RedBin, DistFront, f_{E1}, f_{A1}	88.9	76.5	82.3*	87.8	**89.3**	**88.6***
RedBin, DistFront, f_{E1}, f_{A2}	88.9	76.0	81.9*	**88.4**	88.7	**88.6***
RedBin, DistFront, f_{E2}, f_{A2}	**89.3**	76.6	**82.5***	87.9	88.9	88.4*

Table 2: MaxEnt aligner with distortion model. Results on Sakhr and MT03 datasets comparing distortion, Arabic mapping and English mapping functions. * denote statistically significant improvements from baseline

	Sakhr			MT 03			# Features(K)	
	Iter 0	Iter 1	Iter 2	Iter0	Iter 1	Iter 2	Total	Dist
All	82.5*	82.6	82.1	**88.4***	88.2	88.2	226	91
≥ 2	82.7*	82.2	82.2	88.2	88.2	88.3	147	35
≥ 3	**82.9***	82.2	82.0	88.1	87.8	87.9	130	20
≥ 4	82.8*	82.1	81.8	88.2	87.8	88.0	125	14
≥ 5	82.5*	81.9	81.9	87.9	87.8	88.0	121	11

Table 3: Impact of distortion feature selection on Sakhr and MT03, measured across IIS iterations. Also included are the total number and the number of distortion features in the model.

	Baseline			Best model		
	P	R	F	P	R	F
Src Func	75.8	56.2	64.6	84.4	64.7	73.2
Src Cont	88.9	74.7	81.2	89.5	77.4	83.0
Trg Func	87.7	61.6	72.3	88.8	66.5	76.0
Trg Cont	88.3	78.3	83.0	89.4	80.5	84.7

Table 4: Comparison between baseline and best distortion model for source function(Src Func) words, source content(Src Cont) words, target function(Trg Func) words and target content words(Trg Cont)

While for MT03 distortion feature pruning hurts performance, gradually increasing the pruning cutoff improves Sakhr results up to a threshold of 3 before then starting to drop.

In contrast to our previous experiences with IIS training, all but one of the models peaked after one just one iteration. Experiments with varying the Gaussian prior did not yield any improvements.

4.7 Analysis

In table 4, we present a detailed comparison between the performance of the baseline aligner and the distortion-based aligner on the Sakhr dataset.

While the improvements are across the board, the most dramatic ones occur for function words. The new model proposes one extra link every two sentences, with most of these links aligning source function words that were previously unaligned.

In Figure 2, we can see the impact of the distortion model (Row 5 in Table 2). In the reference alignment, the verb *mounted* is aligned to the Arabic verb *SEdt*, whereas in the baseline model these 2 words are unaligned. This is because *SEdt* has not been seen in the training data (maps to unknown word) and no feature gets fired for the correct link. The distortion model) on the other hand has learnt that unknown Arabic words are very likely to link to English past tense verbs with a distortion of 1.

Figure 3 illustrates the impact of using WordNet synset ids to parametrize the English words. The alignment in the middle of Fig 3 is obtained using the model on Row 5 of Table 2. In that model, *struggled* maps to its POS tag *VBN*. However, the correct Arabic word *kAfHt* is never linked to this POS tag in the training data. In the model with WN synsets(Table 2 - Row 7), *struggled* gets mapped to the same synset ID as *fight* and *struggle*. This feature configuration has been seen often in the training data and a distortion of -4 is a very likely jump such that the model proposes the right alignment.

5 A self-trained alignment model

While our labeled data is relatively small, we have access to an unlabeled parallel corpus of almost 600K sentence pairs. An obvious way to exploit the large unlabeled data-set is via bootstrapping methods which belong to the semi or weakly supervised learning (SSL) algorithm family.

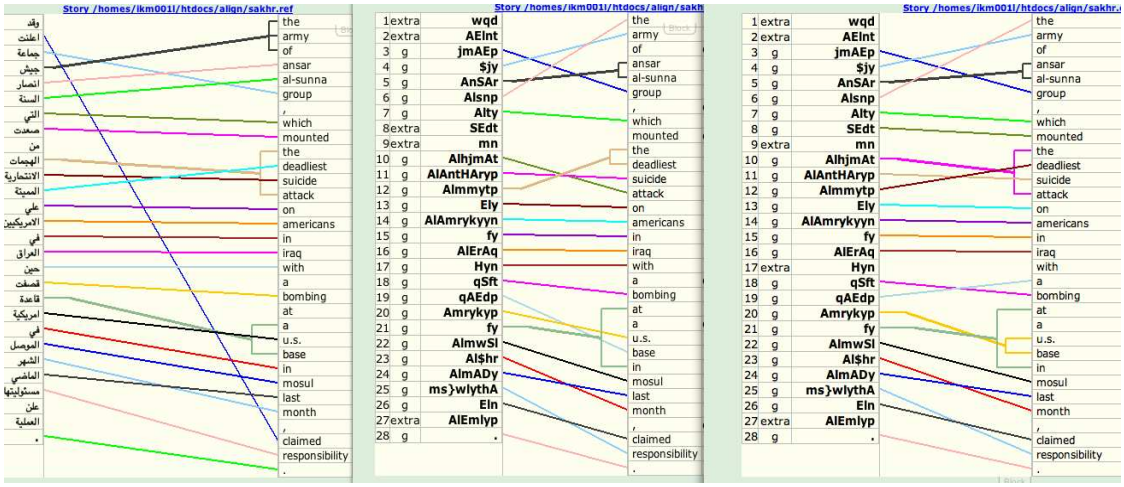


Figure 2: Alignment example. Reference alignment(left), baseline model(middle), distortion model(right)

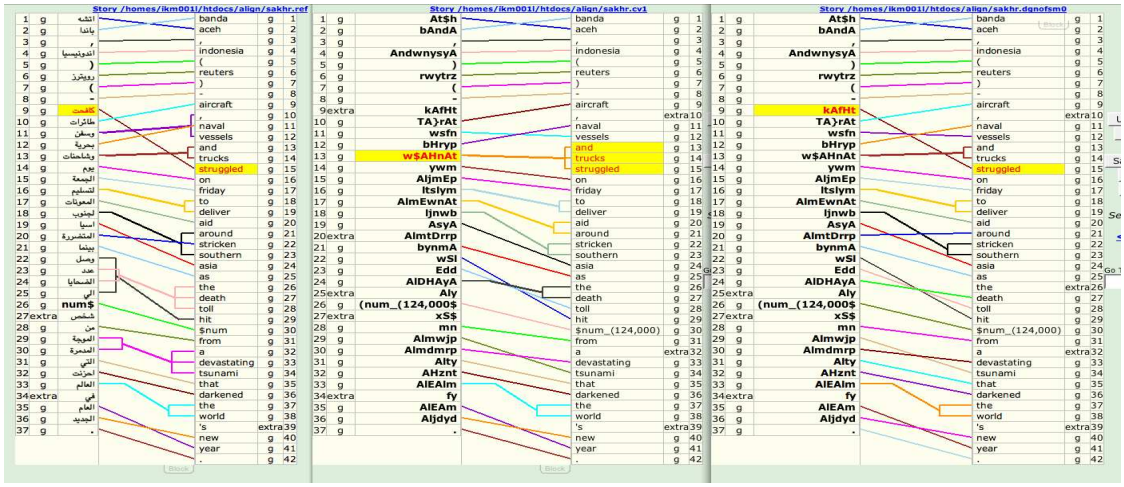


Figure 3: Alignment example. Reference alignment(left), distortion model f_{E1}, f_{A1} (middle), distortion model f_{E2}, f_{A2} (right)

5.1 Bootstrapping methods

Starting from a small set of labeled examples and one of a few weak classifiers, the bootstrapping algorithms aim to improve the system's performance by incorporating unlabeled data into the training set. Two popular bootstrapping methods are self-training and co-training.

5.1.1 Co-training

Co-training (Blum and Mitchell, 1998) works by generating several classifiers trained on the input labeled data, which are then used to tag new unlabeled data. From this newly annotated data, the most confident predictions are sought, and subsequently added to the set of labeled data. This process may continue for several iterations. Co-training has successfully been applied to many NLP tasks such as statistical parsing (Sarkar, 2001), reference resolution (Ng and Cardie, 2003) and part of speech tagging (Clark et al., 2003).

5.1.2 Self-training

A related bootstrapping method is self-training which has been used to refer to a variety of schemes for using unlabeled data. (Ng and Cardie, 2003) implement self-training by bagging and majority voting. A committee of classifiers are trained on the labeled examples, then classify the unlabeled examples independently. Only those examples to which all the classifiers give the same label are added to the training set and those classifiers are retrained. This procedure repeats until a stop condition is met. (Clark et al., 2003) provide a different definition - self-training is a procedure in which "a tagger is re-

trained on its own labeled cache on each round". We adopt this second definition in our work.

A single classifier can carry out its own self-training procedure. This classifier is trained on the initial labeled data and then applied on a set of unlabeled data. Those examples meeting a selection criterion are added to the labeled set and the classifier is retrained on this new labeled data set. This training procedure continues for several rounds.

While both co-training and self-training have shown good results on many tasks, improvements have been highly dependent on the nature of the task, the characteristic of the data and the tuning of parameters. Moreover, on large scale natural language processing tasks, these algorithms have shown limitations (Pierce and Cardie, 2001).

5.2 Previous work in word alignment

We review some previous work using semi-supervised learning methods for the word alignment task. (Callison-Burch et al., 2004) present a mixture model where a generative (IBM Model 4) model trained on a large unlabeled dataset is interpolated with a small amount of automatically word aligned data which is however treated as gold standard. To control the relative contributions of the sentence-aligned and word-aligned data in the parameter estimation procedure, they introduce a mixing weight λ ranging between 0 and 1, with the best results obtained by weighting the hand annotated data at 0.9.

(Callison-Burch et al., 2004) get good improvements on both alignment (AER) and translation quality (BLEU) on the German to English task but their experiments are only a very limited amount of data (16K sentences), raising the question as to whether their approach would work for larger sized datasets.

In (Fraser and Marcu, 2006), expectation-maximization (EM) is used to train a generative model of word alignment from a large parallel text. The generative model is decomposed into several sub-models using independence assumptions. Each sub-model is then used in a log-linear model for word alignment, with the weighs trained on a small set of hand aligned sentences. The training regime iteratively alternated between approximate EM (Neal and Hinton, 1998) and gradient descent until the error rate on a held-out set is minimized. The predicted Viterbi word alignments are then used to train a phrase-based SMT system yielding significant improvements on BLEU for both Arabic-English and French-English

5.3 Machine Translation experiments

Due to time constraints, we only ran a limited number of self-training experiments taking as baseline model the last aligner in Table 2. In our experimental set-up we aligned 600K sentence pairs and re-estimated a MaxEnt model by merging all of the newly labeled data to the original 14.5K hand aligned sentence pairs. To speed up the estimation, we parallelized the MaxEnt training algorithm to run on our cluster. We only ran the self-training algorithm for one round.

	Sakhr			MT 03		
	P	R	F	P	R	F
Iter 1	91.3	77.9	84.1	88.2	88.2	88.2
Iter 2	**91.3**	**78.5**	**84.4**	88.2	**88.9**	88.5
Iter 3	91.2	78.4	84.3	**88.2**	**88.9**	**88.6**

Table 5: Self-training results for 3 iterations of IIS training

Test results are shown in Table 5. Self-training improves performance on both datasets, with the amelioration on Sakhr quite pronounced on both precision and recall measures. For MT03, improvements are seen only on precision. In contrast to the supervised models, best self-training results are obtained after 2 and 3 iterations of IIS training.

6 Future Work

Previous work has repeatedly shown that improved word alignments are no guarantee for improvements in translation quality. Due to time limitations, we were unfortunately unable to carry out any translation experiments, leaving them for future work.

Also, we would like to carry out further self-training experiments such as running the algorithm for more than one rounds and varying the criteria for choosing the newly labeled examples to add to the training set.

7 Conclusion

We presented a number of experiments to improve a baseline MaxEnt Arabic to English word aligner. While our experiments to use syntactical features were not succesful, adding a distortion feature in the log-linear model brought about significant improvements, mainly by boosting recall. In order to exploit the large amount of unlabeled data at our disposal, we started looking into self-training the aligner. Our preliminary results were encouraging with increases in precision on both our test sets. The resulting best aligner improved the absolute F-Score by 4.2% on Sakhr and 1% on the MT03 dataset.

References

Adam L. Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy ap-

proach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, pages 92–100.

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

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chris Callison-Burch, David Talbot, and Miles Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 175–182, Barcelona, Spain, July.

Colin Cherry and Dekang Lin. 2003. A probability model to improve word alignment. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Sapporo, Japan.

S. Clark, J. Curran, and M. Osborne. 2003. Bootstrapping pos taggers using unlabelled data.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1995. Inducing features of random fields. *Technical Report, Department of Computer Science, Carnegie-Mellon University, CMU-CS-95-144*, May.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 304–3111, Morristown, NJ, USA. Association for Computational Linguistics.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 769–776, Sydney, Australia, July. Association for Computational Linguistics.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 89–96, Morristown, NJ, USA. Association for Computational Linguistics.

Adam Lopez and Philip Resnik. 2005. Improved HMM alignment models for languages with scarce resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 83–86, Ann Arbor, Michigan, June. Association for Computational Linguistics.

G. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Robert C. Moore, Wen-tau Yih, and Andreas Bode. 2006. Improved discriminative bilingual word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 513–520, Sydney, Australia, July. Association for Computational Linguistics.

Robert C. Moore. 2004. Improving IBM Word-Alignment Model 1. In *42nd Annual Meeting of the Association for Computational Linguistics*, pages 518–525, Barcelona, Spain.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

R. M. Neal and G. E. Hinton. 1998. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

Vincent Ng and Claire Cardie. 2003. Weakly supervised natural language learning without redundant views.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China.

David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-2001)*.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Ben Taskar, Lacoste-Julien Simon, and Klein Dan. 2005. A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 73–80, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM BasedWord Alignment in Statistical Machine Translation. In *Proc. of the 16th Int. Conf. on Computational Linguistics (COLING 1996)*, pages 836–841, Copenhagen, Denmark, August.

Dekai Wu. 1995. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 244–251, Morristown, NJ, USA. Association for Computational Linguistics.