Refining Word Alignment with Discriminative Training

Nadi Tomeh and **Alexandre Allauzen** and **Guillaume Wisniewski** and **François Yvon**
LIMSI/CNRS and Univ. Paris Sud, France
BP 133, 91403 Orsay Cedex
Firstname.Lastname@limsi.fr

Abstract

The quality of statistical machine translation systems depends on the quality of the word alignments that are computed during the translation model training phase. IBM alignment models, as implemented in the GIZA++ toolkit, constitute the *de facto* standard for performing these computations. The resulting alignments and translation models are however very noisy, and several authors have tried to improve them. In this work, we propose a simple and effective approach, which considers alignment as a series of independent binary classification problems in the alignment matrix. Through extensive feature engineering and the use of stacking techniques, we were able to obtain alignments much closer to manually defined references than those obtained by the IBM models. These alignments also yield better translation models, delivering improved performance in a large scale Arabic to English translation task.

1 Introduction

The translation quality of phrase-based machine translation systems depends heavily on the quality of the translation model, the so-called *phrase table* consisting of a set of aligned phrase-pairs in mutual translation relationship. Since finding the optimal phrase alignment in parallel sentences is NP-hard (DeNero and Klein, 2008), most practical approaches rely on pre-computed word alignments to restrict the search space and use a heuristic to extract phrase pairs that are consistent with them (Och and Ney, 2003). Phrase extraction therefore boils down to the problem of word alignments, that consists in finding a many-to-many correspondence between source and target words of a bilingual sentence-pair. Many approaches have been proposed to solve this problem. The most widely used in practice are generative IBM models (Brown et al., 1993) which allow to construct directional one-to-many alignments in both translation directions. Theses alignments are then symmetrized during a post-processing step to obtain a many-to-many symmetric alignment. Training these models only requires sentence-aligned bitext and is performed in an unsupervised way with the EM algorithm. This approach has two main caveats, leaving room for improving the alignment quality and, consequently, the translation quality. Firstly, the generative paradigm is not well suited to incorporate arbitrary and possibly interdependent information sources. Secondly, the symmetrization heuristic acts locally at the sentence-pair level and lacks a global view of the entire training corpus.

A natural remedy to the first problem is to use discriminative models, which are able to consider arbitrary features of the involved words. In this framework, the alignment task is casted as a classification problem: a binary classifier predicts, for each possible assignment, whether it should be included or not in the alignment. Discriminative models can also consider predictions provided by other alignment models as features, and therefore constitute a solution to the second problem: by applying these features to learn symmetrization decisions in light of a global view of the data. By applying these ideas (Ayan and Dorr, 2006) obtained promising results. However, their model remains unable to model interactions between alignment decisions which are, intuitively, of great help to correctly prevent or encourage certain configurations in the predicted alignment. To overcome this shortcoming, we propose to extend their model by introducing a stacked classification layer (Wolpert, 1992) that operates globally and, hence, enables arbitrary features, describing interactions between alignment decisions, to be taken into consideration.

The main contribution of this work is a reexamination of (Ayan and Dorr, 2006) work which we extend in several ways. On the one hand, we present a careful study of the impact of several novel features on the performance; on the other hand, we investigate the use of the stacking technique to improve the alignment quality. By conjoining these techniques, we were able to greatly reduce the AER as compared to previously published work, and to achieve better BLEU results. In this paper, we also contrast alignments obtained by the symmetrization heuristic with those obtained by the discriminative matrix model, in the light of their Alignment Error Rate (AER) and their impact on translation quality as measured by BLEU (Papineni et al., 2002) on NIST MT08 large-scale task.

The rest of the paper is organized as follows: after reviewing the related work in Section 2, we present our approach in Section 3, focusing on the design of our feature set, and on our implementation of stacking. We then present experimental results both in terms of AER and BLEU in Section 4.

2 Related Work

Several discriminative approaches of word alignment have been carried out recently (Cherry and Lin, 2003; Ittycheriah and Roukos, 2005; Liu et al., 2005), attempting to reach a good balance between the expressivity of the model and its complexity (in terms of tractability and the possibility of performing exact inference and learning). In one type of approaches, a word alignment between two sentences is evaluated with a global score using a non-decomposable discriminative scoring function. This scheme enables to take into consideration the complete observation of the sentence-pair and the hypothesized word alignment when extracting features (Moore, 2005). However, as no restriction on the form of considered alignments is imposed, the size of the resulting search space makes the search intractable and requires the application of a heuristic beam search. In (Taskar et al., 2005), tractability of the search problem is achieved by casting the word alignment task as a maximum weighted matching problem. This comes at the price of constraining possible alignments to one-to-one matchings and making local decisions with no global interactions. These limitations are fixed in (Lacoste-Julien et al., 2006), by modeling alignment as a quadratic assignment problem which is NP-hard in general.

In another type of approaches, word alignment is viewed as a classification problem of the cells in the alignment matrix. The scoring function, which is usually the probability of the hypothesized alignment, is decomposable under some independence assumptions. In (Blunsom and Cohn, 2006) word alignment is considered as a sequence labeling problem, in which, source words are tagged with target positions using a linear chain conditional random field (CRF). The linear chain assumption enables exact inference and training. However the underlying graphical structure is similar to the directed hidden Markov model (HMM) used in generative alignment, hence only one-to-many alignments can be obtained, and the symmetrization step is still needful. In (Niehues and Vogel, 2008), the alignment matrix is directly modeled by a more complex CRF structure, which allows to get rid of the symmetrization step, at the expense of an approximate inference and a complicated two-step training. Many of these discriminative models do not entirely dispense with the generative models, but rather integrate their predictions as supplementary features.

3 Maximum Entropy for Alignment Matrix Modeling

In this section, we present the task of word alignment as a binary classification problem, in which we model the alignment matrix directly. We also explain how to improve the expressivity of the model using a *stacked generalization* approach.

3.1 Word Alignment as a Classification Problem

The task of word alignment is to find a many-to-many correspondence between the words of a source sentence $\mathbf{f}_1^I = f_1, f_2, \ldots, f_I$, and a target sentence $\mathbf{e}_1^J = e_1, e_2, \ldots, e_J$. Alignment information between both sentences are represented by an alignment matrix $\mathbf{A} = \{l_{i,j} : 1 \leq i \leq I, 1 \leq j \leq J\}$, in which a particular link $l_{i,j}$ is considered to be *active* if the source word f_i is aligned to the target word e_j, and *inactive* otherwise. Word alignment can be seen as a binary classification task, in which the goal is to predict a class $y \in \{active, inactive\}$ for every candidate link $l_{i,j} \in \mathbf{A}$.

Since the alignment matrix is typically sparse, with a majority of inactive links, the classification task we consider is unbalanced. To avoid learning a biased classifier with high tendency toward labeling all links as inactive, we use a set of input alignments to reduce the set of links to be predicted to a subset of the alignment matrix: a point that has not been proposed by at least one input alignment will be labeled as *inactive*; the others are labeled by the classifier. The union of all input alignments is hence used to reduce the search space and avoid biasing the classifier as in (Ayan and Dorr, 2006; Elming and Habash, 2007). Input alignments are pre-computed separately using GIZA++.

During inference, the model assigns a probability to each proposed alignment link. The final output matrix consists of active links whose probability exceeds a threshold p (optimized on a development set using a grid search). This parameter is used to control the density of the resulting alignment and therefore the balance between its precision and recall.

In this work, we used a maximum entropy (ME) classifier to estimate the probability of a link of $\mathbf{A}$:

$$p\left(y|\mathbf{x}\right) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k\left(y, \mathbf{x}\right)\right),$$

where $\mathbf{x}$ denotes the observation, $Z(\mathbf{x})$ is a normalization constant, $(f_k)_{k=1}^{K}$ defines a set of feature functions, and each f_k is associated with a weight, λ_k.

3.2 Features

In our discriminative models, we consider two kinds of features: word and alignment matrix features, some of them are illustrated in Figure 1.

Word features aim to describe the linguistic context of a given link, and depend on the sentence-pair in which it occurs, augmented by part-of-speech tags and related corpus statistics. They include (1) part-of-speech tags **(WPOS)** for a window of words, with variable size[1], surrounding the source and target words. POS tags for English are generated using the Stanford Tagger[2], while a POS tagger provided by ArabicSVMTools is used for Arabic; (2) surface lexical form **(WLex)** which is active if the source/target word is one of the N most frequent words[1]; (3) monotonicity **(WMon)** of the link $l_{i,j}$ which includes the difference between source and target absolute positions $|i - j|$ and their relative positions to the sentence length $\frac{i}{I}$, $\frac{j}{J}$ and $|\frac{i}{I} - \frac{j}{J}|$.

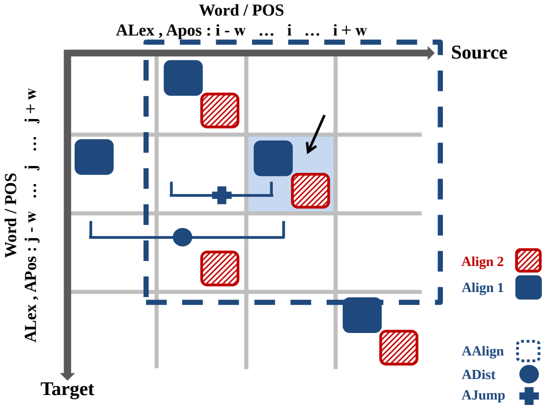
Word / POS
ALex , Apos : i - w ... i ... i + w
Source
Word / POS
ALex , APos : j - w ... j ... j + w
Target
Align 2
Align 1
AAlign
ADist
AJump

Figure 1: Features extracted to label the link pointed to by the arrow.

Alignment matrix features characterize the set of input alignment matrices, in addition to their union matrix $\mathbf{A}_{\cup}$. They include (1) predictions **(AAlign)** of individual input alignment systems (and their union $A_{\cup}$) for the current link and its neighborhood (a window of size $w \times w$)[1]. These features include whether a particular link in this neighborhood exists according to each input alignment and the total number of input alignments supporting it. Neighbor features are used to inform the current link about its surrounding points, motivated by the fact that alignments are usually centered around the diagonal in adjacent points; (2) source/target word fertility **(AFert)** which represents the number of target (source) words aligned to the current source (target) word according to a given input alignment and/or the union alignment; (3) distance features **(ADist)** describing the minimum/maximum distance between

[1] This variable introduces a model parameter to be used to optimize AER on a development set.

[2] http://nlp.stanford.edu/software/tagger.shtml

the current link and the previous/following links of same line/column according to the union alignment matrix. (4) jump features **(AJump)** characterizing the absolute distance between the current word and closest aligned one, on both source and target side according to the union alignment matrix. Most of these features have been already proposed (Ayan and Dorr, 2006; Elming and Habash, 2007; Blunsom and Cohn, 2006), exceptions are ADist and AJump, which are novelties of this work.

In (Ayan and Dorr, 2006), each feature function is conditioned twice on the POS tags of the source word and the target word. We add another conditioning criterion on their conjunction. Thus, we learn a separate weight for each feature for each source,target and source/target POS tags, allowing the model to pay more or less attention to each feature depending on the related tags.

3.3 Stacked Generalization

A problem with the ME framework, is that structure is not taken into account and labels are assumed to be independent. While this keeps the model simple, interactions between individual predictions cannot be modeled, and global decisions cannot be made. In order to incorporate structure and dependencies into the ME model, without sacrificing efficient, model-optimal predictions, we use a *stacked generalization* method (Wolpert, 1992). Stacked generalization is an approximation approach to structured learning. It allows to indirectly model dependencies between predicted labels at a low computational cost. It has been successfully applied to NLP problems, like dependency parsing (Martins et al., 2008), named entity recognition (Krishnan and Manning, 2006) and sequential partitioning problems (Cohen and Carvalho, 2005).

In stacked learning, all labels are jointly predicted in two steps. (1) For each training example $(x_i, \tilde{y_i})$, the entire set of observations $\mathbf{x} = [x_1, \ldots, x_n]$ is considered to extract features, that are then fed to a *first-level* classifier. This classifier is used to assign a label y_i to each observation x_i without taking dependencies between labels into consideration; then (2) observations are augmented with predictions of the local classifier $\mathbf{y} = [y_1, \ldots, y_n]$ to generate an *extended representation* of the training corpus, on which, a *second-level* classifier is trained. This classifier is able to make global decisions, using features that characterize the dependency between labels, produced by the first-level classifier.

A K-fold selection process When building training data for the global classifier, a K-*fold* selection process is used to avoid getting trapped in a *label-bias* problem. The entire training dataset is divided into K blocks, and K first-level classifiers are trained, each on a different subset (of $K - 1$ blocks) of training data. Each of these classifiers is then used to label the held-out block. These predictions, along with the original data, constitute training examples for the second-level classifier. Stacking avoids explicit joint modeling of labels and is thus merely an approximation method of structured learning. Nevertheless, it allows any type of dependency to be taken into account without complicating the model. The runtime of the training algorithm is $O(KT_f + T_s)$ where T_f and T_s are the individual runtimes required for training a first- and a second-level classifier respectively.

Stacking for word alignment For the task of word alignment as presented in this paper, the use of stacking consists in augmenting input alignments by one additional matrix, which is the output of the first-level classifier. Over this matrix, features characterizing the interactions between links in the final output alignment can be computed. The same set of features used for the first-level classifier is also used for the second-level one. That is we label the data with a first pass aligner and then we train another model using its prediction as features. Features like *ADist* and *AJump* are more suitable to capture characteristics of symmetric alignment matrices like the union alignment and the output of the first-level classifier, and hence, are calculated exclusively for them.

4 Experiments

In this section, we present several experiments to compare different word alignment strategies. We start by stating the experimental setup and then report AER results, as well as translation performances.

4.1 Experimental Setup and Metrics

We experimented the various models with the Arabic-English language pair using data described

Data source		#Sent	#Ar tok	#En tok
IBMAC	*test*	663	16K	19K
	dev	3,486	71K	89K
	train	10K	215K	269K
MT'08	*test set*	1,360	43K	53K
MT'06	*dev set*	1,797	46K	55K
MT'09 constrained track		5M	165M	163M

Table 1: Experimental data: number of sentences and running words.

in Table 1. The IBM Arabic-English aligned corpus (IBMAC) (Ittycheriah et al., 2006) provides manual word alignments. It includes a training set that we split into disjoint train and dev sets, used respectively for training and tuning our discriminative models. We use the IBMAC test set (NIST MT Eval'03) to evaluate different alignments in terms of *Alignment Error Rate* (AER). For ME training we used a freely available toolkit[3]. The model parameters are estimated using L-BFGS (Byrd et al., 1994) to maximize the regularized log-likelihood on a training corpus. A Gaussian prior is used during optimization to prevent overfitting. GIZA++ (Och and Ney, 2003) is used to train our generative alignments, with the additional parallel data made available by NIST MT Eval'09 constrained training condition. We used Moses[4] with SRILM[5] with the same data in our translation experiments. A 4-gram back-of language model is estimated using all English available data. Minimum Error-Rate Training (Och, 2003) is carried on to tune the parameters of the translation system on the NIST MT'06 test set. Translations are evaluated on NIST MT'08 test set.

Arabic pre-processing scheme and remappings Arabic is a morphologically complex, highly-inflected language. This makes normalization necessary to reduce the sparsity of the data. We use MADA+TOKAN[6] for morphological analysis, disambiguation and tokenization for Arabic. Given previous experiments on the NIST MT'09 task, we use the *D2* tokenization scheme that showed to perform best under large resource conditions (Habash and Sadat, 2006). For example, the Arabic phrase "wsyktbhA!"[7] ("and he will write it!" in English) is tokenized according to the D2 scheme as follows: "w+ s+ yktbhA !".

Since the hand-aligned IBMAC corpus is not tokenized with this scheme, two issues arise. (1) For evaluation, the IBMAC manual alignments and the ones estimated on D2-tokenized data should be compatible. Hence all words need to be mapped back (remapped) to the original form before preprocessing. In the previous example, an aligner will link the tokens in "w+ s+ yktbhA !" to different words on the English side. In the remapping step, the union of these links is assigned to the original word "wsyktbhA!". (2) For training, it is the other way around. The IBMAC manual alignments are split to match the tokenized words. When tokenizing an Arabic word, aligned to some English word(s), all resulting tokens are assumed to have the same set of alignment links as the original word. For instance, suppose that the word "wsyktbhA!" is aligned to all English words in "and he will write it!" in the IBMAC corpus. After applying the D2 tokenization scheme, we link each of the resulting tokens to all the English words. Although this assumption results in noisy reference alignments, it is still the easiest way to obtain reference alignments for D2 tokenized training data.

Metrics Alignment models described in this paper are compared and evaluated using two families of metrics. In the first one, the alignment under evaluation is compared to a gold standard, while, in the second one, its impact on the quality of the final translation is directly assessed.

When comparing alignments to a gold standard, the most commonly used metric is the alignment error rate (AER) (Och and Ney, 2003). Usually gold alignments are marked with "sure" or "possible" labels, but since the IBMAC corpus we are using has only sure ones, the AER reduces to balanced $1 - F_\alpha$ measure with $\alpha = 0.5$:

$$F_\alpha = \frac{Pr\ Rc}{\alpha Rc + (1-\alpha)Pr}$$

where Pr denotes the precision and Rc the recall. We also use F_α with different values for α in the F_α

[3] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

[4] http://www.statmt.org/moses/

[5] http://www-speech.sri.com/projects/srilm/

[6] http://www1.ccls.columbia.edu/ cadim/MADA.html

[7] All Arabic transliterations are provided in the Buckwalter transliteration scheme

Model	Direction	Pr%	Rc%	AER%
IBM1	Ar → En	56.4	66.2	39.1
	En → Ar	41.3	64.8	49.6
	gdfa	70.2	71.0	29.4
HMM	Ar → En	66.8	78.4	27.9
	En → Ar	51.0	72.6	40.1
	gdfa	73.9	81.3	22.6
IBM3	Ar → En	68.5	80.4	26.0
	En → Ar	56.5	77.3	34.8
	gdfa	**75.2**	83.8	20.7
IBM4	Ar → En	71.0	83.3	23.3
	En → Ar	58.9	79.8	32.3
	gdfa	75.0	**86.3**	**19.8**

Table 2: AER, precision and recall results for GIZA++ alignments with gdfa symmetrization heuristic.

measure formula to vary the trade-off between precision and recall as desired: α less (greater) than 0.5 weights recall (precision) higher (Fraser and Marcu, 2007). The quality of the translation is evaluated with BLEU (Papineni et al., 2002).

4.2 Alignment Error Rate Results

In this section we present precision, recall and AER results calculated using the IBMAC MT'03 test set, for generative and discriminative alignments.

4.2.1 GIZA++ Generative Alignments

Table 2 summarizes our baseline results obtained with three classical generative alignment models, as estimated by GIZA++, in both translation directions, and symmetrized using the *grow-diag-final-and* heuristic. Each step from IBM1 to IBM4 through HMM and IBM3 expectedly results in a better performance. The HMM model achieves a big error reduction over IBM1, with limited added computational complexity. While IBM3 and IBM4 continue to improve the quality of the alignments over HMM, they are much more computationally expensive (learning them takes a few days instead of a few hours) with smaller relative error reduction. Ar → En alignments are always better than En → Ar, which is due to differences in morphology between Arabic and English. For all the models, the symmetrization heuristic is able to improve both precision and recall, therefore AER, over the combined alignments.

Input Al.[#]	st	Pr%	Rc%	AER%
IBM1 [2]	✗	90.4	71.1	20.4
	✓	90.9	72.9	19.6
HMM [2]	✗	90.5	80.7	14.7
	✓	91.0	81.0	14.3
IBM3 [2]	✗	91.1	81.4	14.0
	✓	91.0	81.9	13.8
IBM4 [2]	✗	91.9	83.1	12.7
	✓	92.4	83.0	12.6
IBM1+ HMM [4]	✗	91.0	81.7	13.9
	✓	92.9	81.5	13.2
ALL [8]	✗	92.3	84.0	12.1
	✓	92.1	84.4	**11.9**
IBM4 *gdfa*		75.0	86.3	**19.8**

Table 3: Precision, recall and AER results for different sets of input alignments and stacking. [#] is the number of input alignments, st. denotes stacking.

4.2.2 Input Alignments and Stacking

Table 3 shows precision, recall and AER results for different set of input alignments. The best alignment with maximum entropy approach, augmented with stacking, achieves a much better precision than the best generative alignment, with worst recall and yields a 11.9% AER (a relative error reduction of 39.9% over the best GIZA++ alignment).

Since, in our approach, the alignment links considered by the ME classifier are only those proposed by the union of the input GIZA++ alignments, the recall of the latter is an upper bound on recall for the ME alignments. This explains the decrease in recall for our best alignment over IBM4. The same reason lies behind improvements seen when combining 8 input alignments instead of only 2: the more input alignments, the bigger their union is. Hence, the explored search space is wider, and more correct alignment links are allowed to be fetched, which leads to a higher recall. Table 4 shows the oracle AER for different sets of input alignments. The combination of the four generative models (IBM1, HMM, IBM3, IBM4) yields further improvement with an AER of 12.1% (11.9% with stacking). Stacking systematically improves the performance and achieves a state-of-the-art AER of 11.9% on the IBMAC test set. We also note that the difference between the worst pre-

Input Align.	Union Rc%	Oracle AER%
IBM1 [2]	75.9	13.7
HMM [2]	85.2	8.0
IBM3 [2]	86.4	7.3
IBM4 [2]	88.7	6.0
IBM1+HMM [4]	87.3	6.8
ALL [8]	90.8	4.8

Table 4: Union's recall and the corresponding AER's oracle for different set of input alignments.

cision (90.4%) and the best precision (92.9%) for all ME alignments, is much smaller than the difference between the worst recall (71.1%) and the best recall (84.4%). This result suggests that while the ME approach easily achieves a good precision even when using noisy input alignments, it is more difficult to improve its recall because of the upper bound imposed by the recall of the union of these input alignments.

The discriminative model systematically outperforms IBM models and the symmetrization heuristic. First, when combining two IBM1 directional alignments, an AER of 20.4% is achieved (19.6% with stacking) which is a big improvement compared to an AER of 29.4%, the result of combining the same two alignments with the symmetrization heuristic (a relative error reduction of 33.3%, when stacking is used). This result is quite impressive since the best input alignment from IBM1 has an AER of 39.0%, which means that even when using noisy input alignments, the ME model is able to perform a good error correction. Further more, the ME model, using only IBM1 alignments, allows to obtain comparable performance with the symmetrization heuristic using IBM4 alignments. This result is interesting since IBM4 is much more computationally expensive than IBM1 and HMM. Moreover, we can use more accurate input alignments to increase the gain: combining HMM alignments yields to an AER relative reduction of 28%.

4.2.3 Analysis of the Training Set Size

The discriminative approach requires hand-aligned data that are expensive to obtain. Hence, we are interested in knowing how many aligned sentences we need to train a model that performs reasonably. Figure 2 depicts AER as a function of the size of the training set (number of sentences) when using IBM4 input alignments. Although the bigger the training set the better the model, only small improvements are achievable when using more than 2000 sentences. It is worth noting that with only 10 training sentences (392 training examples), we get an AER of 18.5% which is lower than the AER obtained with the gdfa symmetrization heuristic (19.8%).

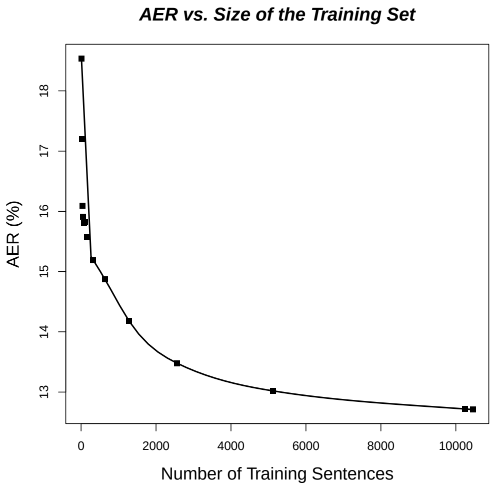

Figure 2: Relation between AER and the number of sentences in the training set when combining two IBM4 alignments).

4.2.4 Analysis of Features Contributions

To assess the impact of different kinds of features, a contrastive experiment is reported in table 5. The *basic* set of feature families includes the features previously used in (Ayan and Dorr, 2006), slightly modified. The *allF* set of features includes three additional families namely ADist, AJump and WLex. Each feature family is individually removed from the system containing all of them to tack its contribution. The increase in AER when removing a feature family indicates its importance. The basic family of features obtains an AER of 13.3% (a relative error reduction of 32.8% over IBM4), which confirms the results reported in (Ayan and Dorr, 2006).

Adding the new feature families further improves the AER: with these extra features the error rate falls down to 12.8%. While all features families have a positive contribution, their impact on AER varies. Both the AAlign feature family and data partition-

Features	**Pr%**	**Rc%**	**AER%**
Basic	90.0	83.7	13.3
AllF	91.6	83.2	**12.8**
− cond/WPOS	89.1	81.6	14.8
− AAlign	88.9	83.4	14.0
− AFert	91.7	82.2	13.3
− WLex	91.7	82.7	13.1
− preserved	91.9	82.9	12.9
− AJump	92.0	82.9	12.8
− ADist	92.1	82.9	12.8
− WMono	92.3	82.8	12.8
+ Union	91.9	83.1	12.7
AllF+Union+Stack	92.4	83.0	**12.6**
−AJump	92.0	82.9	12.7
−ADist	92.0	83.0	12.7

Table 5: Precision, recall and AER results for combining two IBM4 models with different features configurations. *basic:* features found in literature; *AllF* = basic + new features; and Union indicate using the union alignment in input.

ing (using WPOS feature family) have high positive contributions, since removing any one of them significantly worsen the AER. Other feature families, including AFert and WLex, have a less important impact, while, in our experiments, WMono does not produce any improvement.

Introducing two additional symmetrical alignments, namely "Union" and "Stack" seems to help features like ADist, AJump and AFert, whose contributions increase. In the "AllF" configuration, ADist and AJump are not of big help. However, when used in the "AllF+Union+Stack" configuration, they result in a small improvement. This could be explained by the fact that ADist and AJump are engineered to capture characteristics of a symmetrical alignment. Hence, enhanced performance can only be seen when using "Union" and "Stack" configurations, in which additional symmetrical alignments are used to extract features.

4.3 Machine Translation Results

In this section, we evaluate the interest of our method by measuring the impact of the various word alignment methods on translation quality. Results we present allow to gain insight into the relation between translation quality and different properties of the alignments, including different features configurations, different set of input alignments and different thresholds and stacking options.

Experiments are carried out using a large-scale Arabic to English phrase-based system developed for the NIST MT Eval'09 in the constrained training condition[8]. Although large-scale phrase-based systems tend to be robust to word alignment errors (Lopez and Resnik, 2006), improvements in translation quality are still attainable. We use the Moses toolkit to build a phrase-based system for different alignment methods using the data sets presented in table 1. All these systems are identical except for the word alignment component. BLEU scores are calculated using multi-reference BLEU without any post-processing of the output.

Table 6[9] shows BLEU, AER and $F_{0.3}$ scores obtained for GIZA++ gdfa alignments and various discriminative alignments. In terms of BLEU, the best performing discriminative alignment is the one combining all eight GIZA++ models (IBM1, HMM, IBM3 and IBM4) with a threshold $p = 0.4$: it achieves a BLEU score of 41.1% and a 0.7% absolute improvement over the best generative model.

Results of discriminative alignments suggest that they systematically improve translations over generative models. However, the impact of the features set and of stacking is unclear: using the features configuration that gives the best AER (denoted *best* 6) leads to slight improvements in BLEU (0.1%) over the *basic* feature set proposed by (Ayan and Dorr, 2006). Stacking does not seem to improve translation performances, even though it slightly improves the AER (Table 3). This suggests that in order to have significant improvements in BLEU, relatively big improvements in AER should be achieved. It is also worth noting that for a given input alignment set, BLEU results are not very sensitive to differences in threshold values around the best threshold. For example, when combining two IBM4 alignments, using the basic features configuration, threshold values 0.3, 0.4 and 0.6 produce comparable BLEU results of 40.9, 40.9 and 40.8, respectively. This suggests that picking-up an acceptable value for the threshold does not require an exhaus-

[8] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[9] In this table we show BLEU scores for thresholds that give either the best AER (usually $p = 0.7$) or the best $F_{0.3}$ (usually $p = 0.4$)

				AER	$F_{0.3}$	BLEU
GDFA Align.						
	IBM1			29.4	70.8	39.3
	HMM			22.6	78.9	40.0
	IBM4			19.8	82.6	40.4
Discriminative Align.						
model	*feat*	*p*	*st*			
IBM4 [2]	basic	0.6	✗	13.3	85.2	40.8
		0.3	✗	15.2	85.7	40.9
		0.4	✗	14.2	86.0	40.9
	best	0.7	✗	12.7	85.6	40.7
		0.4	✗	13.5	86.6	41.0
		0.7	✓	12.6	85.6	40.8
		0.4	✓	13.3	86.6	40.7
IBM1+ HMM [4]	best	0.4	✗	14.4	85.1	40.5
ALL [8]	best	0.4	✗	12.9	87.4	**41.1**
		0.7	✗	12.1	86.3	40.7
		0.4	✓	13.0	87.5	40.9

Table 6: Translation results in BLEU for different GIZA++ and discriminative word alignments. *best* corresponds to AllF+Union and *st.* to stacking. In bold is the best system's score.

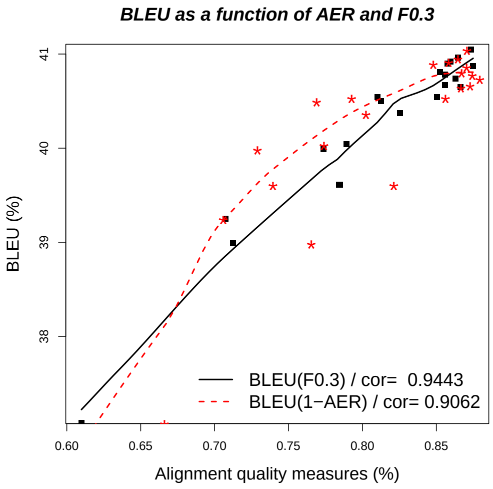

Figure 3: Correlation between BLEU and alignment quality measures.

tive search for the optimal solution.

As we have explained our approach can be used to learn the symmetrization heuristic from the data. Table 6 shows that the combination of two IBM4 models by a Maximum Entropy model results in an absolute gain of 0.6% BLEU point over a combination of these two alignments by a heuristic. An other interesting result is that a discriminative alignment considering the computationally not expensive IBM1 and HMM alignments as an input, does, at least, as well as the standard IBM4-gdfa[10].

In order to gain more insight into this aspect, we conducted a systematic experiment to evaluate the correlation between the BLEU score and various alignment metrics: we built 22 systems using different word alignment methods or parameters (11 systems were using a generative alignment method with different symmetrization heuristic; 11 systems were using a discriminative alignment method with different feature sets or input alignments or thresholds). We then calculated F_α for all values of $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ (with $1-$AER corresponding to $\alpha = 0.5$). The highest correlation between the alignment metric and the translation metric is obtained for $F_{0.3}$: their correlation, measured by the Pearson coefficient, is 94.34%. Note that the AER metric that is usually used to assess the performances of alignment methods also correlates well with BLEU but at a lower coefficient of 90.62%. The threshold p is used to control the density of the resulting alignments and therefore shifting the balance between precision and recall. Alignments with lower p are denser, and hence tend to have higher recall. Translation results show that alignments with an higher recall tend to perform better, suggesting that recall is preferable over precision and is the most influencing alignment quality component on the translation quality. Consequently, BLEU is expected to correlate better with measures favoring recall like $F_{0.3}$.

5 Conclusion

In this paper, we have presented a simple discriminative model for refining alignments produced by the IBM models. This model can be used, when supervised training data is available, as an alternative to the standard heuristic approach. By integrating several novel features and combining several input alignments, we were able to attain state-of-the art performance in terms of AER, and to help estimate better translation models: we showed that these improved alignments result in a increase of 0.7

[10] Runtime needed to train the maxent model is negligible and labeling is linear in the size of the corpus, which is in total faster than training IBM3 and IBM4 models (minutes vs. hours/days)

BLEU points of a large-scale Arabic-to-English system. We have also demonstrated that it is possible to achieve, by combining IBM1 and HMM alignments through discriminative training, models that outperform the conventional setting (IBM4 symmetrized alignments), at a much lower computational expense. Finally, we showed, in a series of systematic experiments, that there is a correlation between the quality of the word alignment measured by the $F_{0.3}$ metric and the BLEU score.

We plan to develop this work in several directions. Firstly, we intend to continue increasing the number of input alignments, especially alignments which can be computed efficiently. Since a relatively good precision can be achieved easily with our model, the upper bound on the recall is still a problem, currently dealt with by using expensive IBM4 models to push its limit. An alternative to improve the recall is to consider multiple IBM1 or HMMs alignments (either through the use of n-best alignments, or through the use of multiple initializations).

As word alignments are only meant to identify phrase-pairs, a second important direction of research will be to consider training with alternative global loss functions, so as to take into account the fact that some alignment links are more important than others. In the stacking framework, such extensions of the model can be performed efficiently.

Acknowledgments

This work was partly funded by the French agency for innovation OSEO under the Quaero Program.

References

N. F. Ayan and B. J. Dorr. 2006. A maximum entropy approach to combining word alignments. In *HLT-NAACL*, pages 96–103.

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *ICCL and ACL*, pages 65–72.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311.

R. H. Byrd, J. Nocedal, and R. B. Schnabel. 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Math. Program.*, 63(2):129–156.

C. Cherry and D. Lin. 2003. A probability model to improve word alignment. In *ACL*, pages 88–95.

W. W. Cohen and V. R. Carvalho. 2005. Stacked sequential learning. In *IJCAI*, pages 671–676.

J. DeNero and D. Klein. 2008. The complexity of phrase alignment problems. In *HLT*, pages 25–28.

J. Elming and N. Habash. 2007. Combination of statistical word alignments based on multiple preprocessing schemes. In *NAACL-HLT*, pages 25–28.

A. Fraser and D. Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303.

N. Habash and F. Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *NAACL-HLT*, pages 49–52.

A. Ittycheriah and S. Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *HLT '05*, pages 89–96.

A. Ittycheriah, Y. Al-Onaizan, and S. Roukos. 2006. The ibm arabic-english word alignment corpus. Technical Report RC24024, IBM.

V. Krishnan and C. D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *ICCL and ACL*, pages 1121–1128.

S. Lacoste-Julien, B. Taskar, D. Klein, and M. I. Jordan. 2006. Word alignment via quadratic assignment. In *NAACL-HLT*, pages 112–119.

Y. Liu, Q. Liu, and S. Lin. 2005. Log-linear models for word alignment. In *ACL*, pages 459–466.

A. Lopez and P. Resnik. 2006. Word-based alignment, phrase-based translation: What's the link? In *AMTA*, pages 90–99.

A. F. T. Martins, D. Das, N. A. Smith, and E. P. Xing. 2008. Stacking dependency parsers. In *EMNLP*, pages 157–166.

R. C. Moore. 2005. A discriminative framework for bilingual word alignment. In *HLT*, pages 81–88.

J. Niehues and S. Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proc. of the 3rd Workshop on SMT*, pages 18–25.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

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

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *HLT '05*, pages 73–80.

D. H. Wolpert. 1992. Original contribution: Stacked generalization. *Neural Netw.*, 5(2):241–259.