IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 239–255
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 08/02/11

Designing an Improved Discriminative Word Aligner

NADI TOMEH, ALEXANDRE ALLAUZEN,
THOMAS LAVERGNE, AND FRANÇOIS YVON

LIMSI/CNRS and Univ. Paris Sud, France

ABSTRACT

The quality of statistical machine translation systems depends on the quality of the word alignments, computed during the translation model training phase. IBM generative alignment models, despite their poor quality compared to a gold standard, perform well in practice. In this paper, we propose an improved word aligner based on a maximum entropy alignment combination model, which employ better feature engineering, ℓ^1 regularization, and an enhanced search space to improve the quality of both alignment and translation. For the Arabic-English language pair, we are able to reduce the Alignment Error Rate by 43.4%, and achieve ≈ 1 BLEU point enhancement over the IBM model 4 symmetrized alignments. These improvement are attainable at a lower computational cost, using only easy to estimate HMM and IBM model 1 features. An analysis of the obtained results shows that a good balance between several alignment characteristics should be maintained in order to deliver good translation quality.

1 INTRODUCTION

Word alignment aims to find word-level, non-compositional translational equivalences between two parallel sentences. In phrase-based, finding the optimal set of phrase pairs, which represents the translation model, is NP-hard [1]. To simplify the problem, constraints from a pre-computed word alignment are applied to restrict the search space, from which an extraction heuristic build the phrase-table. This two-steps approach makes the

problem of phrase pairs extraction boils down to the problem of word alignment, for which a wide range of models have been explored in the literature. Generative IBM models [2] are widely used in practice to construct two directional, one-to-many alignments in both translation directions, that are then symmetrized, on a sentence level, during a heuristical, post-processing step [3]. Training these models requires only a sentence-aligned bi-text, and is performed with the EM algorithm.

For linguistically different languages such as Arabic and English, a discriminative approach to word alignment is shown to be more effective even with a limited amount of labeled data. Indeed, the discriminative framework allows to model arbitrary and possibly inter-dependent aspects of the alignment process. In [4–6] word alignments is considered as a classification task, in which a binary classifier predicts for each possible assignment whether it should be included in the alignment or not. Discriminative models constitute a replacement to the local symmetrization heuristic that learns decisions in light of a global view of the data, by employing an arbitrary set of features including other models' predictions. Within this approach, the model extend the concept of symmetrization of two alignments into a combination of several ones.

However, in order to obtain a competitive performance, discriminative models face two issues that prevent their outspread application in practice. (1) The necessity to employ features based on predictions of IBM model 4 alignments, which are computationally demanding, and (2) technical issues (memory consumption and training/inference time) arising when incorporating a large number of features. In this paper, we extend the alignment combination and matrix modeling framework presented in [6] with an improved features engineering, combined with the use of ℓ^1 regularization for training the maximum entropy classifier. This kind of regularization allows the manipulation of a large number of features which will be selected during the training step. The resulting model is thus more compact and achieves similar results. Moreover, an improved search space is also investigated in order to increase the recall.

Using only easy to compute and exact models (IBM1 and HMM) as input, we are able to improve both alignment and translation quality, over the baselines. The improved search combined with stacking techniques yield the best performance. Three translation tasks of different sizes were considered to validate our findings. In order to shed some light on the nature of the relation between alignment and translation, we analyze BLEU scores in terms of alignment quality and other characteristics described in [7].

The rest of the paper is organized as follows. Related work is previewed in section 2 while the model with its components are presented in section 3. Finally, experiments and results are discussed in section 4.

2 Related Work

Several approaches for word alignment have been carried out recently and can be categorized in two major streams. In the first one word alignments is considered as a sequence labeling task, in which source words are tagged with target positions, using either generative models like HMM and IBM models [8] or discriminative ones like linear chain conditional random field (CRF) [9]. This representation of the problem results in directional, that require an additional symmetrization step to derive the many-to-many alignments. Symmetrization heuristics [8, 3], which starts with the intersection points of two directional alignments and progressively adds points from the union to cover unaligned words, performs well in practice. Even better performance can be achieved by tightly integrating symmetrization and model training [10].

The other stream aims to model the alignment matrix directly and produce many-to-many alignments, either employing generative models [11–14] or discriminative ones [15–18]. In the later, methods attempt 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 [19], an alignment between two sentences is evaluated with a global score using a non-decomposable discriminative scoring function. This model resorts to a beam search since no restriction on the form of the resulting alignments is considered and the search space is intractable. In [20], tractability is achieved by casting the word alignment task as a maximum weighted matching problem, 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 [21], by modeling alignment as a quadratic assignment problem which is NP-hard in general. Word alignment is also casted as a structured classification problem, in which a decision must be made to activate (or disactivate) each cell of the alignment matrix, admitting some dependency structure between decisions. In [18] a CRF with a complex structure is used to predict the alignment, with approximate inference and a complicated two-step training. In [4] an independence assumption helps simplifying the problem while sacrificing the ability to model interactions between decisions. A middle ground solution is proposed in [6], where exact learning

and inference is insured within the maximum entropy framework, while interactions are modeled by an additional stacked classification layer. Our work falls in this framework with improved feature engineering, ℓ^1 regularization for ME training, and enhanced search space.

3 WORD ALIGNMENT AS A STRUCTURED PREDICTION PROBLEM

Following [6] we represent the task of word alignment as a structured classification one, where we aim to predict the alignment matrix using a maximum entropy classifier. We discuss the impact of the search space and regularization on obtained alignments.

3.1 *Maximum Entropy Classifier for Word Alignment*

Let $\mathbf{f}_1^I = f_1, f_2, \ldots, f_I$ and $\mathbf{e}_1^J = e_1, e_2, \ldots, e_J$ be a source and a target sentence, respectively. The task of word alignment is to find a a mapping between subsets of **f** and subsets of **e** (a many-to-many correspondence between words of **f** and words of **e**).

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. Thus, word alignment can be seen as a structured binary classification task. We employ a maximum entropy (ME) classifier to estimate the probability of a link of **A**:

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

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

In order to incorporate structure and dependencies into the ME model, without sacrificing efficiency, we use a *stacked generalization* method [22] which has been successfully applied to NLP problems [23], including word alignment [6].

In stacked learning, all labels are jointly predicted in two-steps, using two classifiers. The *second-level* classifier is trained using data *extended* with the predictions of the *first-level* classifier, which characterizes the dependency between labels. The task of word alignment implements this concept using a first pass aligner and uses its prediction as features to train the second pass aligner.

A *K*-fold selection process is employed to build training data for the global classifier [6].

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. It also helps marginalizing the impact of the class-imbalance problem described below.

IMBALANCED DATASET Since the alignment matrix is typically sparse, with a majority of inactive links, the classification task we consider is imbalanced due to bias in training data acquisition. Whenever a class is over-represented its a priori probability to be chosen is higher than that of under-represented classes. Hence, attention should be payed to avoid learning a biased classifier with high tendency toward labeling all links as inactive. In previous work, the union of all input alignments is used to prune the search space and induce a more balanced dataset by reducing the number of links to be predicted to a subset of the alignment matrix: only points that have been proposed by at least one input alignment are labeled by the classifier, the others are assumed to be *inactive*. This reduction of the search space implies an upper bound on the recall by excluding a lot of plausible links, which become unreachable by the model. Since the ME classifier performs well precision-wise [6], the recall upper bound becomes a bottleneck.

ENHANCED ALIGNMENT SEARCH SPACE To push up the upper bound on recall, we exploit the observation that good candidate alignment points neighbor other good alignment points. Then the search space can be extended with additional links residing in a window of a fixed size, neighboring links proposed by input alignments. A down side for this heuristic is the increased number of negative examples, which shifts away training data from balance point. Possible solutions include random sub-sampling of the training data, and adjusting the selection threshold to neutralize the a priori probability assigned to over-represented *inactive* class.

TRAINING AND REGULARIZATION. The model is trained to optimize the regularized log-likelihood of the parameters. The most common regularization used in literature is the Gaussian prior (ℓ^2 penalty) which re-

duces overfitting and thus improve performance on most tasks. An alternative is to use a Laplacian prior (or ℓ^1 penalty). Such regularizer allows an efficient feature selection and yields sparse parameter vectors [24]. The regularization hyper-parameter aims to balance the pruning effect on the trained model.

To optimize the regularized log-likelihood, we use a second order quasi-Newton method. This kind of method requires a fully derivable function to optimize, which is not the case at zero for the ℓ^1 penalty. To overcome this problem, we use an adaptation of the classical L-BFGS, called OWL-QN, proposed in [25].

In addition to the ℓ^1 regularization term, a small ℓ^2 term is also added to overtake numerical problems that can results from using the second order method, leading to the so called *elastic-net* penalty [26]. Benefits of the *elastic-net* regularization are two-fold. It enables efficient features selection, without any loss in resulting model's quality. Moreover, the obtained models are interpretable, allowing for features contribution analysis. It should be noted that these advantages do not entail a change in the number of model's parameters, nor a higher computational complexity.

3.2 *Features*

The maximum entropy framework, along with ℓ^1-regularization allow for a wide marge of freedom when engineering features. The ones described in [6] are used in addition to features described here. Discretization of continuous features is performed in a pre-processing step, using an unsupervised equal frequency interval binning method. Fine-grained versions of all feature functions are added by conditioning on current POS tags. Learning a separate weight for each, allows the model to pay more or less attention to features depending on the related tags.

WORD FEATURES describe the source and target words associated with the given link. In addition to features described in [6] we include (1) *Lexical probability (WProb)* A separate feature for each discretized probability $p(f_i|e_j)$ and $p(e_j|f_i)$, produced by IBM model 1. (2) *Word frequency (WFreq)* The source and target word frequency (and their ratio) computed as the number of occurrences of the word form in training data. (3) *Lexical Prefix/Suffix (WPref,WSuff)* A separate feature for each prefix/suffix of a predefined length (and their concatenation), for $l_{i,j}$ source and target words. (4) *Word similarity (WSim)* These features reflects that proper nouns are likewise translated in different languages, e.g. "SdAm

Hsyn" [1] and "Saddam Hussein". A separate feature is defined per distinct value of the word similarity between $l_{i,j}$ source and target words. We employ the Levenshtein (edit) distance as a measure of similarity. (5) *Identity (WIdent)* is a binary feature which is active whenever f_i is equal to e_j (useful for untranslated numbers, symbols, names, and punctuation). (6) *Punctuation mismatch (WPunct)* indicates whenever a punctuation is aligned to a non-punctuation.

ALIGNMENT MATRIX FEATURES characterize the set of input alignment matrices, in addition to their union matrix $\mathbf{A}_{\cup}$. In addition to features described in [6] we include *multiple distortion (AMultd)* feature, which indicates if a link involves a duplicated word. Indeed, duplicated words could be misaligned due to a weak distortion model in comparison with lexical probabilities in IBM alignments [27]. E.g. several "fy" on the source side could be erroneously aligned to the same "in" on the target side regardless of the distortion. This feature is active for the link $l_{i,j}$ if f_i or e_j is duplicated, returning the distance to the diagonal.

4 EXPERIMENTAL RESULTS

All reported results have been obtained on the Arabic-English language pair, using data described in Table 1. The IBM Arabic-English manually aligned corpus (IBMAC) supplies our gold alignments. It includes the NIST MT Eval'03 as a test set, and a training set that we split into disjoint train and dev sets, used respectively for training and tuning our discriminative models. For ME training we used Wapiti [26][2], whereas the generative models are estimated using MGIZA++[3]. Default configurations are considered for the phrase based translation system Moses[4]. A 4-gram back-off language model, estimated with SRILM[5] is used in all our experiments. Minimum Error-Rate Training [28] is carried on to tune the parameters of the translation system. MADA+TOKAN[6] D2 tokenization scheme is used in a pre-processing step, to take Arabic rich

[1] All Arabic transliterations are provided in the Buckwalter transliteration scheme
[2] http://wapiti.limsi.fr/
[3] http://geek.kyloo.net/
[4] http://www.statmt.org/moses/
[5] http://www-speech.sri.com/projects/srilm/
[6] http://www1.ccls.columbia.edu/ cadim/MADA.html

Table 1. Experimental data: number of sentences and running words. G and D are acronyms for *generative* and *discriminative*, respectively.

Data source		#Sent	#Ar tok	#En tok	Usage
IBMAC	*test*	663	16K	19K	Evaluating all alignments
	dev	3,486	71K	89K	Tuning discriminative alignments
	train	10K	215K	269K	Training discriminative alignments
MT'08	*test*	1,360	43K	53K	Evaluating translations
MT'06	*test*	1,797	46K	55K	Tuning Moses parameters
MT'09	*constrained*	5M	165M	163M	Training MGIZA, SRILM and Moses

morphology into consideration. Inconsistency with tokenization of the IBMAC corpus is handled using the *splitting/remapping* technique described in [6]. We evaluate the quality of alignments compared to a gold standard using Alignment Error Rate (AER). The impact on translation quality is measured using multi-reference BLEU [29].

4.1 *Oracle study*

As explained in 3.1, limiting the search space to the union of input alignments, establishes an upper bound on the recall, preventing the model from reaching plausible links. In this oracle study, we quantify manual alignment reachability by several combination of input alignments, with different window sizes. Table 2 summarizes the percentage of alignment matrix covered by the union of input alignments, with its recall and AER according to the gold alignment. Oracle AER drops drastically when increasing the size of the window. Take, for instance, the case of IBM1 models: using a window of size 1 instead of 0 reduces the oracle by 10.8 points (from 13.7 to 2.9) at the cost of exploring larger area of alignment matrix (23.5% instead of 4.1%). It is worth noticing that the HMM model achieves similar oracle scores as IBM4, while its training and inference are fast and exact. Moreover, combining IBM1 and HMM results in comparable performances to the standard symmetrization heuristic (which has an oracle of 6.0 for the best IBM model), while exploring a slightly wider search space. Increasing the window size allows to largely outperform the heuristic with a much wider search space. This study suggests that most manual alignments are proposed by the input generative

Table 2. Search space coverage for different window sizes, and associated Oracle AER for different input alignments. W is the window size.

Input Alignments	Search Space %			Union Recall %			Oracle AER %		
	W=0	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*
IBM1	4.1	23.5	43.9	75.9	94.3	98.7	13.7	2.9	0.7
HMM	3.3	15.9	26.9	85.3	97.0	98.7	7.9	1.5	0.6
IBM4	**3.3**	15.7	26.6	88.7	98.4	99.4	**6.0**	0.8	0.3
IBM1 + HMM	**5.0**	**25.4**	45.4	87.3	98.3	99.6	**6.8**	**0.8**	0.2
ALL	5.5	26.8	**47.0**	90.8	99.2	99.7	4.8	0.4	**0.1**

models, and justifies their use to prune the search space (an AER of 0.1 is achievable by examining 47% of the matrix).

4.2 *Impact of system components on AER*

The framework we consider for word alignment includes several interacting components. We design the following experiment, in order to quantify the contribution of each of them. We train a baseline system and measure its performance in terms of AER. We then train several models, in each of which, only one component differs from the baseline, and we compare their performance.

All the new features added over the baseline described in [6] help improving both recall and precision. They can be divided in two classes according to their discrimination power: first come WPreff, WSuff, WProb and WFreq with about 0.5 AER reduction each, then AMultd, WIdent, WPunct and WSim with about to 0.2 AER reduction each. Including all the new features improves AER by 1.6 over the baseline. Different thresholds α are employed to test different balance point between precision and recall. Thresholds between 0.1 and 0.9 shift precision from 81.9 to 92.7 and recall from 72.5 to 64.7. The lowest AER for the baseline (22.4) is achieved at $\alpha = 0.5$. The more annotated training data is used by the discriminative model, the better. Increasing the training corpus size from 10 sentences to 10K, enhances the AER by 7 points.

Results for the other components can be founds in Table 3. We note that aggressive features pruning with high values of the ℓ^1 regularizer, results in improved precision and recall (hence AER). The biggest AER reduction of 1.2 points (at $\ell^1 = 3$) is attainable while discarding 97% of the features. We note also that increasing the size of the search space,

Table 3. Only the component in the first columns changes with respect to the baseline. The left part of the table shows the impact of (1) different values for ℓ^1 regularization. While the right part shows the impact of (2) different search space controlled by the size of the window, (3) different input alignments quality determined by the size of their trainind data, and (4) stacking. Baseline: IBM1 input alignments, window w=0, $\ell^1 = 0$, $\ell^2 = 0.01$, threshold $alpha = 0.5$, 2k word-aligned sentence (to train the discriminative models) and 5M parallel sentence to train IBM input models.

Align	Pr	Rc	AER	# ftr
Baseline	87.9	69.4	22.4	501238
(1) ℓ^1 *regularization*				
ℓ^1=0.1	86.7	69.4	22.9	92590
ℓ^1=0.5	88.0	69.9	22.1	50380
ℓ^1=1	88.8	70.2	21.6	35268
ℓ^1=2	89.3	70.3	21.3	19610
ℓ^1=3	89.4	70.4	21.2	13806
ℓ^1=4	89.3	70.3	21.3	10704
ℓ^1=5	89.4	70.0	21.5	8528
ℓ^1=6	89.1	70.2	21.5	7334

Align	Pr	Rc	AER
(2) Search space: window size			
W=0	87.9	69.4	22.4
W=1	78.0	82.0	20.1
W=2	77.2	81.6	20.6
(3) Input alignment quality			
30K	85.9	64.0	26.7
130K	87.2	66.1	24.8
1030K	87.3	68.3	23.4
(4) Stacking			
Stack	89.1	70.2	21.4

using larger windows around the current link (e.g. w = 1), reduces the AER by 2.3. When exploring a wider search space, the model is able to retrieve more links, improving the recall by 12.6 points. But it makes more mistake, since it has to make more decisions, and hence degrade precision by 9.9 points. It should be mentioned subsampling methods to treat the imbalanced data problem related to wide search spaces did not help.

To evaluate the model's sensitivity to the quality of input alignments, we exploit the fact that training MGIZA alignments with less data results in alignments with degraded quality: we train IBM model 1 with MGIZA using corpora of different sizes (30K, 130K, 1030K). Each on of these alignments is then used as an input to build a discriminative system. The resulting systems are then compared to the baseline, which is build using IBM model 1 alignment trained on the entire 5M parallel corpus.

The baseline's AER drops from 22.4 to 26.7 for the worst input alignment (IBM1 trained on 30K). Stacking helps correcting errors in the baseline and improve its AER by 1 point by enhancing both recall and precision.

Table 4. Sample of selected features with high weights

Feature	Weight
$l_{i,j} = active \wedge WPref(f_i) = Al\$ \wedge WPref(e_j) = el-$	1.7313
$l_{i,j} = active \wedge WPref(f_i) = Anh \wedge WPref(e_j) = tha$	1.6652
$l_{i,j} = active \wedge POS(f_i) = CC \wedge POS(e_j) = CC$	1.4559
$l_{i,j} = inactive \wedge WPunc(f_i, e_j)$	1.2070
$l_{i,j} = active \wedge MGIZA\_HMM(f_i, e_j) = active$	0.7639

4.3 *Model and features selection*

As described in section 3.1, the use of ℓ^1 regularization yields a sparse model where the most useful features have been selected during the training step. Some of these features are shown in Table 4: the first binary feature indicates if the Arabic word starts with the prefix *Al* while the English word begins with the *el* prefix. This feature indeed embeds a rule of thumb to translate Arabic proper noun, and is sufficient to ensure correct alignments for all the related occurrences in the test set. The second feature encodes the punctuation mismatch and prevents to align punctuations with regular words. With this feature, the model prefers to leave a punctuation not aligned, rather than aligned with a regular word. This decision is generally the best if a punctuation cannot be aligned with another punctuation.

Even if most of the selected features are related to the input generative models HMM and IBM1 (40% of the features), a more global study shows that all classes are represented in the final model and so are useful for alignment. Moreover, it is worth noticing that 90% of the selected features are those conditioned on current POS tags.

4.4 *Alignments characteristics and BLEU*

In these experiments, we aim to assess the quality of the new alignments measured by AER, and their impact on the translation quality measured by BLEU. We use two different baselines: generative IBM alignments and discriminative ME alignments described in [6]. In Phrase Based Statistical Machine Translation (PBSMT), the basic translation unit is a phrase pair and its associated scores. Phrase pairs are extracted from a parallel corpus to build the phrase table which contains ideally all sub-sentential translational equivalences. The quality of these phrase

pairs is determined by the number of correct translational correspondences they can capture. In practice, word-level correspondences are precomputed, then fed to an extraction heuristic that generalizes these correspondences to the phrase level. In this scenario, two sources of errors may affect phrase pairs consistency. On the one hand, word alignments are error prone, and they fail sometimes to detect word-level translations which carry on to the extracted phrase pairs. However, the extraction heuristic achieves generalization by combining aligned words into phrases, and growing over unaligned ones around them. This can be helpful to treat cases where no word-level alignment exists, such as in the translation of propositions, idiomatic expressions and compound words. However, since this heuristic operates locally on a sentence level and makes heursitic decisions, it can easily extract noisy phrases, especially when given a wide margin of freedom by leaving many words unaligned.

Since word alignments are the only constraints on the extraction heuristic, they become the only way to control both sources of errors mentioned earlier: by settling on a good balance between the alignment quality and the number of unaligned words. Therefore, the tradeoff between precision and recall for word alignments has a great impact on the quality of the extracted phrases. For instance, let us consider the case of a perfect precision (all links are correct), but with a low recall (not all word-level correspondences are detected). Then the alignment matrix is sparser than it should be, and the proportion of unaligned words results in many phrase pairs, with moderate scores (since they allow for multiple translations which over-flatten the probability distribution). Human-perceived quality of resulting phrases also degrades [7]. In the other case, with a high recall and low precision, the alignment matrix is denser than it should be, and generalization fails, with fewer and over-deterministic phrase pairs. Thus, the quality of a phrase table depends on the interaction between the quality of word alignments (precision and recall) and the sparsity of the alignment matrix: the number of unaligned source or target words, and the resulting gaps.

Our results are interpretable in light of this discussion. IBM1 is an example of alignments were both sources of errors are apparently affecting its performance. Compared to a manual alignment: IBM1 (1) produces poor alignment quality with low precision and recall, which causes the extraction of erroneous phrase pairs, and (2) leaves more words unaligned, which adds to the noise in extracted phrases.

Subsequent generative models are more efficient: HMM and IBM4 improves both precision and recall, while aligning more words. These

Table 5. Characteristics of alignments in terms of their quality compared to gold standard, number of links and unaligned source/target words. Number of extracted phrases is included with average number of gap per source/target word, and percentage of gapless phrases. These statistics are calculated using the IBMAC test set 1. Finally the quality of alignments in terms of their impact on BLEU for three different MT tasks. Th is the threshold.

Alignment Characteristics						**Phrase-pairs Characteristics**					**BLEU**		
Recall	***Precision***	***AER***	***#links***	***#UnalignSrc***	***#UnalignTgt***	***#Phrases***	***avgSrcGap***	***%GaplessSrc***	***avgTgtGap***	***%GaplessTgt***	***30K***	***130K***	***1030K***
Manual alignments													
100	100	0.0	16171	2655	3457	86642	0.68	56.5	0.83	47.6	-	-	-
Generative baselines (gdfa): *IBM1, HMM, IBM4*													
70.2	71.0	29.4	16394	3032	4752	72369	0.98	47.6	1.28	36.9	36.0	39.2	40.6
73.7	81.4	22.6	17985	1967	3524	74782	0.63	62.8	0.96	46.6	37.5	40.5	41.5
75.1	86.1	**19.8**	18715	1422	2460	60029	0.34	75.4	0.57	61.0	38.0	41.1	**42.0**
Discriminative baseline [6]: IBM1-HMM ($th = 0.6$), +Stacking ($th = 0.5$)													
90.7	82.0	13.9	14733	3435	4851	119303	1.04	44.2	1.29	33.9	38.0	41.3	42.3
92.7	81.5	**13.2**	14953	3371	4542	122412	0.99	44.8	1.13	34.4	38.2	41.4	**42.3**
New system: *IBM1-HMM ($th = 0.5$), +Window ($th = 0.8$), +Stacking ($th = 0.7$)*													
91.4	82.7	13.2	15215	3197	4552	106490	0.93	47.5	1.18	36.8	38.2	41.4	42.8
89.7	86.6	11.8	17143	3436	4019	107063	1.05	43.7	1.14	39.2	37.4	40.5	41.8
93.1	86.5	**11.2**	16054	3008	4173	108825	0.91	46.6	1.15	38.9	38.5	41.7	**42.9**

enhancements lead to the extraction of less noisy phrase pairs, which perform better. IBM4 for example, improves BLEU by 2 points over IBM1 for the smallest task, and 1.4 points for the biggest one.

Discriminative baseline alignments have different profile than IBM models. They are more efficient in retrieving more correct links, as well as greatly improving recall (from 75% for IBM4 to 92.7% for the stacked baseline). Thus, the quality of extracted phrase pairs should be improved significantly since they are based on better word-level correspondences, and the first source of errors is limited. But on the other hand, these alignments tend to be sparser than gold standards (9% less links) and IBM4 model (21% less links), which causes more extraction errors, degrading back the quality of phrase pairs. Otherwise stated, the improvement in

phrase table quality due to AER improvement, is almost cancelled out by increasing the percentage of gaps. Which explains why discriminative baselines achieve small improvements over IBM models.

The new system has a profile similar, in general, to the discriminative baseline: improved AER, and sparser alignments. The first line in the new system part of the table 5 describes the first new system, which uses the better feature engineering and ℓ^1 regularization (without the enhanced search space). It achieves comparable alignment quality (precision, recall) to the discriminative baseline, but it is able to align more words, and decrease the percentage of gaps. This results in higher BLEU scores on all the three tasks.

Adding the enhanced search space (window of size 1) to the previous system, allows for a significant increase in recall (from 82.7% to 86.6%) with slightly degraded precision, which improves the AER. These alignments change the balance between unaligned source and target words, with respect to the previous system: more *source* words, and *less* target words are aligned, in a comparable sized phrase table. This configuration is harmful and results in about 1 BLEU point loss on all tasks. An interesting result comes in the next line, when adding a stacking layer to the system with the enhanced search space. Stacking fixes the problem with precision, without harming recall, improves the over all quality of the alignment, and reduces the number of unaligned source words, shifting the balance back. This system achieves the lowest alignment error rate of 11.2%, and the best BLEU score on all three tasks, with significant improvements over the generative and discriminative baselines (for the biggest task).

5 CONCLUSIONS

In this paper we introduced an improved discriminative alignment system, that is based on a maximum entropy framework for combining word alignments. Better feature engineering, combined with ℓ^1 regularization and an enhanced search space allow the model to improve both the quality of alignment and translation. The introduced model achieves an overall reduction of 43% of the alignment error rate over the standard IBM model 4 symmetrized alignment, resulting in 0.9 point enhancement in BLEU. While adding a stacked classification layer may not be very helpful to the discriminative baseline introduced in [6], it proves to be necessary to allow the new system to benefit from the enhanced search space and achieve improvements in translation quality.

We analyzed the BLEU results in light of several alignment characteristics and noticed that finding a better balance between the alignment quality measured by its precision and recall, its sparsity, and its number of unaligned words and extracted phrases is necessary to deliver better translation models.

ACKNOWLEDGMENTS The work was partly realized as part of the Quaero Program, funded by OSEO, the French agency for innovation.

REFERENCES

1. DeNero, J., Klein, D.: The complexity of phrase alignment problems. In: HLT. (2008) 25–28
2. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist. **19** (1993) 263–311
3. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: NAACL. (2003) 48–54
4. Ayan, N.F., Dorr, B.J.: A maximum entropy approach to combining word alignments. In: HLT-NAACL. (2006) 96–103
5. Elming, J., Habash, N.: Combination of statistical word alignments based on multiple preprocessing schemes. In: NAACL-HLT. (2007) 25–28
6. Tomeh, N., Allauzen, A., Yvon, F., Wisniewski, G.: Refining word alignment with discriminative training. In: AMTA. (2010)
7. Guzman, F., Gao, Q., Vogel, S.: Reassessment of the role of phrase extraction. In: 12th MT Summit. (2009)
8. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29** (2003) 19–51
9. Blunsom, P., Cohn, T.: Discriminative word alignment with conditional random fields. In: ICCL and ACL. (2006) 65–72
10. Matusov, E., Zens, R., Ney, H.: Symmetric word alignments for statistical machine translation. In: COLING. (2004)
11. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics **23** (1997) 377–403
12. Tiedemann, J.: Word to word alignment strategies. In: COLING. (2004) 212–218
13. Lin, D., Cherry, C.: Word alignment with cohesion constraint. In: HLT-NAACL. (2003)
14. Zhao, B., Vogel, S.: Word alignment based on bilingual bracketing. In: HLT-NAACL 2003. (2003) 15–18
15. Cherry, C., Lin, D.: A probability model to improve word alignment. In: ACL. (2003) 88–95

16. Ittycheriah, A., Roukos, S.: A maximum entropy word aligner for Arabic-English machine translation. In: HLT '05. (2005) 89–96
17. Liu, Y., Liu, Q., Lin, S.: Log-linear models for word alignment. In: ACL. (2005) 459–466
18. Niehues, J., Vogel, S.: Discriminative word alignment via alignment matrix modeling. In: Proc. of the 3rd Workshop on SMT. (2008) 18–25
19. Moore, R.C.: A discriminative framework for bilingual word alignment. In: HLT. (2005) 81–88
20. Taskar, B., Lacoste-Julien, S., Klein, D.: A discriminative matching approach to word alignment. In: HLT '05. (2005) 73–80
21. Lacoste-Julien, S., Taskar, B., Klein, D., Jordan, M.I.: Word alignment via quadratic assignment. In: NAACL-HLT. (2006) 112–119
22. Wolpert, D.H.: Original contribution: Stacked generalization. Neural Netw. **5** (1992) 241–259
23. Cohen, W.W., Carvalho, V.R.: Stacked sequential learning. In: IJCAI. (2005) 671–676
24. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B **58** (1996) 267–288
25. Andrew, G., Gao, J.: Scalable training of L1-regularized log-linear models. In: ICML. (2007) 33–40
26. Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: ACL. (2010)
27. Riesa, J., Marcu, D.: Hierarchical search for word alignment. In: ACL. (2010)
28. Och, F.J.: Minimum error rate training in statistical machine translation. In: ACL. (2003) 160–167
29. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL. (2002) 311–318

NADI TOMEH
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <NADI.TOMEH@LIMSI.FR>

ALEXANDRE ALLAUZEN
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <ALEXANDRE.ALLAUZEN@LIMSI.FR>

THOMAS LAVERGNE
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <THOMAS.LAVERGNE@LIMSI.FR>

FRANÇOIS YVON
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <FRANCOIS.YVON@LIMSI.FR>