A Correction Model for Word Alignments

J. Scott McCarley, Abraham Ittycheriah, Salim Roukos, Bing Xiang, Jian-ming Xu
IBM T.J. Watson Research Center
1101 Kitchawan Road, Rt. 134
Yorktown Heights, NY 10598
{jsmc,abei,roukos,bxiang,jianxu}@us.ibm.com

Abstract

Models of word alignment built as sequences of links have limited expressive power, but are easy to decode. Word aligners that model the alignment matrix can express arbitrary alignments, but are difficult to decode. We propose an alignment matrix model as a correction algorithm to an underlying sequence-based aligner. Then a greedy decoding algorithm enables the full expressive power of the alignment matrix formulation. Improved alignment performance is shown for all nine language pairs tested. The improved alignments also improved translation quality from Chinese to English and English to Italian.

1 Introduction

Word-level alignments of parallel text are crucial for enabling machine learning algorithms to fully utilize parallel corpora as training data. Word alignments appear as hidden variables in IBM Models 1-5 (Brown et al., 1993) in order to bridge a gap between the sentence-level granularity that is explicit in the training data, and the implicit word-level correspondence that is needed to statistically model lexical ambiguity and word order rearrangements that are inherent in the translation process. Other notable applications of word alignments include cross-language projection of linguistic analyzers (such as POS taggers and named entity detectors,) a subject which continues to be of interest. (Yarowsky et al., 2001), (Benajiba and Zitouni, 2010)

The structure of the alignment model is tightly linked to the task of finding the optimal alignment. Many alignment models are factorized in order to use dynamic programming and beam search for efficient marginalization and search. Such a factorization encourages - but does not require - a sequential (often left-to-right) decoding order. If left-to-right decoding is adopted (and exact dynamic programming is intractable) important right context may exist beyond the search window. For example, the linkage of an English determiner may be considered before the linkage of a distant head noun.

An alignment model that jointly models all of the links in the entire sentence does not motivate a particular decoding order. It simply assigns comparable scores to the alignment of the entire sentence, and may be used to rescore the top-N hypotheses of another aligner, or to decide whether heuristic perturbations to the output of an existing aligner constitute an improvement. Both the training and decoding of full-sentence models have presented difficulties in the past, and approximations are necessary.

In this paper, we will show that by using an existing alignment as a starting point, we can make a significant improvement to the alignment by proposing a series of heuristic perturbations. In effect, we train a model to fix the errors of the existing aligner. From any initial alignment configuration, these perturbations define a multitude of paths to the reference (gold) alignment. Our model learns alignment moves that modify an initial alignment into the reference alignment. Furthermore, the resulting model assigns a score to the alignment and thus could be used in numerous rescoring algorithms, such as top-N rescorers.

In particular, we use the maximum entropy frame-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 889–898,
Edinburgh, Scotland, UK, July 27–31, 2011.

work to choose alignment moves. The model is symmetric: source and target languages are interchangeable. The alignment moves are sufficiently rich to reach arbitrary phrase to phrase alignments. Since most of the features in the model are not language-specific, we are able to test the correction model easily on nine language pairs; our corrections improved the alignment quality compared to the input alignments in all nine. We also tested the impact on translation and found a 0.48 BLEU improvement on Chinese to English and a 1.26 BLEU improvement on English to Italian translation.

2 Alignment sequence models

Sequence models are the traditional workhorse for word alignment, appearing, for instance, in IBM Models 1-5. This type of alignment model is not symmetric; interchanging source and target languages results in a different aligner. This parameterization does not allow a target word to be linked to more than one source word, so some phrasal alignments are simply not considered. Often the choice of directionality is motivated by this restriction, and the choice of tokenization style may be designed (Lee, 2004) to reduce this problem. Nevertheless, aligners that use this parameterization internally often incorporate various heuristics in order to augment their output with the disallowed alignments - for example, swapping source and target languages to obtain a second alignment (Koehn et al., 2007) with different limitations. Training both directions jointly (Liang et al., 2006) and using posterior probabilities during alignment prediction even allows the model to see limited right context. Another alignment combination strategy (Deng and Zhou, 2009) directly optimizes the size of the phrase table of a target MT system.

Generative models (such as Models 1-5, and the HMM model (Vogel et al., 1996)) motivate a narrative where alignments are selected left-to-right and target words are then generated conditioned upon the alignment and the source words. Generative models are typically trained unsupervised, from parallel corpora without manually annotated word-level alignments.

Discriminative models of alignment incorporate source and target words, as well as more linguistically motivated features into the prediction of alignment. These models are trained from annotated word alignments. Examples include the maximum entropy model of (Ittycheriah and Roukos, 2005) or the conditional random field jointly normalized over the entire sequence of alignments of (Blunsom and Cohn, 2006).

3 Joint Models

An alternate parameterization of alignment is the alignment matrix (Niehues and Vogel, 2008). For a source sentence F consisting of words $f_1...f_m$, and a target sentence $E = e_1...e_l$, the alignment matrix $A = \{\sigma_{ij}\}$ is an $l \times m$ matrix of binary variables. If $\sigma_{ij} = 1$, then e_i is said to be linked to f_j. If e_i is unlinked then $\sigma_{ij} = 0$ for all j. There is no constraint limiting the number of source tokens to which a target word is linked either; thus the binary matrix allows some alignments that cannot be modeled by the sequence parameterization. All 2^{lm} binary matrices are potentially allowed in alignment matrix models. For typical l and m, $2^{lm} \gg (m+1)^l$, the number of alignments described by a comparable sequence model. This parameterization is symmetric - if source and target are interchanged, then the alignment matrix is transposed.

A straightforward approach to the alignment matrix is to build a log linear model (Liu et al., 2005) for the probability of the alignment A. (We continue to refer to “source” and “target” words only for consistency of notation - alignment models such as this are indifferent to the actual direction of translation.) The log linear model for the alignment (Liu et al., 2005) is

$$p(A|E,F) = \frac{\exp\left(\sum_i \lambda_i \phi_i(A,E,F)\right)}{Z(E,F)} \qquad (1)$$

where the partition function (normalization) is given by

$$Z(E,F) = \sum_A \exp\left(\sum_i \lambda_i \phi_i(A,E,F)\right). \qquad (2)$$

Here the $\phi_i(A,E,F)$ are feature functions. The model is parameterized by a set of weights λ_i, one for each feature function. Feature functions are often binary, but are not required to be. Feature functions may depend upon any number of components σ_{ij} of the alignment matrix A.

The sum over all alignments of a sentence pair (2^{lm} terms) in the partition function is computationally impractical except for very short sentences, and is rarely amenable to dynamic programming. Thus the partition function is replaced by an approximation. For example, the sum over all alignments may be restricted to a sum over the n-best list from other aligners (Liu et al., 2005). This approximation was found to be inconsistent for small n unless the merged results of several aligners were used. Alternately, loopy belief propagation techniques were used in (Niehues and Vogel, 2008). Loopy belief propagation is not guaranteed to converge, and feature design is influenced by consideration of the loops created by the features. Outside of the maximum entropy framework, similar models have been trained using maximum weighted bipartite graph matching (Taskar et al., 2005), averaged perceptron (Moore, 2005), (Moore et al., 2006), and transformation-based learning (Ayan et al., 2005).

4 Alignment Correction Model

In this section we describe a novel approach to word alignment, in which we train a log linear (maximum entropy) model of alignment by viewing it as correction model that fixes the errors of an existing aligner. We assume a priori that the aligner will start from an existing alignment of reasonable quality, and will attempt to apply a series of small changes to that alignment in order to correct it. The aligner naturally consists of a *move generator* and a *move selector*.

The move generator perturbs an existing alignment A in order to create a set of candidate alignments $\mathcal{M}_t(A)$, all of which are nearby to A in the space of alignments. We index the set of moves by the decoding step t to indicate that we generate entirely different (even non-overlapping) sets of moves at different steps t of the alignment prediction. Typically the moves affect linkages local to a particular word, e.g. the t'th source word.

The move selector then chooses one of the alignments $A_{t+1} \in \mathcal{M}_t(A_t)$, and proceeds iteratively: $A_{t+2} \in \mathcal{M}_{t+1}(A_{t+1})$, etc. until suitable termination criteria are reached. Pseudocode is depicted in Fig. (1.) In practice, one move for each source and target word is sufficient.

```
**Input:** sentence pair $E_1$ .. $E_l$, $F_1$ .. $F_m$
**Input:** alignment $A$
**Output:** improved alignment $A_{final}$
  **for** $t = 1 \rightarrow l$ **do**
    generate moves: $\mathcal{M}_t(A_t)$
    select move:
    $A_{t+1} \leftarrow \text{argmax}_{A \in \mathcal{M}_t(A_t)} p(A|A_t, E, F)$
  $A_{final} \leftarrow A_{l+1}$
  {repeat for source words}
```

Figure 1: pseudocode for alignment correction

4.1 Move generation

Many different types of alignment perturbations are possible. Here we restrict ourselves to a very simple move generator that changes the linkage of exactly one source word at a time, or exactly one target word at a time. Many of our corrections are similar to those of (Setiawan et al., 2010), although our motivation is perhaps closer to (Brown et al., 1993), who used similar perturbations to approximate intractable sums that arise when estimating the parameters of the generative models Models 3-5, and approach refined in (Och and Ney, 2003). We note that our corrections are designed to improve even a high-quality starting alignment; in contrast the model of (Fossum et al., 2008) considers deletion of links from an initial alignment (union of aligners) that is likely to overproduce links.

From the point of view of the alignment matrix, we consider changes to one row or one column (generically, one slice) of the alignment matrix. At each step t, the move set $\mathcal{M}_t(A_t)$ is formed by choosing a slice of the current alignment matrix A_t, and generating all possible alignments from a few families of moves. Then the move generator picks another slice and repeats. The $m + l$ slices are cycled in a fixed order: the first m slices correspond to source words (ordered according to a heuristic top-down traversal of the dependency parse tree if available), and the remaining l slices correspond to target words, similarly parse-ordered. For each slice we consider the following families of moves, illustrated by rows.

- add link to row i - for one j such that $\sigma_{ij} = 0$, make $\sigma_{ij} = 1$ (shown here for row $i = 1$.)

	α	β	γ
a	○	○	○
b	○	●	○
c	○	○	○

$\Longrightarrow$

	α	β	γ
a	●	○	○
b	○	●	○
c	○	○	○

- remove one or more links from row i - for some j such that $\sigma_{ij} = 1$, make $\sigma_{ij} = 0$ (shown here for $i = 3$.)

	α	β	γ
a	●	○	○
b	○	●	○
c	○	○	●

$\Longrightarrow$

	α	β	γ
a	●	○	○
b	○	●	○
c	○	○	○

- move a link in row i - for one j and one j' such that $\sigma_{ij} = 1$ and $\sigma_{ij'} = 0$, make $\sigma_{ij} = 0$ and $\sigma_{ij'} = 1$ (shown here for $i = 1$.)

	α	β	γ
a	○	●	○
b	○	●	○
c	○	○	○

$\Longrightarrow$

	α	β	γ
a	●	○	○
b	○	●	○
c	○	○	○

- leave row i unchanged

Similar families of moves apply to column slices (source words.) In practice, perturbations are restricted by a window (typically ±5 from existing links.) If the given source word is unlinked, we consider adding a link to each target word in a window (±5 from nearby links.) The window size restrictions mean that some reference alignments are not reachable from the starting point. However, this is unlikely to limit performance - an oracle aligner achieves $97.6\% F$-measure on the Arabic-English training set.

4.2 Move selection

A log linear model for the selection of the candidate alignment at $t+1$ from the set of alignments $\mathcal{M}_t(A_t)$ generated by the move generator at step t takes the form:

$$p(A_{t+1}|E, F, \mathcal{M}_t(A_t)) = \frac{e^{\sum_i \lambda_i \phi_i(A_{t+1}, E, F)}}{Z(E, F, \mathcal{M}_t(A_t))} \quad (3)$$

where the partition function is now given by

$$Z(E, F, \mathcal{M}) = \sum_{A \in \mathcal{M}} e^{\sum_i \lambda_i \phi_i(A, E, F)} \quad (4)$$

and $A_{t+1} \in \mathcal{M}_t(A_t)$ is required for correct normalization. This equation is notationally very similar to equation (1), except that the predictions of the model are restricted to a small set of nearby alignments. For the move generator considered in this paper, the summation in Eq.(4) is similarly restricted, and hence training the model is tractable. The set of candidate alignments $\mathcal{M}_t(A_t)$ typically does not contain the reference (gold) alignment; we model the best alignment among a finite set of alternatives, rather than the correct alignment from among all possible alignments. This is a key difference between our model and (Liu et al., 2005).

Note that if we extended our definition of perturbation to the limiting case that the alignment set included all possible alignments then we would clearly recover the standard log linear model of alignment.

4.3 Training

Since the model is designed to predict perturbation to an alignment, it is trained from a collection of errorful alignments and corresponding reference sequences of aligner moves that reach the reference (gold) alignment. We construct a training set from a collection of sentence pairs and reference alignments for training $(A^{*n}, E^n, F^n)_{n=1}^N$, as well as collections of corresponding "first pass" alignments A_1^n produced by another aligner. For each n, we form a number of candidate alignment sets $\mathcal{M}_t(A_t^n)$, one for each source and target word. For training purposes, the true alignment from the set is taken to be the one identical with A^{*n} in the slice targeted by the move generator at the current step. (A small number of move sets do not have an exact match and are discarded.) Then we form an objective function from the log likelihood of reference alignment, smoothed with a gaussian prior

$$\mathcal{L} = \sum_n \mathcal{L}_n + \sum_i (\lambda_i / \gamma)^2 \quad (5)$$

where the likelihood of each training sample is

$$\begin{aligned}\mathcal{L}_n &= \sum_\alpha \log p_1(A_n^0|E,F_n;\mathcal{M}(f_\alpha,A_n^0,E,F_n)) \\ &+ \sum_\beta \log p_1(A_n^0|E,F_n;\mathcal{M}(e_\beta,A_n^0,E,F_n)) \quad (6)\end{aligned}$$

The likelihood has a term for each sentence pair and for each decoder step. The model is trained by gradient ascent using the l-BFGS method (Liu and Nocedal, 1989), which has been successfully used for training log linear models (Blunsom and Cohn, 2006) in many natural language tasks, including alignment.

5 Features

A wide variety of features were used in the model. We group the features in three broad categories: link-based, geometrical, and parse-based.

Link-based features are those which decompose into a (linear) sum of alignment matrix elements σ_{ij}. An example link-based feature is one that fires if a source language noun is linked to a target language determiner. Note that this feature may fire more than once in a given sentence pair: as with most features in our model, it is an integer-valued feature that counts the number of times a structure appears in a sentence pair. These features do not capture any correlation between different σ_{ij}. Among the link-based features are those based on Model 1 translation matrix parameters $\tau(e_i|f_j)$ and $\tau(f_j|e_i)$. We bin the model 1 parameters, and form integer-valued features for each bin that count the number of links with $\tau_0 < \tau(e_i|f_j) < \tau_1$.

Geometrical features are those which capture correlation between different σ_{ij} based on adjacency or nearness. They capture the idea that nearby words in one language link to nearby words in the other language - the motivation of HMM-based models of alignment. An example is a feature that counts the number of times that the next word in the source language is linked to the next word in the target language:

$$\phi(A,E,F) = \sum_{ij} \sigma_{ij}\sigma_{i+1,j+1} \quad (7)$$

Parse-based features are those which capture correlation between different σ_{ij}, but use parsing to determine links which are correlated - for example, if a determiner links to the same word as its head noun. As an example, if e_i is the headword of $e_{i'}$, and f_j is the headword of $f_{j'}$, then

$$\phi(A,E,F) = \sum_{ij} \sigma_{ij}\sigma_{i'j'} \quad (8)$$

counts the number of times that a dependency relation in one language is preserved by alignment in the other language. This feature can also be decorated, either lexically, or with part-of-speech tags (as many features in all three categories are.)

5.1 Unsupervised Adaptation

We constructed a heuristic phrase dictionary for unsupervised adapatation. After aligning a large unannotated parallel corpus with our aligner, we enumerate fully lexicalized geometrical features that can be extracted from the resulting alignments - these are entries in a phrase dictionary. These features are tied, and treated as a single real-valued feature that fires during training and decoding phases if a set of hypothesized links matches the geometrical feature extracted from the unannotated data. The value of this real-valued feature is the log of the number of occurrences of the identical (lexicalized) geometrical feature in the aligned unannotated corpus.

6 Results

We design our experiments to validate that a correction model using simple features, mostly non-language-specific, can improve the alignment accuracy of a variety of existing aligners for a variety of language pairs; we do not attempt to exactly match features between comparison aligners - this is unlikely to lead to a robust correction model.

6.1 Arabic-English alignment results

We trained the Arabic-English alignment system on 5125 sentences from Arabic-English treebanks (LDC2008E61, LDC2008E22) that had been annotated for word alignment. Reference parses were used during the training. Results are measured on a 500 sentence test set, sampled from a wide variety of parallel corpora, including various genres. During alignment, only automatically-generated parses (based on the parser of (Ratnaparkhi, 1999)) were available. Alignments on

initial align	correction model	R (%)	P (%)	$F(\%)$	ΔF
GIZA++		76	76	76	
	corr(GIZA++)	86	94	90	14*
	corr(ME-seq)	88	92	90	14*
HMM		73	73	73	
	corr(HMM)	87	92	89	16*
	corr(ME-seq)	87	93	90	17*
ME-seq		82	84	83	
	corr(HMM)	88	92	90	7*
	corr(GIZA++)	87	94	91	8*
	corr(ME-seq)	89	94	**91**	8*

Table 1: Alignment accuracy for Arabic-English systems in percentage recall (R), precision(P), and F-measure. * denotes statistical significance (see text.)

lang	method	R (%)	P(%)	$F(\%)$	ΔF
ZH→EN	GIZA++	55	67	61	
	ME-seq	66	72	69	
	corr(ME-seq)	74	76	**75**	6*

Table 2: Alignment accuracy for Chinese(ZH)-English(EN) systems. * denotes statistical significance

lang	aligner	R(%)	P(%)	$F(\%)$	ΔF
IT→ EN	ME-seq	74	87	80	
	corr(ME-seq)	84	92	88	8*
EN→IT	ME-seq	75	86	80	
	corr(ME-seq)	84	92	88	8*
PT→EN	ME-seq	77	83	80	
	corr(ME-seq)	87	91	89	9†
EN→PT	ME-seq	79	87	83	
	corr(ME-seq)	88	90	89	6†
JA→EN	ME-seq	72	78	75	
	corr(ME-seq)	77	83	80	5*
RU→EN	ME-seq	81	85	83	
	corr(ME-seq)	82	92	87	4*
DE→EN	ME-seq	77	82	79	
	corr(ME-seq)	78	87	82	3*
ES→EN	ME-seq	93	86	90	
	corr(ME-seq)	92	88	90	0.6
FR→EN	ME-seq	89	91	90	
	corr(ME-seq)	88	92	90	0.1

Table 3: Alignment accuracy for additional languages. * denotes statistical significance; † statistical significance not available. IT=Italian, PT=Portuguese, JA=Japanese, RU=Russian, DE=German, ES=Spanish, FR=French

the training and test sets were decoded with three other aligners, so that the robustness of the correction model to different input alignments could be validated. The three aligners were GIZA++ (Och and Ney, 2003) (with the MOSES (Koehn et al., 2007) postprocessing option `-alignment grow-diag-final-and`) the posterior HMM aligner of (Ge, 2004), a maximum entropy sequential model (ME-seq) (Ittycheriah and Roukos, 2005). ME-seq is our primary point of comparison: it is discriminatively trained (on the same training data,) uses a rich set of features, and provides the best alignments of the three. Three correction models were trained: corr(GIZA++) is trained to correct the alignments produced by GIZA++, corr(HMM) is trained to correct the alignments produced by the HMM aligner, and corr(ME-seq) is trained to correct the alignments produced by the ME-seq model.

In Table (1) we show results for our system correcting each of the aligners as measured in the usual recall, precision, and F-measure.[1] The resulting improvements in F-measure of the alignments produced by our models over their corresponding baselines is statistically significant ($p < 10^{-4}$, indicated by a *.) Statistical significance is tested by a Monte Carlo bootstrap (Efron and Tibshirani, 1986) - sampling with replacement the difference in F-measure of the two system's alignments of the same sentence pair. Both recall and precision are improved, but the improvement in precision is somewhat larger. We also show cross-condition results in which a correction model trained to correct HMM alignments is applied to correct ME-seq alignments. These results show that our correction model is robust to different starting aligners.

6.2 Chinese-English alignment results

Table (2) presents results for Chinese-English word alignments. The training set for the corr(ME-seq) model consisted of approximately 8000 hand-aligned sentences sampled from LDC2006E93 and LDC2008E57. The model was trained to correct the output of the ME-seq aligner, and tested on the same condition. For this language pair, reference parses were not available in our training set, so automatically-generated parses were used for both training and test sets. Results are measured on a 512 sentence test set, sampled from a wide variety of parallel corpora of various genres. We compare performance with GIZA++, and with the ME-seq aligner. Again the resulting improvement over the ME-seq aligner is statistically significant. However, here the improvement in recall is somewhat larger than the improvement in precision.

6.3 Additional language pairs

Table (3) presents alignment results for seven other language pairs. Separate alignment corrector models were trained for both directions of Italian ↔ English and Portuguese ↔ English. The training and test data vary by language, and are sampled uniformly from a diverse set of corpora of various genres, including newswire, and technical manuals. Manual alignments for training and test data were annotated. We compare performance with the ME-seq aligner trained on the same training data. As with the Chinese results above, customization and feature development for the language pairs was minimal. In general, machine parses were always available for the English half of the pair. Machine parses were also available for French and Spanish. Machine part of speech tags were available for all language (although character-based heuristic was substituted for Japanese.) Large amounts (up to 10 million sentence pairs) of unaligned parallel text was available for model 1 type features. Our model obtained improved alignment F-measure in all language pairs, although the improvements were small for ES→EN and FR→EN, the language pairs for which the baseline accuracy was the highest.

6.4 Analysis

Some of the improvement can be attributed to "look-ahead" during the decoding. For example, the English word "the", which (during Arabic-English alignment) should often be aligned to the same Arabic words to which its headword is linked. The number of errors associated with "the" dropped from 383 (186 false alarms, 197 misses) in the ME-seq model to 137 (60 false alarms and 77 misses) in the current model.

In table 5, we show contributions to performance resulting from various classes of features. The

[1] We do not distinguish sure and possible links in our annotations - under this circumstance, alignment error rate(Och and Ney, 2003) is $1 - F$.

		Zh-En			Ar-En	
method	correct	miss	fa	correct	miss	fa
hmm				147	256	300
GIZA++	139	677	396	132	271	370
ME-seq	71	745	**133**	127	276	191
corr(ME-seq)	**358**	**458**	231	**264**	**139**	**114**

Table 4: Analysis of $2-1$ alignments errors (misses and false alarms) for Zh-En and Ar-En aligners

largest contribution is noted by removing features based on the Model 1 translation matrices. These features contain a wealth of lexical information learned from approximately 7×10^6 parallel sentences - information that cannot be learned from a relatively small amount of word-aligned training data. Geometrical features contribute more than parse-based features, but the contribution from parse-based features is important, and these are more difficult to incorporate into sequential models. We note that all of the comparison aligners had equivalent lexical information.

We show a small improvement from the unsupervised adaptation - learning phrases from the parallel corpus that are not captured by the lexical features based on model 1. The final row in the table shows the result of running the correction model on its own output. The improvement is not statistically significant, but it is important to note the performance is *stable* - a further indication that the model is robust to a wide variety of input alignments, and that our decoding scheme is a reasonable approach to finding the best alignment.

In table 4, we characterize the errors based on the fertility of the source and target words. We focus on the case that exactly one target word is linked to exactly two source words. These are the links that are poorly suited for the HMM and ME-seq models used in this comparison because of the chosen directionality: the source (Arabic, Chinese) words are the states and the target (English) words are the observation. The HMM is able to produce these links only by the use of posterior probabilities, rather than viterbi decoding. The ME-seq model only produces these links because of language-specific post-processing. GIZA++ has an underlying sequential model, but uses both directionalities. The correction model improved performance across all three of these links structures. The single exception is that the number of $2-1$ false alarms increased (Zh-En alignments) but in this case, the first pass ME-seq alignment produced few false alarms because it simply proposed few links of this form. It is also notable that $1-2$ links are more numerous than $2-1$ links, in both language pairs. This is consequence of the choice of directionality and tokenization style.

feature	R(%)	P(%)	$F(\%)$	N_{exact}
base	89	94	91	136
base-M1	82	88	85	89
base-geometric	83	90	86	92
base-parse	87	93	90	116
base+un.adapt	89	94	92	141
+iter2	90	94	92	141

Table 5: Importance of feature classes - ablation experiments

	corpus-level		p90	
alignment	TER	BLEU	TER	BLEU
ME-seq	56.06	32.65	64.20	21.31
corr(Me-seq)	56.25	33.10	63.47	22.02
both	56.07	33.13	63.41	22.14

Table 6: Translation results, Zh to En. BLEU=BLEUr4n4

alignment	TER	BLEUr1n4
ME-seq	35.02	69.94
corr(Me-seq)	33.10	71.20

Table 7: Translation results, En to It

6.5 Translation Impact

We tested the impact of improved alignments on the performance of a phrase-based translation system (Ittycheriah and Roukos, 2007) for three language pairs. Our alignment did not improve the performance of a mature Arabic to English translation system, but two notable successes were obtained: Chinese to English, and English to Italian. It is well known that improved alignment performance does not always improve translation performance (Fraser and Marcu, 2007). A mature machine translation system may incorporate alignments obtained from multiple aligners, or from both directions of an asymmetric aligner. Furthermore, with large amounts of training data (the Gale Phase 4 Arabic English corpus consisting of 8×10^6 sentences,) a machine translation system is subject to a saturation effect: correcting an alignment may not yield a significant improvement because the the phrases learned from the correct alignment have already been acquired in other contexts.

For the Chinese to English translation system (table 6) the training corpus consisted of 11×10^6 sentence pairs, subsampled to 10^6. The test set was NIST MT08 Newswire, consisting of 691 sentences and 4 reference translations. Corpus-level performance (columns 2 and 3) improved when measured by BLEU, but not by TER. Performance on the most difficult sentences (near the 90th percentile, columns 4 and 5) improved on both BLEU and TER (Snover et al., 2006), and the improvement in BLEU was larger for the more difficult sentences than it was overall. Translation performance further improved, by a smaller amount, using both ME-seq and corr(ME-seq) alignments during the training.

The improved alignments impacted the translation performance of the English to Italian translation system (table 7) even more strongly. Here the training corpus consisted of 9.4×10^6 sentence pairs, subsampled to 387000 pairs. The test set consisted of 7899 sentences. Overall performance improved as measured by both TER and BLEU (1.26 points.)

7 Conclusions

A log linear model for the alignment matrix is used to guide systematic improvements to an existing aligner. Our system models arbitrary alignment matrices and allows features that incorporate such information as correlations based on parse trees in both languages. We train models to correct the errors of several existing aligners; we find the resulting models are robust to using different aligners as starting points. Improvements in alignment F-measure, often significant improvements, show that our model successfully corrects input alignments from existing models in all nine language pairs tested. The resulting Chinese-English and English-Italian word alignments also improved translation performance, especially on the English-Italian test, and notably on the particularly difficult subset of the Chinese sentences. Future work will assess its impact on translation for the other language pairs, as well as its impact on other tasks, such as named entity projection.

8 Acknowledgements

We would like to acknowledge the support of DARPA under Grant HR0011-08-C-0110 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. 2005. Alignment link projection using transformation-based learning. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 185–192, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yassine Benajiba and Imed Zitouni. 2010. Enhancing mention detection using projection via aligned corpora. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 993–1001. Association for Computational Linguistics.

Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *In Proc. of ACL-2006*, pages 65–72.

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

Yonggang Deng and Bowen Zhou. 2009. Optimizing word alignment combination for phrase table training. In *Proceedings of the ACL-IJCNLP 2009 Conference*

Short Papers, ACLShort '09, pages 229–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

B. Efron and R. Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):pp. 54–75.

Victoria Fossum, Kevin Knight, and Steven Abney. 2008. Using syntax to improve word alignment precision for syntax-based machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 44–52. Association for Computational Linguistics.

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

Niyu Ge. 2004. Improvement in word alignments. In *DARPA/TIDES MT workshop*.

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

Abraham Ittycheriah and Salim Roukos. 2007. Direct translation model 2. In *Human Language Technologies 2007: The Conference of the NA-ACL*, pages 57–64, Rochester, New York, April. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers on XX*, HLT-NAACL '04, pages 57–60. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

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

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

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *In Proceedings of HLT-EMNLP*, pages 81–88.

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

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Mach. Learn.*, 34:151–175, February.

Hendra Setiawan, Chris Dyer, and Philip Resnik. 2010. Discriminative word alignment with a function word reordering model. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 534–544. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Ben Taskar, Simon Lacoste-julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *In Proceedings of HLT-EMNLP*, pages 73–80.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, HLT '01, pages 1–8. Association for Computational Linguistics.