Alignment Link Projection Using Transformation-Based Learning

Necip Fazil Ayan, Bonnie J. Dorr and **Christof Monz**
Department of Computer Science
University of Maryland
College Park, MD 20742
{nfa,bonnie,christof}@umiacs.umd.edu

Abstract

We present a new word-alignment approach that learns errors made by existing word alignment systems and corrects them. By adapting transformation-based learning to the problem of word alignment, we project new alignment links from already existing links, using features such as POS tags. We show that our alignment link projection approach yields a significantly lower alignment error rate than that of the best performing alignment system (22.6% relative reduction on English-Spanish data and 23.2% relative reduction on English-Chinese data).

1 Introduction

Word-level alignment is a critical component of a wide range of NLP applications, such as construction of bilingual lexicons (Melamed, 2000), word sense disambiguation (Diab and Resnik, 2002), projection of language resources (Yarowsky et al., 2001), and statistical machine translation. Although word-level aligners tend to perform well when there is *enough* training data, the quality of word alignment decreases as the size of training data decreases. Moreover, word-alignment systems are often tripped up by many-to-many correspondences, morphological language distinctions, paraphrased and free translations, and a high percentage of function words (about 50% of the tokens in most texts).

At the heart of the matter is a set of assumptions that word-alignment algorithms must make in order to reduce the hypothesis space, since word alignment is an exponential problem. Because of these assumptions, learning algorithms tend to make similar errors throughout the entire data.

This paper presents a new approach—*Alignment Link Projection (ALP)*—that learns common alignment errors made by an alignment system and attempts to correct them. Our approach assumes the initial alignment system adequately captures certain kinds of word correspondences but fails to handle others. ALP starts with an initial alignment and then fills out (i.e., *projects*) new word-level alignment relations (i.e., *links*) from existing alignment relations. ALP then deletes certain alignment links associated with common errors, thus improving precision and recall.

In our approach, we adapt transformation-based learning (TBL) (Brill, 1995; Brill, 1996) to the problem of word alignment. ALP attempts to find an ordered list of transformation rules (within a prespecified search space) to improve a baseline annotation. The rules decompose the search space into a set of consecutive words (windows) within which alignment links are added, to or deleted from, the initial alignment. This window-based approach exploits the clustering tendency of alignment links, i.e., when there is a link between two words, there is frequently another link in close proximity.

TBL is an appropriate choice for this problem for the following reasons:

1. It can be optimized directly with respect to an evaluation metric.
2. It learns rules that improve the initial prediction iteratively, so that it is capable of correcting previous errors in subsequent iterations.
3. It provides a readable description (or classification) of errors made by the initial system, thereby enabling alignment refinements.

The rest of the paper is organized as follows: In the next section we describe previous work on improving word alignments. Section 3 presents a brief overview of TBL. Section 4 describes the adaptation of TBL to the word alignment problem. Section 5 compares ALP to various alignments and presents results on English-Spanish and English-Chinese. We show that ALP yields a significant reductions in alignment error rate over that of the best performing alignment system.

2 Related Work

One of the major problems with the IBM models (Brown et al., 1993) and the HMM models (Vogel et al., 1996) is that they are restricted to the alignment of each source-language word to at most one target-language word. The standard method to overcome this problem to use the model in both directions (interchanging the source and target languages) and applying heuristic-based combination techniques to produce a *refined alignment* (Och and Ney, 2000; Koehn et al., 2003)—henceforth referred to as "RA."

Several researchers have proposed algorithms for improving word alignment systems by injecting additional knowledge or combining different alignment models. These approaches include an enhanced HMM alignment model that uses part-of-speech tags (Toutanova et al., 2002), a log-linear combination of IBM translation models and HMM models (Och and Ney, 2003), techniques that rely on dependency relations (Cherry and Lin, 2003), and a log-linear combination of IBM Model 3 alignment probabilities, POS tags, and bilingual dictionary coverage (Liu et al., 2005). A common theme for these methods is the use of additional features for enriching the alignment process. These methods perform better than the IBM models and their variants but still tend to make similar errors because of the bias in their alignment modeling.

We adopt an approach that post-processes a given alignment using linguistically-oriented rules. The idea is similar to that of Ayan et al. (2004), where manually-crafted rules are used to correct alignment links related to language divergences. Our approach differs, however, in that the rules are extracted automatically—not manually—by examining an initial alignment and categorizing the errors according to features of the words.

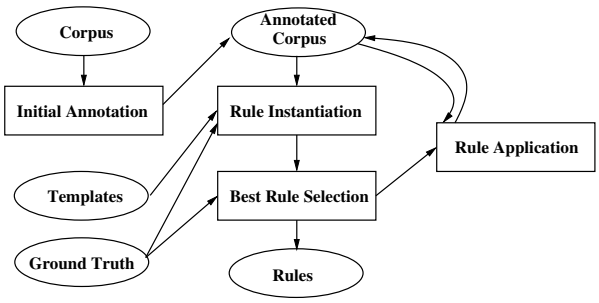

Figure 1: TBL Architecture

3 Transformation-based Learning

As shown in Figure 1, the input to TBL is an unannotated corpus that is first passed to an initial annotator and then iteratively updated through comparison to a manually-annotated reference set (or *ground truth*). On each iteration, the output of the previous iteration is compared against the ground truth, and an ordered list of transformation rules is learned that make the previous annotated data better resemble the ground truth.

A set of *rule templates* determines the space of allowable transformation rules. A rule template has two components: a triggering environment (condition of the rule) and a rewrite rule (action taken). On each iteration, these templates are instantiated with features of the constituents of the templates when the condition of the rule is satisfied.

This process eventually identifies all possible instantiated forms of the templates. Among all these possible rules, the transformation whose application results in the best score—according to some objective function—is identified. This transformation is added to the ordered list of transformation rules. The learning stops when there is no transformation that improves the current state of the data or a prespecified threshold is reached.

When presented with new data, the transformation rules are applied in the order that they were added to the list of transformations. The output of the system is the annotated data after all transformations are applied to the initial annotation.

4 Alignment Link Projection (ALP)

ALP is a TBL implementation that projects alignment links from an initial input alignment. We induce several variations of ALP by setting four parameters in different ways:

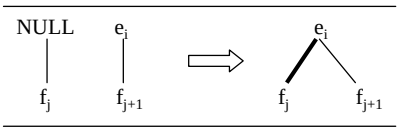

Figure 2: Graphical Representation of a Template

1. Initial alignment
2. Set of templates
3. Simple or generalized instantiation
4. Best rule selection

We describe each of these below using the following definitions and notation:

- $E = e_1, \ldots, e_i, \ldots, e_t$ is a sentence in language L_1 and $F = f_1, \ldots, f_j, \ldots, f_s$ is a sentence in language L_2.
- An *alignment link* (i, j) corresponds to a translational equivalence between e_i and f_j.
- A *neighborhood* of an alignment link (i, j)—denoted by $N(i, j)$—consists of 8 possible alignment links in a 3×3 window with (i, j) in the center of the window. Each element of $N(i, j)$ is called a *neighboring link* of (i, j).
- $nullE_A(i)$ is *true* if and only if e_i is not aligned to any word in F in a given alignment A. Similarly, $nullF_A(j)$ is *true* if and only if f_j is not aligned to any word in E in a given alignment A.

4.1 Initial Alignment

Any existing word-alignment system may be used for the initial annotation step of the TBL algorithm. For our experiments, we chose GIZA++ (Och and Ney, 2000) and the RA approach (Koehn et al., 2003)— the best known alignment combination technique— as our initial aligners.[1]

4.2 TBL Templates

Our templates consider consecutive words (of size 1, 2 or 3) in both languages. The condition portion of a TBL rule template tests for the existence of an alignment link between two words. The action portion involves the addition or deletion of an alignment link. For example, the rule template in Figure 2 is applicable only when a word (e_i) in one language is aligned to the second word (f_{j+1}) of a phrase (f_j, f_{j+1}) in the other language, and the first word (f_j) of the phrase is unaligned in the initial alignment. The action taken by this rule template is to add a link between e_i and f_j.[2]

ALP employs 3 different sets of templates to project new alignment links or delete existing links in a given alignment:

1. Expansion of the initial alignment according to another alignment
2. Deletion of spurious alignment links
3. Correction of multi-word (one-to-many or many-to-one) correspondences

Each of these is described below.

4.2.1 Expansion Templates

Expansion templates are used to extend an initial alignment given another alignment as the validation set. This approach is similar to the one used in the RA method in that it adds links based on knowledge about neighboring links, but it differs in that it *also* uses features of the words themselves to decide which neighboring links to add.

Our expansion templates are presented in Table 1. The first 8 templates add a new link to the initial alignment A if there is a neighboring link in the validation alignment V. The final two templates enforce the presence of at least two neighboring links in the validation set V before adding a new link.

Condition	**Action**
$(i,j) \in A, (i-1,j-1) \in V$	add $(i-1,j-1)$
$(i,j) \in A, (i-1,j) \in V$	add $(i-1,j)$
$(i,j) \in A, (i-1,j+1) \in V$	add $(i-1,j+1)$
$(i,j) \in A, (i,j-1) \in V$	add $(i,j-1)$
$(i,j) \in A, (i,j+1) \in V$	add $(i,j+1)$
$(i,j) \in A, (i+1,j-1) \in V$	add $(i+1,j-1)$
$(i,j) \in A, (i+1,j) \in V$	add $(i+1,j)$
$(i,j) \in A, (i+1,j+1) \in V$	add $(i+1,j+1)$
$(i-1,j-1) \in A, (i+1,j+1) \in A, (i,j) \in V$	add (i,j)
$(i+1,j-1) \in A, (i-1,j+1) \in A, (i,j) \in V$	add (i,j)

Table 1: Templates for Expanding the Alignment A According to a Validation Alignment V

4.2.2 Deletion Templates

Existing alignment algorithms (e.g., GIZA++) are biased toward aligning some words, especially infrequent ones, in one language to many words in the other language in order to minimize the number of unaligned words, even if many incorrect alignment

[1] We treat these initial aligners as black boxes.

[2] A thick line indicates an added link.

links are induced.[3] Deletion templates are useful for eliminating the resulting spurious links.

The basic idea is to remove alignment links that do not have a neighboring link if the word in question has already been aligned to another word. Table 2 lists two simple templates to clean up spurious links. We define the predicate $neighbor\_exists_A(i,j)$ to denote whether there is an alignment link in the neighborhood of the link (i,j) in a given alignment A. For example, the first template deletes spurious links for a particular word e_i in E.

Condition	Action
$(i,j) \in A, (i,k) \in A,$ $neighbor\_exists_A(i,j),$ $not(neighbor\_exists_A(i,k))$	del (i,k)
$(i,j) \in A, (k,j) \in A,$ $neighbor\_exists_A(i,j),$ $not(neighbor\_exists_A(k,j))$	del (e,j)

Table 2: Templates for Deleting Spurious Links in a Given Alignment A

4.2.3 Multi-Word Correction Templates

Current alignment algorithms produce one-to-one word correspondences quite successfully. However, accurate alignment of phrasal constructions (many-to-many correspondences) is still problematic. On the one hand, the ability to provide *fully* correct phrasal alignments is impaired by the occurrence of high-frequency function words and/or words that are not exact translations of the words in the other language. On the other hand, we have observed that most alignment systems are capable of providing *partially* correct phrasal alignments.[4]

Our templates for handling multi-word correspondences are grounded in the outcome of this finding. That is, we make the (frequently correct) assumption that at least one alignment link in a many-to-many correspondence is correctly identified in the initial alignment. Table 3 lists the templates for correcting alignment links in multi-word correspondences. The first five templates handle ($e_i \rightarrow f_j f_{j+1}$) correspondences, the next five handle ($e_i e_{i+1} \rightarrow f_j$) correspondences, the next four handle ($e_i e_{i+1} \rightarrow f_j f_{j+1}$) correspondences, and the final two handle ($e_{i-1} e_i e_{i+1} \rightarrow f_j$) and ($e_i \rightarrow f_{j-1} f_j f_{j+1}$) correspondences.

Condition	Action
$nullF_A(j), (i,j+1) \in A$	add (i,j)
$nullF_A(j+1), (i,j) \in A$	add $(i,j+1)$
$(i,j) \in A, (i,j+1) \in A$	del (i,j)
$(i,j) \in A, (i,j+1) \in A$	del $(i,j+1)$
$nullF_A(j), nullF_A(j+1)$	add (i,j), add $(i,j+1)$
$nullE_A(i), (i+1,j) \in A$	add (i,j)
$nullE_A(i+1), (i,j) \in A$	add $(i+1,j)$
$(i,j) \in A, (i+1,j) \in A$	del (i,j)
$(i,j) \in A, (i+1,j) \in A$	del $(i+1,j)$
$nullE_A(i), nullE_A(i+1)$	add (i,j) add $(i+1,j)$
$(i+1,j+1) \in A$ $nullE_A(i), nullF_A(j),$	add (i,j)
$(i,j) \in A, nullE_A(i+1),$ $nullF_A(j+1)$	add $(i+1,j+1)$
$(i,j) \in A, (i+1,j) \in A,$ $(i+1,j+1) \in A$	add $(i,j+1)$
$(i,j) \in A, (i,j+1) \in A,$ $(i+1,j+1) \in A$	add $(i+1,j)$
$(i-1,j) \in A, (i+1,j) \in A$ $nullE_A(i)$	add (i,j)
$(i,j-1) \in A, (i,j+1) \in A$ $nullF_A(j)$	add (i,j)

Table 3: Templates for Handling Multi-Word Correspondences in a Given Alignment A

Condition	Action
$(i,j) \in A$	del (i,j)
$nullE_A(i), nullF_A(j)$	add (i,j)

Table 4: Templates for Correcting One-to-One Correspondences in a Given Alignment A

The alignment rules given above may introduce errors that require additional cleanup. Thus, we introduce two simple templates (shown in Table 4) to accommodate the deletion or addition of links between a single pair of words.

4.3 Instantiation of Templates

ALP starts with a set of templates and an initial alignment and attempts to instantiate the templates during the learning process. The templates can be instantiated using two methods: Simple (a word is instantiated with a specific feature) or Generalized (a word is instantiated using a special keyword `any-`

[3] This is a well-known characteristic of statistical alignment systems—motivated by the need to ensure a target-word translation e_i for each source word f_j while modeling $p(F|E)$ —for downstream MT.

[4] Specifically, we conducted a preliminary study using 40 manually-aligned English-Spanish sentences from a mixed corpus (UN + Bible + FBIS) as our gold standard. We found that, in most cases where the human annotator aligned one word to two words, an existing alignment system identified at least one of the two alignment links correctly.

`thing`).

ALP requires only a small amount of manually aligned data for this process—a major strength of the system. However, if we were to instantiate the templates with the actual words of the manual alignment, the frequency counts (from such a small data set) would not be high enough to derive reasonable generalizations. Thus, ALP adds new links based on linguistic features of words, rather than the words themselves. Using these features is what sets ALP apart from systems like the RA approach. Specifically, three features are used to instantiate the templates:

- **POS tags on both sides**: We assign POS tags using the MXPOST tagger (Ratnaparkhi, 1996) for English and Chinese, and Connexor for Spanish.
- **Dependency relations**: ALP utilizes dependencies for a better generalization—if a dependency parser is available in either language. In our experiments, we used a dependency parser only in English (a version of the Collins parser (Collins, 1997) that has been adapted for building dependencies) but not in the other language.
- **A set of closed-class words**: We use 16 different classes, 9 of which are different semantic verb classes while the other 7 are function words, prepositions, and complementizers.[5]

If both POS tags and dependency relations are available, they can be used together to instantiate the templates. That is, a word can be instantiated in a TBL template with: (1) a POS tag (e.g., Noun, Adj); (2) a relation (e.g., Subj, Obj); (3) a parameter class (e.g., Change of State); or (4) different subsets of (1)–(3). We also employ a more generalized form of instantiation, where words in the templates may match the keyword `anything`.

4.4 Best Rule Selection

The rules are selected using two different metrics: The accuracy of the rule or the overall impact of the application of the rule on the entire data.

Two different mechanisms may be used for selecting the best rule after generating all possible instantiations of templates:

1. **Rule Accuracy:** The goal is to minimize the errors introduced by the application of a transformation rule. To measure accuracy of a rule r, we use $good(r) - 2 \times bad(r)$, where $good(r)$ is the number of alignment links that are corrected by the rule, and $bad(r)$ is the number of incorrect alignment links produced.
2. **Overall impact on the training data:** The accuracy mechanism (above) is useful for biasing the system toward higher precision. However, if the overall system is evaluated using a metric other than precision (e.g., recall), the accuracy mechanism may not guarantee that the best rule is chosen at each step. Thus, we choose the best rule according to the evaluation metric to be used for the overall system.

5 Experiments and Results

This section describes our evaluation of ALP variants using different combinations of settings of the four parameters described above. The two language pairs examined are English-Spanish and English-Chinese.

5.1 Evaluation Metrics

Let A be the set of alignment links for a set of sentences. We take S to be the set of sure alignment links and P be the set of probable alignment links (in the gold standard) for the same set of sentences. Precision (Pr), recall (Rc) and alignment error rate (AER) are defined as follows:

$$Pr = \frac{|A \cap P|}{|A|} \quad Rc = \frac{|A \cap S|}{|S|}$$

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

A manually aligned corpus is used as our gold standard. For English-Spanish data, the manual annotation was done by a bilingual English-Spanish speaker. Every link in the English-Spanish gold standard is considered a sure alignment link.

For English-Chinese, we used 2002 NIST MT evaluation test set, and each sentence pair was aligned by two native Chinese speakers who are fluent in English. Each alignment link appearing in both annotations was considered a sure link, and

[5] These are based on the parameter classes of (Dorr et al., 2002).

links appearing in only one set were judged as probable. The annotators were not aware of the specifics of our approach.

5.2 Evaluation Data

We evaluated ALP using 5-fold cross validation on two different data sets:

1. A set of 199 English-Spanish sentence pairs (nearly 5K words on each side) from a mixed corpus (UN + Bible + FBIS).
2. A set of 491 English-Chinese sentence pairs (nearly 13K words on each side) from 2002 NIST MT evaluation test set.

We divided the pairs of sentences randomly into 5 groups. Then, for each fold, we used 4 groups as the ground truth (for training), and used the other group as our gold standard (for evaluation). This process was repeated 5 times so that each sentence pair was tested exactly once. We computed precision, recall and error rate on the entire set for each data set.[6]

For an initial alignment, we used GIZA++ in both directions (E-to-F and F-to-E, where F is either Chinese (C) or Spanish (S)), and also two different combined alignments: intersection of E-to-F and F-to-E; and RA using a heuristic combination approach called *grow-diag-final* (Koehn et al., 2003).

For the English-Spanish experiments, GIZA++ was trained on 48K sentence pairs from a mixed corpus (UN + Bible + FBIS), with nearly 1.2M of words on each side, using 10 iterations of Model 1, 5 iterations of HMM and 5 iterations of Model 4. For the English-Chinese experiments, we used 107K sentence pairs from FBIS corpus (nearly 4.1M English and 3.3M Chinese words) to train GIZA++, using 5 iterations of Model 1, 5 iterations of HMM, 3 iterations of Model 3, and 3 iterations of Model 4.

5.3 Results for English-Spanish

For our initial alignments we used: (1) Intersection of GIZA++ English-to-Spanish and Spanish-to-English; (2) GIZA++ English-to-Spanish; (3) GIZA++ Spanish-to-English; and (4) RA. Of these, RA is the best, with an error rate of 21.2%. For ease of comparison, the RA score appears in all result tables below.

[6]The number of alignment links varies over each fold. Therefore, we chose to evaluate all data at once instead of evaluating on each fold and then averaging.

Tables 5–7 compare ALP to each of these four alignments using different settings of 4 parameters: ALP[*IA, T, I, BRS*], where *IA* is the initial alignment, T is the set of templates, I is the instantiation method, and *BRS* is the metric for the best rule selection at each iteration. T_E is the set of expansion templates from Table 1, T_D is the set of deletion templates from Table 2, and T_{MW} is the set of multi-word templates from Table 3 (supplemented with templates from Table 4).

As mentioned in Section 4.3, we use two instantiation methods: (1) simple instantiation (sim), where the words are instantiated using a specific POS tag, relation, parameter class or combination of those; and (2) generalized instantiation (gen), where the words can be instantiated using the keyword `anything`. Two different metrics are used to select the best rule: The accuracy of the rule (acc) and the AER on the entire training data after applying the rule (aer).[7]

We performed statistical significance tests using two-tailed paired t-tests. Unless otherwise indicated, the differences between ALP and initial alignments (for all ALP variations and all initial alignments) were found to be statistically significant within the 95% confidence interval. Moreover, the differences among ALP variations themselves were statistically significant within 95% confidence interval.

Using Intersection as Initial Alignment We ran ALP using the intersection of GIZA++ (E-to-S) and GIZA++(S-to-E) alignments as the initial alignment in two different ways: (1) With T_E using the union of the unidirectional GIZA++ alignments as the validation set, and (2) with T_D and T_{MW} applied one after another. Table 5 presents the precision, recall and AER results.

Alignments	Pr	Rc	AER
Intersection (Int)	**98.2**	59.6	25.9
ALP[Int, T_E, gen, aer]	90.9	69.9	21.0
ALP[$Int, (T_D, T_{MW}), gen, aer$]	88.8	72.3	**20.3**
RA	83.8	**74.4**	21.2

Table 5: ALP Results Using GIZA++ Intersection as Initial Alignment for English-Spanish

Using the expansion templates (T_E) against a val-

[7]We use only sure alignment links as the ground truth to learn rules inside ALP. Therefore, AER here refers to the AER of sure alignment links.

Alignments	Pr	Rc	AER
E-to-S	**87.0**	67.0	24.3
ALP[E-to-S,$(T_D, T_{MW}), gen, aer$]	85.6	**76.4**	**19.3**
S-to-E	**88.0**	67.5	23.6
ALP[S-to-E,$(T_D, T_{MW}), gen, aer$]	87.1	**76.7**	**18.4**
RA	83.8	74.4	21.2

Table 6: ALP Results Using GIZA++ (Each Direction) as Initial Alignment for English-Spanish

Alignments	Pr	Rc	AER
ALP[$RA, (T_D, T_{MW}), sim, acc$]	87.8	77.7	17.6
ALP[$RA, (T_D, T_{MW}), sim, aer$]	**87.9**	79.0	16.8
ALP[$RA, (T_D \cup T_{MW}), gen, aer$]	86.2	80.0	17.0
ALP[$RA, (T_D, T_{MW}), gen, aer$]	86.9	**80.5**	**16.4**
RA	83.8	74.4	21.2

Table 7: ALP Results Using RA as Initial Alignment for English-Spanish

idation set produced results comparable to the RA method. The major difference is that ALP resulted in a much higher precision but in a lower recall because ALP is more selective in adding a new link during the expansion stage. This difference is due to the additional constraints provided by word features. The version of ALP that applies deletion (T_D) and multi-word (T_{MW}) templates sequentially achieves lower recall but higher precision than RA. In the best case, ALP achieves a statistically significant relative reduction of 21.6% in AER over the Intersection alignment. When compared to RA, ALP achieves a lower AER but the difference is not significant.

Using Unidirectional GIZA++ Alignments as Initial Alignment In a second set of experiments, we applied ALP to the unidirectional GIZA++ alignments, using deletion (T_D) and multi-word (T_{MW}) templates, generalized instantiation, and AER for the best rule selection. Table 6 presents the precision, recall and AER results.

For both directions, ALP achieves a lower precision but much higher recall than that of the initial unidirectional alignment. Overall, there was a relative reduction of 20.6–22.0% in AER. When compared to RA, the version of ALP that uses unidirectional GIZA++ alignments brings about significant reductions in AER: 9.0% relative reduction in one direction and 13.2% relative reduction in the other direction.

Using RA as Initial Alignment In a third experiment, we compared RA with variations of ALP using RA as the initial alignment. We used the templates in two different ways: (1) with a combination of T_D and T_{MW} (i.e., $T_D \cup T_{MW}$), and (2) with two consecutive runs of ALP, first with T_D and then with T_{MW} using the output of the first run as the initial annotation in the second run (i.e., T_D, T_{MW}). Table 7 presents precision, recall and AER results, using different methods for template instantiation and best rule selection.

The results indicate that using AER is better than using accuracy for choosing the best rule. Using generalized instantiation instead of simple instantiation results in a better AER. Running ALP with deletion (T_D) templates followed by multi-word (T_{MW}) templates results in a lower AER than running ALP only once with combined templates.

The highest performing variant of ALP, shown in the fourth line of the table, uses RA as the initial alignment, template sets T_D, T_{MW}, generalized instantiation, and AER for best rule selection. This variant is significantly better than RA, with a 22.6% relative reduction in AER. When compared to the unidirectional alignments (E-to-S and S-to-E) given in Table 6, this variant of ALP yields nearly the same precision (around 87.0%) but a 19.2% relative improvement in recall. The overall relative reduction in AER is 30.5% in the S-to-E direction and 32.5% in the E-to-S direction.

5.4 Results for English-Chinese

Our experiments for English-Chinese were designed with a similar structure to that of English-Spanish, i.e., the same four initial alignments. Once again, RA performs the best out of these initial alignments, with an error rate of 29.7%. The results of the initial alignments, and variations of ALP based on different initial alignments are shown in Table 8. For brevity, we include only the ALP parameter settings resulting in the best configurations from the English-Spanish experiments. For learning rules from the templates, we used only the sure alignment links as the ground truth while learning rules inside ALP.

On the English-Chinese data, ALP yields significantly lower error rates with respect to the initial alignments. When ALP is run with the intersection of two GIZA++ alignments, the relative reduction is 5.4% in AER. When ALP is run with E-to-C as initial alignment, the relative reduction in AER is 13.4%. For the other direction, ALP produces a rel-

Alignments	Pr	Rc	AER
Intersection (Int)	**94.8**	53.6	31.2
ALP[$Int, (T_D, T_{MW}), gen, aer$]	91.7	**56.8**	**29.5**
E-to-C	70.4	**68.3**	30.7
ALP[E-to-C,$(T_D, T_{MW}), gen, aer$]	**79.1**	68.1	**26.6**
C-to-E	66.0	**69.8**	32.2
ALP[C-to-E,$(T_D, T_{MW}), gen, aer$]	**83.3**	66.0	**26.2**
RA	61.9	**82.6**	29.7
ALP[RA,$(T_D, T_{MW}), gen, aer$]	**82.1**	72.7	**22.8**

Table 8: ALP Results Using Different Initial Alignments for English-Chinese

ative reduction of 18.6% in AER. Finally, when RA is given to ALP as an initial alignment, ALP results in a relative reduction of 23.2% in AER. When compared to RA, all variations of ALP, except the one starting with the intersection, yield statistically significantly lower AER. Another important finding is that ALP yields significantly higher precision than the initial alignments but usually lower recall.

6 Conclusion

We have presented ALP, a new approach that refines alignments by identifying the types of errors made by existing alignment systems and correcting them. Our approach adapts TBL to the problem of word-level alignment by examining word features as well as neighboring links. We use POS tags, closed-class words in both languages, and dependency relations in one language to classify the errors made by the initial alignment system. We show that ALP yields at least a 22.6% relative reduction on English-Spanish data and 23.2% relative reduction on English-Chinese data in alignment error rate over that of the best performing system.

We should note that ALP is not a stand-alone word alignment system but a supervised learning approach to improve already existing alignment systems. ALP takes advantage of clustering of alignment links to project new links given a reasonable initial alignment. We have shown that ALP is quite successful in projecting alignment links for two different languages—Spanish and Chinese.

Statistical alignment systems are more successful with increasing amount of training data. Whether ALP improves the statistical alignment systems when they are trained on more data is an interesting research problem, which we plan to tackle in future.

Finally, we will evaluate the improved alignments in the context of an end-to-end application, such as machine translation.

Acknowledgments This work has been supported, in part, by ONR MURI Contract FCPO.810548265, Cooperative Agreement DAAD190320020, and NSF ITR Grant IIS-0326553.

References

Necip F. Ayan, Bonnie J. Dorr, and Nizar Habash. 2004. Multi-Align: Combining linguistic and statistical techniques to improve alignments for adaptable MT. In *Proceedings of AMTA'2004*, pages 17–26.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Eric Brill. 1996. Learning to parse with transformations. In *Recent Advances in Parsing Technology*. Kluwer Academic Publishers.

Peter F. Brown, Stephan A. Della-Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin. 2003. A probability model to improve word alignment. In *Proceedings of ACL'2003*, pages 88–95.

Micheal Collins. 1997. Three generative lexicalized models for statistical parsing. In *Proceedings of ACL'1997*.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of ACL'2002*.

Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa, and Nizar Habash. 2002. DUSTer: A method for unraveling cross-language divergences for statistical word–level alignment. In *Proceedings of AMTA'2002*.

Philip Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL/HLT'2003*.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of ACL'2005*.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Franz J. Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL'2000*, pages 440–447.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):9–51, March.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of EMNLP'1996*.

Kristina Toutanova, H. Tolga Ilhan, and Christopher D. Manning. 2002. Extensions to HMM-based statistical word alignment models. In *Proceedings of EMNLP'2002*, pages 87–94.

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING'1996*, 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 HLT'2001*, pages 109–116.