TakeTwo: A Word Aligner based on Self Learning

Jim Chang, Jian-Cheng Wu, Jason S. Chang
Department of Computer Science
National Tsing Hua University
101, Guangfu Road, Hsinchu, Taiwan
{jim.chang.nthu, wujc86, jason.jschang}@gmail.com

Abstract

State of the art statistical machine translation systems are typically trained by symmetrizing word alignments in two translation directions. We introduce a new method that improves word alignment results, based on self learning using the initial symmetrized word alignments results. The method involves aligning words and symmetrizing alignments, generating labeled training data, and construct a classifier for predicting word-translation relation in another alignment round. In the first alignment round, we use the original *grow-diag-final-and* procedure, while in the second round, we use the classifier and a modified GDFA procedure to validate and fill in alignment links. We present a prototype system, *TakeTwo*, which applies the method to improve on GDFA. Preliminary experiments and evaluation on a hand-annotated dataset show that the method significantly increases the precision rate by a wide margin (+16%) with comparable recall rate (-3%).

1 Introduction

The first statistical machine translation (SMT) models are the IBM models, based on statistics collected over a parallel corpus of translated text. These generative IBM models break up the translation process into a number of steps. The most important step is word translation, which is modelled by the lexical translation probability, trained from a parallel corpus, typically with the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977).

However, EM word aligners are data-hungry and produce noisy links due to data sparseness. Many researchers (e.g., Gale and Church 1992, Johnson et al., 2007) have pointed out that, even with a large parallel corpus, the EM algorithms running IBM models still produces noisy links for low frequency words and non-literal translations.

Koehn, Och, and Marcu (2003) propose an improved word alignment method based on running IBM models in both translation directions for the two languages involved, and symmetrizing the results using a so-called *grow-diag-final-and* (GDFA) procedure. In a nutshell, GDFA is a heuristic greedy algorithm that starts by accepting reliable links in the intersection of the two alignments. Then, GDFA attempts to add union links neighboring intersection links. Finally, other non-neighboring links are added, subject to 1-1 alignment constraint. This progressively expanding scheme substantially enhances word alignment accuracy. However, the GDFA procedure still leaves much room for improvement, especially for low-frequency translations, non-literal translations, and sentences with extraneous/deleted translations.

Consider the following English sentence with Mandarin Chinese translation in a parallel corpus:

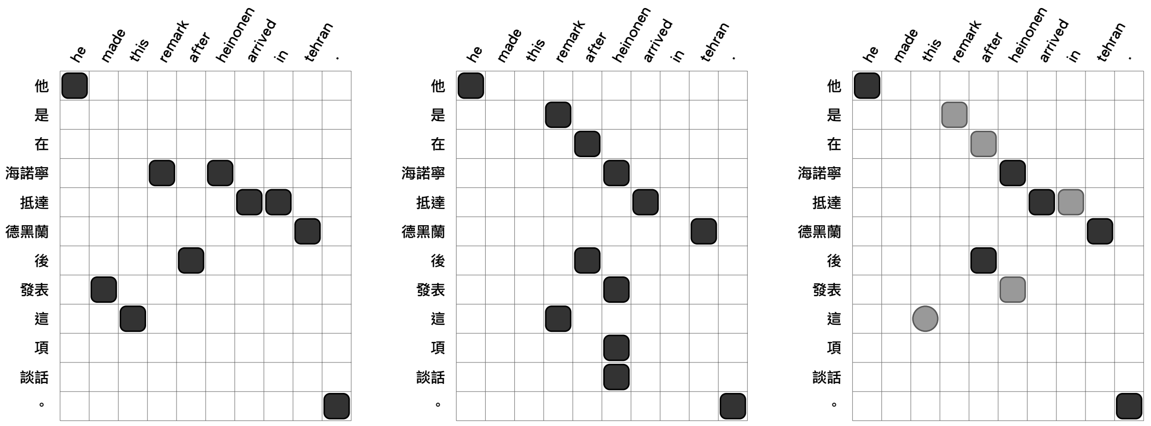
(1) *He made this remark after Heinonen arrived in Tehran.*

他	是	在	海諾寧	抵達	德黑蘭
ta	*shi*	*zai*	*hainuoning*	*dida*	*deheilan*
he	*is*	*when*	*Heinonen*	*arrive*	*Tehran*

後	發表	這	項	談話	。
hou	*fabiao*	*zhe*	*xiang*	*tanhua*	.
after	*deliver*	*this*	*MEASURE*	*talk*	.

See Figures 1(c) for examples of noisy and missing links, produced by *Giza++* with the GDFA sym-

28th Pacific Asia Conference on Language, Information and Computation pages 282–291



(a) (b) (c)

Figure 1: Three example alignments produced by *Giza*++ for Ex. (1): **(a)** Chinese-English alignment. **(b)** English-Chinese alignment. **(c)** The symmetrized alignment of combining (a) and (b) by running the *grow-diag-final-and* procedure. Note that the dark cells (in Figure 1(c)) represent links in the intersection of two alignments, while the gray cells represent links in the rest of the union.

metrizing procedure. For Example (1), a good word alignment should include *hard-to-align* links (e.g., [*made*, 發表 (fabiao)] and [*remark*, 談話 (tanhua)] (in addition to *easy* links (e.g., [*he*, 他 (ta)] and [*arrived*, 抵達 (dita)]), and exclude invalid union links like [*remark*, 是 (shi)] and [*heinonen*, 發表 (fabiao)] (picked up by GDFA, because they are neighbors of intersection links).

In Figure 1(c), a hard-to-align link [*remark*, 談話 (tanhua)] is missed out by GDFA, because [*remark*, 談話] are not common mutual translations (*remark* is commonly translated into 評論, while [談話(tanhua)] is commonly translated to *talk*). For the same reason, the missing link [*made*, 發表 (fabiao)] is also hard to align.

Intuitively, these hard-to-align links could be identified using a classifier for predicting word-translation relation, if we have sufficient training data. Ideally, we should avoid human effort in preparing the training data. Based on the concept of *self training*, we can generate slightly imperfect training data with the most reliable links (e.g, intersection links of the two initial sets of alignments) as positive instances, and very unreliable links as negative instances (e.g., [*hienonen*, 項 (xiang)] and [*hienonen*, 談話 (tanhua)] not picked up by GDFA).

We present a new system, *TakeTwo*, that uses the concept of self training to cope with translation variants and non-literal translations, aimed at improving on GDFA. An example *TakeTwo* alignment for Example (1) is shown in Figure 2. *TakeTwo* has used predicted word-translation probability to exclude invalid links [*remark*, 是] and [heinonen, 談話], and fill in valid links [*made*, 發表] and [*remark*, 談話], leading to an improved alignment.

The rest of the paper is organized as follows. We review the related work in the next section. Then we present our method for *TakeTwo* (Section 3). To evaluate the performance of *TakeTwo*, we compare the quality of alignments produced by *TakeTwo* with those produced by *Giza*++ with GDFA (Section 4 and Section 5) over a set of parallel sentences with hand-annotated word alignment.

2 Related Work

Machine translation (MT) has been an area of active research. (Dorr, 1993) summarizes various approaches to MT, while (Lopez, 2007) surveys recent work on statistical machine translation (SMT). We focus on the first part of developing an SMT system, namely, aligning words in a given parallel corpus.

The state of the art in word alignment focuses on automatically learning generative translation models via Expectation Maximization algorithm (Brown et al., 1990; Brown et al., 1993). (Och and Ney, 2003) describe Giza++, an implementation of the

Input: ... He made this remark after Heinonen arrived in Tehran.
他 是 在 海諾寧 抵達 德黑蘭 後 發表 這 項 談話 。 ...

Initial word alignments in two directions (En-Ch and Ch-En):

he(他) made this remark(是) after(在 後) heinonen(海諾寧 發表 項 談話) arrived(抵達) in tehran(德黑蘭)
他(he) 是 在 海諾寧(remark heinonen) 抵達(arrive in) 德黑蘭(tehran) 後(after) 發表(made) 這(this) 項 談話

Crosslingual relatedness:

x-sim(remark, 是) = *sim*(remark, be) = .0, *x-sim*(heinonen, 發表) = *sim*(heinonen, publish) = .0,
x-sim(made, 發表） = *sim*(make, publish) = .32, *x-sim*(remark, 談話) = *sim*(remark, talk) = .25

Output:

he(他) made(發表) this(這) remark(談話)
after(後) heinonen(海諾寧) arrived(抵達) in(抵達)
tehran(德黑蘭) . (。)

Alignment dotplot (see figure on the right)
Note that the dark cells represent links in the intersection of two alignments, while the gray cells represent links in the rest of the union

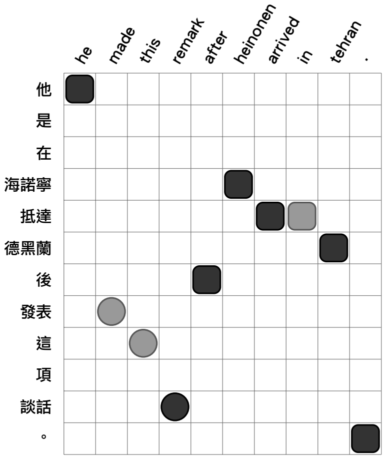


Figure 2: An example *TakeTwo* session and results

IBM models, which has since become the tool of choice for developing SMT systems.

As an alternative to the EM algorithm, researchers have been exploring various knowledge sources for word alignment, using automatically derived lexicons or handcrafted dictionaries (Gale and Church, 1991; Ker and Chang, 1997), or syntactic structure (Gildea, 2003; Cherry and Lin, 2003; Wang and Zong, 2013). There has been work on translating phrases using mixed-code web-pages (e.g., (Nagata et al., 2001; Wu and Chang, 2007)). Similarly, (Lin et al., 2008) propose a method that performs word alignment for parenthetic translation phrases to improve the performance of SMT systems.

Researchers have also studied sublexical models for machine transliteration (Knight and Graehl, 1998). More recently, (Chang et al., 2012) introduce a method for learning a CRF model to find translations and transliterations of technical terms on the Web. We use similar transliteration-based features derived from transliteration model in a different setting.

Word alignment is closely related to measuring word similarity, and especially in the form of crosslingual relatedness. Much work has been done on word similarity and crosslingual relatedness. Early research efforts have been devoted to design the knowledge-based measures, based, in particular, on WordNet (Fellbaum, 1999). Researchers have extensively investigated WordNet and other taxonomic structure in an attempt to calculate the word similarity by counting conceptual distance (Lin, 1998b). On the other hand, there has been much work on distributional word similarity, for example, (Lin, 1998a).

In the area of cross-lingual relatedness, (Michelbacher et al., 2010) present a graph-based method for building a a cross-lingual thesaurus. The method uses two monolingual corpora and a basic dictionary to build two monolingual word graphs, with nodes representing words and edges representing linguistic relations between words.

In the research area of supervised training for word alignment, (Moore, 2005) demonstrates that a discriminative model with the main feature of Log Likelihood Ratio (LLR) could result in a smaller model comparable to more complex generative EM models in alignment accuracy. (Taskar et al., 2005) independently propose a similar approach. (Liu et al., 2005) also propose a log-linear model incorporating features (alignment probability, POS correspondence and bilingual dictionary coverage).

The main difference from our current work is that previous methods use manually labeled data (typically hundreds sentences with thousands of word-translation relations) to train a word alignment model. In contrast, we take a self learning approach and automatically generate labelled training data. More specifically, We train our model based on a much larger training set (hundred of thousand of word-translation instances in partially labeled sentences) based on self learning.

Recently, some researchers have begun using syntax in word alignment, by incorporating features such as inversion transduction grammar or parse tree. Supervised (Cherry and Lin, 2006; Setiawan et al., 2010) and unsupervised (Pauls et al., 2010) methods have been proposed, showing that syntax can improve alignment performance. All these features can be used to training the classifier used in *TakeTwo*.

In a word alignment approach closer to our method, (Deng and Zhou, 2009) propose a method to optimize word alignment combination to derive a more effective phrase table. Similarly, (Nakov and Tiedemann, 2012) propose combining word-level and character-Level alignment models for improving machine translation between two closely-related languages.

In contrast to the previous research in word alignment, we present a system that automatically generates instances of word-translation relations based on self learning, with the goal of training a model to estimate translation probability for effective word alignment. We exploit the inherent crosslingual regularity in parallel corpora and use automatically annotated data for training a discriminative model.

3 The *TakeTwo* Aligner

Aligning words and translation using the EM algorithm based on generative IBM models is not effective for aligning low frequency words and nonliteral translations, especially across disparate languages. To align words and translations reliably in a given parallel corpus, a promising approach is to self-train a classifier with linguistics features, in order to impose additional requirements in combining alignments in two translation directions.

3.1 Problem Statement

We focus on producing word alignments, i.e., a set of word and translation links (word pairs), in each pair of sentences in a parallel corpus. The word alignment results can be used to estimate lexical and phrasal translation probabilities for machine translation; alternatively they can be helpful for bilingual lexicography and computer aided translation. Thus, it is crucial that we produce high-precision, broad coverage word alignments. We now formally state the problem that we are addressing.

Problem Statement: We are given a parallel corpus (E, F), and a monolingual corpus *MonoCorp*. The parallel corpus, (E, F), contains parallel sentences, (E_k, F_k), $k = 1, N$ where $E_k = e^k_0, e^k_1, ..., e^k_{n_k}$, and $F_k = f^k_0, f^k_1, ..., f^k_{m_k}$. Our goal is to produce a set of word alignments for each sentence pair (E_k, F_k). For this, we use an existing word aligner (e.g., *Giza++*) to produce two directional alignments and a symmetrized alignment:

$$\begin{gathered} \textit{E2F} = (\textit{E2F}_0, \textit{E2F}_1, .., \textit{E2F}_N) \\ \textit{F2E} = (\textit{F2E}_0, \textit{F2E}_1, .., \textit{F2E}_N) \\ \text{SYMM} = (\textit{SYMM}_0, \textit{SYMM}_1, .., \textit{SYMM}_N). \end{gathered}$$

Each alignment A of (E_k, F_k) in *E2F*, *F2E*, and *SYMM* is represented as

$$\{(i, j) | (e^k_i, f^k_j) \text{ is an alignment link in } A\ \}.$$

We then use a post-processing stage to improve on *SYMM* based on word-translation relation, predicted based on a discrimative model derived from *E2F*,

```
Procedure Train-X-SIM(E, F, MonoCorp):

Stage 1 (Section 3.2.1)

(1)  E2F, F2E, SYMM = WordAliger(E, F)
(2)  E2F-m, F2E-m, SYMM-m = WordAligner(E, F-morph)
(3)  POSITIVES, NEGATIVES = INTERSECT(E2F, F2E), UNION(E2F, F2E) - SYMM
(4)  Return TRAIN = POSITIVES + NEGATIVES

Stage 2 (Section 3.2.2)

(1)  Tag each sentence E(k) and F(k) with parts of speech
     For all English word e, foreign word f, and morpheme m of f
(2a)     Estimate LTP, P(e|f) based on F2E
(2b)     Estimate MTP, P(e|m) based on E2F-m
(3)  Build a  transliteration model  P_translit(e|f) based on an EF name list
(4)  Build a distributional similarity model Sim(e, e') based on MonoCorp
     For each link (e, f) in training data TRAIN, augment (e, f) with features
(5a)   f1 = max(e') P(e'|f) Sim(e', e),    f3 = P_translit(e|f),
(5b)   f2 = max(m, e') P(e'|m) Sim(e', e), f4 = (pos(e), pos(f))

Stage 3  (Section 3.2.3)

(1)   Return the classifier X-SIM trained on the feature vectors
```

Figure 3: Ouline of the process to train the *TakeTwo* system.

F2E, *SYMM*, *MonoCorp*, and other linguistic resources.

In the rest of this section, we describe our solution to this problem. We describe the self-learning strategy for training a classifier for predicting word-translation relation (Section 3.2). In this section, we also describe how to enrich the training data with linguistically motivated features. Finally, we show how *TakeTwo* aligns each sentence pairs by applying the trained classifier (Section 3.3).

3.2 Learning to Predict Cross-lingual Relatedness

We attempt to generate automatically annotated word-translation instances in (E, F) to train a classifier expected to predict word-translation relation. Our learning process is shown in Figure 3.

3.2.1 ***Generating Training Instances***. In the first learning stage, we use the initial word alignments to generate positive and negative instances for training a classifier that predicts alignment links via cross-lingual relatedness. Therefore, the output of this stage is a set of (*k*, *i*, *j*, *Pos* or *Neg*) tuples, where *Pos* or *Neg* denotes whether (e_i^k, f_j^k) is a valid alignment link in (E_k, F_k). To produce the output, we compute $TRAIN_k$:

$$\{\, (k, i, j\ Pos) \mid (i, j) \in E2F_k \cap F2E_k \,\} \cup$$
$$\{\, (k, i, j, Neg) \mid (i, j) \in E2F_k \cup F2E_k - SYMM_k \,\}.$$

Finally, we return ($TRAIN_0$, $TRAIN_1$, .., $TRAIN_N$) as output.

In Step (1) of the this stage, we generate two sets of word alignments (*E2F*, *F2E*) and symmetrized alignments *SYMM*. As will be described in Section 4, we used the existing tool *Giza*++ to generate these three sets of alignments.

To illustrate, we show in Figure 4 sample training instances, automatically generated for an example sentence pair. As can be seen in Figure 4, we produce six positive and three negative training instances. In this case, all nine instances are correctly labeled with *Pos* or *Neg*.

To assess the feasibility of the self learning approach, we have checked the annotated instances against hand-tagged links in a small dataset. We

Pos/Neg	i	j	English	Chinese	f_1	f_2	f_3	f_4
Pos	0	0	he	他	.9	.9	.0	PRP-Nh
Pos	4	6	after	後	.9	.9	.0	IN-Ng
Pos	5	3	heinonen	海諾寧	.0	.0	.7	NNP-Nb
Neg	5	9	heinonen	項	.0	.0	.0	NNP-Nf
Neg	5	10	heinonen	談話	.0	.0	.2	NNP-Na
Neg	3	3	remark	海諾寧	.0	.0	.3	NN-Nb
Pos	6	4	arrived	抵達	.9	.9	.0	VBD-VC
Pos	8	5	tehran	德黑蘭	.9	.9	.7	NNP-Nca
Pos	9	11	.	。	.0	.0	.0	.-。

Figure 4: Example positive and negative instances generated from bidirectional alignments of Ex (1). Each instance is augmented with features involving cross-lingual lexical relatedness (f_1), morphological relatedness (f_2), transliteration (f_3), and syntactic compatibility (f_4). In order to generate lexical and syntactic features, the sentences are tagged and lemmatized : *"He/PRP made/VBD this/DET remark/NN after/IN Heinonen/NNP arrived/VBD in Tehran/NNP ./.*", and "他/Nh 是/SHI 在/P 海諾寧/Nb 抵達/VC 德黑蘭/Nca 後/Ng 發表/VC 這/Nep 項/Nf 談話/Na 。/。").

found that around 90% of positive instances are correctly labelled, while around 95% of the negative instances are correctly labelled.

3.2.2 ***Generating features***. In the second stage of the learning process, we augment each training instance (k, i, j, *Pos*/*Neg*) generated in Section 3.2.1 with a set of features. For the sake of generality, we use a set of linguist features, involving lemmatized forms, morpholgical parts, distributional similarity, parts of speech, and transliteration model.

For this, in Step (1) of the second stage (see Figure 3), we perform tokenization and POS tagging on all sentences (E_k, F_k), k = 1, N. We tokenize F_k into words or Chinese characters, in order to perform word alignment on both word and morpheme levels. In Step (2), we estimate word translation probability and morpheme translation probability based on the initial alignment results, using both word-to-word and word-to-morpheme alignments. In Step (3), we estimate syllable-to-syllable transliteration probablity using a bilingual named entity list. In Step (4), we develop a distributional similarity model based on MonoCorp.

Finally, in Step (5), we use these models to generate a set of features for each training instance in TRAIN. The set of features we use include:

- **Cross-lingual lexical similarity.** This lexical feature is based on a simple idea: translating the foreign words f_j^k into English words e, and then measure similarity between the lemmas of e and e_i^k. Therefore, we have

$$feature_1 = \max_e P(e \mid f_j^k)\ sim\ (e, e_i^k).$$

- **Morpheme-based similarity feature.** This feature is similar to $feature_1$, but is estimated based on word part of a foreign word F_j^k aimed at handling compounds that might involves 1-to-many alignment (e.g., [*preserving water*, 節水 (jieshui)]). For this, we use the word-to-morphme and morpheme-to-word alignments to estimate lexical translation probability. Therefore, we have

$$feature_2 = \max_{e,\ m \in f_j^k} P(e \mid m) sim(e, e_i^k).$$

- **Transliteration feature.** The transliteration feature is designed to handle hard-to-align name entities appearing only once or twice in the whole corpus. Therefore, we we have

$$feature_3 = P_{translit}(f_j^k \mid e_j^k),$$

where $P_{translit}$ is a transliteration model trained on a list of bilingual named entities.

- **Syntactic feature.** We use parts of speech to capture cross-lingual regularity of words and translations on the syntactic level. For instance, an English preposition (i.e., IN) tends to align with a Chinese preposition or directional postposition (i.e., P or Ng). Therefore, we have

$$feature_4 = (pos(e_i^k), pos(f_j^k)),$$

where pos returns the part of speech of English word e_i^k or foreign word f_j^k in (E_k, F_k).

See Figure 4 for example training instances augmented with these crosslingual features.

3.2.3 ***Training classifier***. In the third and final stage of training, we train a classifier on a set of positive and negative feature vectors, generated in Section 3.2.2. The output of this stage is *X-Sim*, a classifier that provides probabilistic values indicating the likelihood of word-translation relation for (e_i^k, f_j^k) with features calculated in the context of (E_k, F_k).

3.3 Run-time Word Alignment

Once the classifier *X-Sim* is trained for predicting word-translation relation, *TakeTwo* then combine the two initial sets of alignments, using *X-Sim* to improve performance using the procedure shown in Figure 5. The alignment procedure is a modified version of GDFA procedure, with four steps: INTERSECT, GROW-DIAG-SIM, FILL-IN, and FINAL-AND. We use the same INTERSECT and FINAL-AND step, while modifying GROW-DIAG by requiring crosslingual similarity. The additional step of FILL-IN aimed at adding valid links missing from both *E2F*$_k$ and *F2E*$_k$.

In Step (1), we initalize SYMM/SIM to an empty set. In Steps (2) through (5), we combine the two alignments *E2F*$_k$ and *F2E*$_k$ for each sentence pair (E_k, F_k). And Finally, in Step (6) we output the new symmetrized alignment results.

In Step (2), we start with an alignment with the links in *E2Fk* ∩ *F2E*$_k$. In Step (3), we execute the GROW-DIAG-SIM step to add additional links neighboring the intersection links. A neighboring union link (*E2Fk* ∪ *F2E*$_k$), with high predicted probabiliy, are added to the results. In Step (4), we attempt to fill in links which are probably word-translation pairs, if the link is not in conflict with the current alignment. In Step (5), we execute the FINAL-AND step the same way as in GDFA.

In Step (6), we accumulate symmetrized alignment for a sentence pair. Finally, we add the symmetrized alignment to SYMM/SIM and return SYMM/SIM as output (in Step 7).

4 Experiments and Evaluation

We evaluate our alignment systems directly. We calculate recall, precision, and F-measure.

4.1 Experimental Setting

For self learning, we ran Giza++ on the FBIS corpus with 250 thousand parallel setnences (LDC-2003E14). The training scheme is as follows: 5 iterations of Model 1, followed by 5 iterations of HMM, followed by 5 iterations of Model 3 and then 5 iterations of Model 4. The systems evaluated include:

- *TakeTwo.*
- *TakeTwo (no fill-in).*
- *Giza++: grow-diag-final-and.*
- *Giza++: intersection.*
- *Giza++: union.*

We manually aligned 300 random selected sentences with English and Chinese words as the reference answers. For simplicity, we do not distinguished between sure and uncertain alignment links as described in (Och and Ney, 2004).

For preprocessing and generating syntactic features, we used the Genia Tagger and CKIP Word Segmenter to generate tokens and parts of speech. We also used the Wikipedia Dump (English) to build distributional word similarity measure.

In order to train a classifier for word-translation relation, we used SVM classifier with the tool libsvm. We used lexical, morphological, transliteration, and syntactic features, as described in Section 3.2.2. For simplicity, we used an empirically determined values for the thresholds of similarity constraint in $TakeTwo$.

4.2 Evaluation Metrics

Each word-translation link in the test sentences produced by a word aligner was judged to be either correct or incorrect in context. Precision was calculated as the fraction of correct pairs among the pair derived, recall was calculated as the fraction all correct pairs in the reference key, and the F-measure was

```
Procedure TakeTwo(E2F, F2E, Classifier)
(1)   SYMM/SIM = empty set of word alignments

      For each word alignments, E2F(k), and F2E(k), SYMM(k)
(2)     alignment = INTERSECT(E2F(k), F2E(k))
(3)     GROW-DIAG/SIM(alignment, E2F(k), F2E(k))
(4)     FILL(alignment, E2F(k), F2E(k))
(5)     FINAL-AND(alignment, E2F(k), F2E(k))
(6)     Add alignment to SYMM/SIM

(7)   Return SYMM/SIM

neighboring = [(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)]

GROW-DIAG/RF(Alignment):
  Iterate until no new points added
    For English word e = 0 ... en, foreign word f = 0 ... fm
        If ( e aligned with f )
          For each neighboring point ( e-new, f-new ):
            If ( ( e-new not aligned or f-new not aligned ) and
                 ( e-new, f-new ) in union( E2F(k), F2E(k) ) and
                 ( X-SIM ( e-new, f-new ) > threshold ) )
              Add to Alignment the link ( e-new, f-new )

FILL(alignment):
  Alignment_candidates = []
  For english word e-new = 0 ... en, foreign word f-new = 0 ... fn
    If ( ( e-new not aligned and f-new not aligned ) and
         ( X-SIM ( e-new, f-new ) > threshold ) )
      Add to Alignment_candidates the link ( e-new, f-new )
  Sort Alignment_candidates by decreasing X-SIM values
  For link (e-new, f-new) in Alignment_candidates
    If ( e-new not aligned and f-new not aligned )
      Add to Alignment the link ( e-new, f-new )

FINAL-AND(Alignment):
  For English word e-new = 0 ... en, foreign word f-new = 0 ... fn
    If ( ( e-new not aligned and f-new not aligned ) and
         ( e-new, f-new ) in alignment )
      Add to Alignment the link ( e-new, f-new )
```

Figure 5: Aligning word and translation at run-time.

calculated with equal weights for both precision and recall.

4.3 Experimental Results

In this section, we report the results of the experimental evaluation. Table 1 lists the precision, recall, and F-measure of two $TakeTwo$ variant systems, and the $Giza$++ derived systems. All six systems were tested and evaluated over the test set of 300 parallel sentences sampled from FBIS.

In summary, the $TakeTwo$ with the FILL-IN step has the highest F-measure, while $TakeTwo$ without the FILL-IN step has the second highest F-measure, followed by *GIZA*++ with GDFA symmetrization. Both $TakeTwo$ systems outperform the state of the art systems and gains of 6% and 3% in F-measure, with higher precision rate (+16% and +9%) with small descreases in recall rate (-3% and -1%). These results indicate that relevance feedback combined with a rich set of linguistic features are very effective in improving word alginment accuracy in a post-processing setting.

5 Conclusion and Future work

We have presented a new method for word alignment. In our work, we use self learning to generate training data for classifying word-translation relation, based on a rich set of features. The classifier is used in the second word alignment round to val-

Systems	P	R	F
TakeTwo	**.75**	**.65**	**.70**
TakeTwo w/o FILL-IN	.68	.67	.67
grow-diag-final-and (GDFA)	.59	.68	.64
intersection	.88	.46	.60
union	.47	.75	.58

Table 1: Word alignment performance of six systems compared measured by average precision rate (P), recall rate (R), and F-measure (M).

idate links in inital alignment round 'and to fill in missing links. Preliminary experiments and evaluations show our method is capable of aligning words and translations with high precision.

Many avenues exist for future research and improvement of our system. For example, Bleu score of SMT systems using the word alignment results could be used to evaluate the effectiveness of word alignment. Phrasal translations in the bilingual lexicon could be used to make many-to-many alignment decisions. In addition, natural language processing techniques such as word clustering, and cross-lingual relatedness could be attempted to improve recall. Another interesting direction to explore is training an ensemble of classifiers. Yet another direction of research would be to align word from scratch using the classifier in a beam-search algorithm.

References

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

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

Joseph Z Chang, Jason S Chang, and Jyh-Shing Roger Jang. 2012. Learning to find translations and transliterations on the web. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 130–134. Association for Computational Linguistics.

Colin Cherry and Dekang Lin. 2003. A probability model to improve word alignment. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 88–95. Association for Computational Linguistics.

Colin Cherry and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 105–112. Association for Computational Linguistics.

Yonggang Deng and Bowen Zhou. 2009. Optimizing word alignment combination for phrase table training. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 229–232. Association for Computational Linguistics.

Bonnie Jean Dorr. 1993. *Machine translation: a view from the Lexicon*. MIT press.

Christiane Fellbaum. 1999. *WordNet*. Wiley Online Library.

William A Gale and Kenneth Ward Church. 1991. Identifying word correspondences in parallel texts. In *HLT*, volume 91, pages 152–157.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 80–87. Association for Computational Linguistics.

Sue J Ker and Jason S Chang. 1997. A class-based approach to word alignment. *Computational Linguistics*, 23(2):313–343.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Dekang Lin, Shaojun Zhao, Benjamin Van Durme, and Marius Pasca. 2008. Mining parenthetical translations from the web by word alignment. In *ACL*, volume 8, pages 994–1002.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.

Dekang Lin. 1998b. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

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

Adam Lopez. 2007. A survey of statistical machine translation. Technical report, DTIC Document.

Lukas Michelbacher, Florian Laws, Beate Dorow, Ulrich Heid, and Hinrich Schütze. 2010. Building a cross-lingual relatedness thesaurus using a graph similarity measure. In *LREC*.

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

Masaaki Nagata, Teruka Saito, and Kenji Suzuki. 2001. Using the web as a bilingual dictionary. In *Proceedings of the workshop on Data-driven methods in machine translation-Volume 14*, pages 1–8. Association for Computational Linguistics.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 301–305. 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.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

Adam Pauls, Dan Klein, David Chiang, and Kevin Knight. 2010. Unsupervised syntactic alignment with inversion transduction grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 118–126. Association for Computational Linguistics.

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*, pages 534–544. Association for Computational Linguistics.

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

Zhiguo Wang and Chengqing Zong. 2013. Large-scale word alignment using soft dependency cohesion constraints. *Transactions of Association for Computational Linguistics*, 1(6):291–300.

Jian-Cheng Wu and Jason S Chang. 2007. Learning to find english to chinese transliterations on the web. In *EMNLP-CoNLL*, pages 996–1004.