Improved Discriminative ITG Alignment using Hierarchical Phrase Pairs and Semi-supervised Training

[†]Shujie Liu[*], [‡]Chi-Ho Li and [‡]Ming Zhou
[†] School of Computer Science and Technology
Harbin Institute of Technology
shujieliu@mtlab.hit.edu.cn
[‡]Microsoft Research Asia
{chl, mingzhou}@microsoft.com

Abstract

While ITG has many desirable properties for word alignment, it still suffers from the limitation of one-to-one matching. While existing approaches relax this limitation using phrase pairs, we propose a ITG formalism, which even handles units of non-contiguous words, using both simple and hierarchical phrase pairs. We also propose a parameter estimation method, which combines the merits of both supervised and unsupervised learning, for the ITG formalism. The ITG alignment system achieves significant improvement in both word alignment quality and translation performance.

1 Introduction

Inversion transduction grammar (ITG) (Wu, 1997) is an adaptation of CFG to bilingual parsing. It does synchronous parsing of two languages with phrasal and word-level alignment as by-product. One of the merits of ITG is that it is less biased towards short-distance reordering compared with other word alignment models such as HMM. For this reason ITG has gained more and more attention recently in the word alignment community (Zhang et al., 2005; Cherry et al., 2006; Haghighi et al., 2009) .

The basic ITG formalism suffers from the major drawback of one-to-one matching. This limitation renders ITG unable to produce certain alignment patterns (such as many-to-many alignment for idiomatic expression). For this reason those recent approaches to ITG alignment introduce the notion of *phrase* (or *block*), defined as sequence of contiguous words, into the ITG formalism (Cherry and Lin, 2007; Haghighi et al., 2009; Zhang et al., 2008). However, there are still alignment patterns which cannot be captured by phrases. A simple example is connective in Chinese/English. In English, two clauses are connected by merely one connective (like "*although*", "*because*") but in Chinese we need two connectives (e.g. There is a sentence pattern "虽然 X_1 但是 $X_2 \rightarrow X_2$ although X_1", where X_1 and X_2 are variables for clauses). The English connective should then be aligned to two non-contiguous Chinese connectives, and such alignment pattern is not available in either word-level or phrase-level ITG. As hierarchical phrase-based SMT (Chiang, 2007) is proved to be superior to simple phrase-based SMT, it is natural to ask, why don't we further incorporate hierarchical phrase pairs (henceforth h-phrase pairs) into ITG? In this paper we propose a ITG formalism and parsing algorithm using h-phrase pairs.

The ITG model involves much more parameters. On the one hand, each phrase/h-phrase pair has its own probability or score. It is not feasible to learn these parameters through discriminative/supervised learning since the repertoire of phrase pairs is much larger than the size of human-annotated alignment set. On the other hand, there are also a few useful features which cannot be estimated merely by unsupervised learning like EM. Inspired by Fraser et al. (2006), we propose a semi-supervised learning algorithm which combines the merits of both discrimina-

[*] This work has been done while the first author was visiting Microsoft Research Asia.

Coling 2010: Poster Volume, pages 730–738,
Beijing, August 2010

tive training (error minimization) and approximate EM (estimation of numerous parameters).

The ITG model augmented with the learning algorithm is shown by experiment results to improve significantly both alignment quality and translation performance.

In the following, we will explain, step-by-step, how to incorporate hierarchical phrase pairs into the ITG formalism (Section 2) and in ITG parsing (Section 3). The semi-supervised training method is elaborated in Section 4. The merits of the complete system are illustrated with the experiments described in Section 5.

2 ITG Formalisms

2.1 W-ITG : ITG with only word pairs

The simplest formulation of ITG contains three types of rules: terminal unary rules $X \rightarrow e/f$, where e and f represent words (possibly a null word, ε) in the English and foreign language respectively, and the binary rules $X \rightarrow [X, X]$ and $X \rightarrow \langle X, X \rangle$, which refer to that the component English and foreign phrases are combined in the same and inverted order respectively. From the viewpoint of word alignment, the terminal unary rules provide the links of word pairs, whereas the binary rules represent the reordering factor. Note also that the alignment between two phrase pairs is always composed of the alignment between word pairs (c.f. Figure 1(a) and (b)). The Figure 1 also shows ITG can handle the cases where two languages share the same (Figure 1(a)) and different (Figure 1(b)) word order

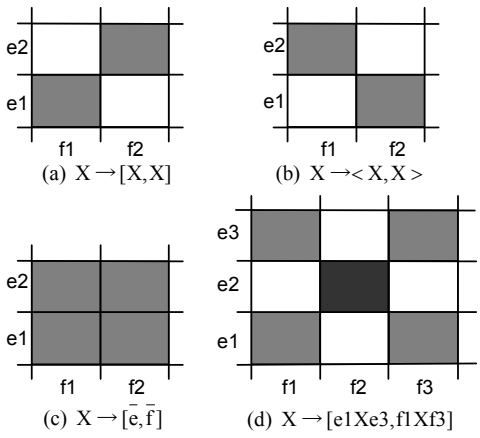

Figure 1. Four ways in which ITG can analyze a multi-word span pair.

Such a formulation has two drawbacks. First of all, the simple ITG leads to redundancy if word alignment is the sole purpose of applying ITG. For instance, there are two parses for three consecutive word pairs, viz. [a/a' [b/b' c/c']] and [[a/a' b/b'] c/c']. The problem of redundancy is fixed by adopting ITG normal form. The ITG normal form grammar as used in this paper is described in Appendix A.

The second drawback is that ITG fails to produce certain alignment patterns. Its constraint that a word is not allowed to align to more than one word is indeed a strong limitation as no idiom or multi-word expression is allowed to align to a single word on the other side. Moreover, its reordering constraint makes it unable to produce the 'inside-out' alignment pattern (c.f. Figure 2).

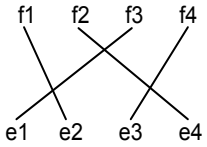

Figure 2. An example of inside-out alignment.

2.2 P-ITG : ITG with Phrase Pairs

A single word in one language is not always on a par with a single word in another language. For example, the Chinese word "白宫" is equivalent to two words in English ("white house"). This problem is even worsened by segmentation errors (i.e. splitting a single word into more than one word). The one-to-one constraint in W-ITG is a serious limitation as in reality there are always segmentation or tokenization errors as well as idiomatic expressions. Therefore, researches like Cherry and Lin (2007), Haghighi et al. (2009) and Zhang et al. (2009) tackle this problem by enriching ITG, in addition to word pairs, with pairs of phrases (or blocks). That is, a sequence of source language word can be aligned, as a whole, to one (or a sequence of more than one) target language word.

These methods can be subsumed under the term phrase-based ITG (P-ITG), which enhances W-ITG by altering the definition of a terminal production to include phrases: $X \rightarrow \bar{e}/\bar{f}$ (c.f. Figure 1(c)). $\bar{e}$ stands for English phrase and $\bar{f}$ stands for foreign phrase. As an example, if there is a simple phrase pair <white house, 白

宫>, then it is transformed into the ITG rule $C \rightarrow$"white house"/"白宫".

An important question is how these phrase pairs can be formulated. Marcu and Wong (2002) propose a joint probability model which searches the phrase alignment space, simultaneously learning translations lexicons for words and phrases without consideration of potentially sub-optimal word alignments and heuristic for phrase extraction. This method suffers from computational complexity because it considers all possible phrases and all their possible alignments. Birch et al. (2006) propose a better and more efficient method of constraining the search space which does not contradict a given high confidence word alignment for each sentence. Our P-ITG collects all phrase pairs which are consistent with a word alignment matrix produced by a simpler word alignment model.

2.3 HP-ITG : P-ITG with H-Phrase pairs

P-ITG is the first enhancement of ITG to capture the linguistic phenomenon that more than one word of a language may function as a single unit, so that these words should be aligned to a single unit of another language. But P-ITG can only treat contiguous words as a single unit, and therefore cannot handle the single units of non-contiguous words. Apart from sentence connectives as mentioned in Section 1, there is also the example that the single word "since" in English corresponds to two non-adjacent words "自" and "以来" as shown the following sentence pair:

自 上周末 ***以来***， 我 一直 在 生病 .

I have been ill ***since*** last weekend .

No matter whether it is P-ITG or phrase-based SMT, the very notion of phrase pair is not helpful because this example is simply handled by enumerating all possible contiguous sequences involving the words "自" and "以来", and thus subject to serious data sparseness. The lesson learned from hierarchical phrase-based SMT is that the modeling of non-contiguous word sequence can be very simple if we allow rules involving h-phrase pairs, like:

$C \rightarrow$"since X"/自 X 以来"

where X is a placeholder for substituting a phrase pair like "上周末/last weekend".

H-phrase pairs can also perform reordering, as illustrated by the well-known example from Chiang (2007), $C \rightarrow$ "have X_2 with X_1" / "与 X_1 有 X_2", for the following bilingual sentence fragment:

与 北韩 ***有*** 邦交

have diplomatic relations ***with*** North Korea

The potential of intra-phrase reordering may also help us to capture those alignment patterns like the 'inside-out' pattern.

All these merits of h-phrase pairs motivate a ITG formalism, viz. hierarchical phrase-based ITG (HP-ITG), which employs not only simple phrase pairs but also hierarchical ones. The ITG grammar is enriched with rules of the format: $X \rightarrow \bar{e}/\bar{f}$ where $\bar{e}$ and $\bar{f}$ refer to either a phrase or h-phrase (c.f. Figure 1(d)) pair in English and foreign language respectively[2]. Note that, although the format of HP-ITG is similar to P-ITG, it is much more difficult to handle rules with h-phrase pairs in ITG parsing, which will be elaborated in the next section.

It is again an important question how to formulate the h-phrase pairs. Similar to P-ITG, the h-phrase pairs are obtained by extracting the h-phrase pairs which are consistent with a word alignment matrix produced by some simpler word alignment model.

3 ITG Parsing

Based on the rules, W-ITG word alignment is done in a similar way to chart parsing (Wu, 1997). The base step applies all relevant terminal unary rules to establish the links of word pairs. The word pairs are then combined into span pairs in all possible ways. Larger and larger span pairs are recursively built until the sentence pair is built.

Figure 3(a) shows one possible derivation for a toy example sentence pair with three words in each sentence. Each node (rectangle) represents a pair, marked with certain phrase category, of

[2] Haghighi et al. (2009) impose some rules which look like h-phrase pairs, but their rules are essentially h-phrase pairs with at most one 'X' only, added with the constraint that each 'X' covers only one word.

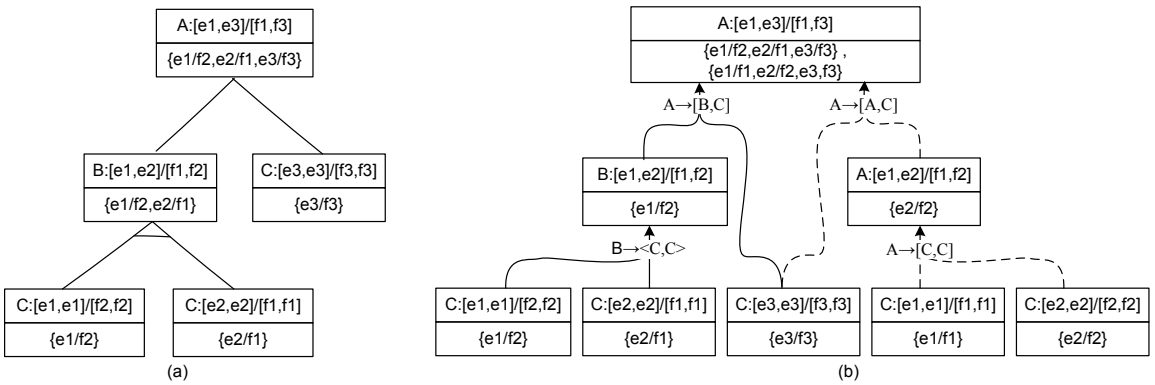

Figure 3. Example ITG parses in graph (a) and hypergraph (b).

foreign span (F-span) and English span (E-span) (the upper half of the rectangle) and the associated alignment hypothesis (the lower half). Each graph like Figure 3(a) shows only one derivation and also only one alignment hypothesis.

The various derivations in ITG parsing can be compactly represented in hypergraph (Klein et al., 2001) like Figure 3(b). Each hypernode (rectangle) comprises both a span pair (upper half) and the list of possible alignment hypotheses (lower half) for that span pair. The hyperedges show how larger span pairs are derived from smaller span pairs. Note that hypernode may have more than one alignment hypothesis, since a hypernode may be derived through more than one hyperedge (e.g. the topmost hypernode in Figure 3(b)). Due to the use of normal form, the hypotheses of a span pair are different from each other.

In the case of P-ITG parsing, each span pair does not only examine all possible combinations of sub-span pairs using binary rules, but also checks if the yield of that span pair is exactly the same as that phrase pair. If so, then this span pair is treated as a valid leaf node in the parse tree. Moreover, in order to enable the parse tree produce a complete word aligned matrix as by-product, the alignment links within the phrase pair (which are recorded when the phrase pair is extracted from a word aligned matrix produced by a simpler model) are taken as an alternative alignment hypothesis of that span pair.

In the case of HP-ITG parsing, an ITG rule like $C \rightarrow$ "have X_2 with X_1"/"与 X_1有 X_2" (originated from the hierarchical rule like $X \rightarrow$ <与 X_1 有 X_2, have X_2 with X_1>), is processed in the following manner: 1) Each span pair checks if it contains the lexical anchors: "have", "with","与" and "有"; 2) each span pair checks if the remaining words in its yield can form two sub-span pairs which fit the reordering constraint among X_1 and X_2 (Note that span pairs of any category in the ITG normal form grammar can substitute for X_1 or X_2). 3) If both conditions hold, then the span pair is assigned an alignment hypothesis which combines the alignment links among the lexical anchors and those links among the sub-span pairs.

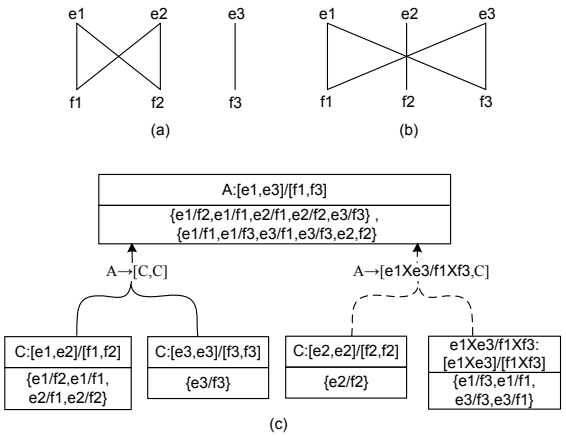

Figure 4. Phrase/h-phrase in hypergraph.

Figure 4(c) shows an example how to use phrase pair and h-phrase pairs in hypergraph. Figure 4(a) and Figure 4(b) refer to alignment matrixes which cannot be generated by W-ITG, because of the one-to-one assumption. Figure 4(c) shows how the span pair [e1,e3]/[f1,f3] can be generated in two ways: one is combining a phrase pair and a word pair directly, and the other way is replacing the *X* in the h-phrase pair with a word pair. Here we only show how h-phrase pairs with one variable be used during the

parsing, and h-phrase pairs with more than one variable can be used in a similar way.

The original (unsupervised) ITG algorithm has complexity of $O(n^6)$. When extended to supervised/discriminative framework, ITG runs even more slowly. Therefore all attempts to ITG alignment come with some pruning method. Zhang and Gildea (2005) show that Model 1 (Brown et al., 1993) probabilities of the word pairs inside and outside a span pair are useful. Tic-tac-toe pruning algorithm (Zhang and Gildea, 2005) uses dynamic programming to compute inside and outside scores for a span pair in $O(n^4)$. Tic-tac-toe pruning method is adopted in this paper.

4 Semi-supervised Training

The original formulation of ITG (W-ITG) is a generative model in which the ITG tree of a sentence pair is produced by a set of rules. The parameters of these rules are trained by EM. Certainly it is difficult to add more non-independent features in such a generative model, and therefore Cherry et al. (2006) and Haghighi et al. (2009) used a discriminative model to incorporate features to achieve state-of-art alignment performance.

4.1 HP-DITG : Discriminative HP-ITG

We also use a discriminative model to assign score to an alignment candidate for a sentence pair $(\bar{f}, \bar{e})$ as probability from a log-linear model (Liu et al., 2005; Moore, 2006):

$$P(a|\bar{e},\bar{f}) = \frac{exp(\sum_i \lambda_i \Psi_i(a,\bar{f},\bar{e}))}{\sum_{a' \in A} exp(\sum_i \lambda_i \Psi_i(a',\bar{f},\bar{e}))} \quad (1)$$

where each $\Psi_i(a,\bar{f},\bar{e})$ is some feature about the alignment matrix, and each λ is the weight of the corresponding feature. The discriminative version of W-ITG, P-ITG, and HP-ITG are then called W-DITG, P-DITG, and HP-DITG respectively.

There are two kinds of parameters in (1) to be learned. The first is the values of the features Ψ. Most features are indeed about the probabilities of the phrase/h-phrase pairs and there are too many of them to be trained from a labeled data set of limited size. Thus the feature values are trained by approximate EM. The other kind of parameters is feature weights λ, which are trained by an error minimization method. The discriminative training of λ and the approximate EM training of Ψ are integrated into a semi-supervised training framework similar to EMD3 (Fraser and Marcu, 2006).

4.2 Discriminative Training of λ

MERT (Och, 2003) is used to train feature weights λ. MERT estimates model parameters with the objective of minimizing certain measure of translation errors (or maximizing certain performance measure of translation quality) for a development corpus. Given an SMT system which produces, with model parameters λ_1^M, the K-best candidate translations $\hat{e}(f_s;\lambda_1^M)$ for a source sentence f_s, and an error measure $E(r_s, e_{s,k})$ of a particular candidate $e_{s,k}$ with respect to the reference translation r_s, the optimal parameter values will be:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{argmin}\left\{\sum_{s=1}^{S} E(r_s, \hat{e}(f_s;\lambda_1^M))\right\}$$
$$= \underset{\lambda_1^M}{argmin}\left\{\sum_{s=1}^{S}\sum_{k=1}^{K} E(r_s, e_{s,k})\delta(\hat{e}(f_s;\lambda_1^M), e_{s,k})\right\}$$

MERT for DITG applies the same equation for parameter tuning, with different interpretation of the components in the equation. Instead of a development corpus with reference translations, we have a collection of training samples, each of which is a sentence pair with annotated alignment result. The ITG parser outputs for each sentence pair a K-best list of alignment result $\hat{e}(f_s;\lambda_1^M)$ based on the current parameter values λ_1^M. The MERT module for DITG takes alignment F-score of a sentence pair as the performance measure. Given an input sentence pair and the reference annotated alignment, MERT aims to maximize the F-score of DITG-produced alignment.

4.3 Approximate EM Training of Ψ

Three kinds of features (introduced in section 4.5 and 4.6) are calculated from training corpus given some initial alignment result: conditional probability of word pairs and two types of conditional probabilities for phrase/h-phrase.

[3] For simplicity, we will also call our semi-supervised framework as EMD.

The initial alignment result is far from perfect and so the feature values thus obtained are not optimized. There are too many features to be trained in supervised way. So, unsupervised training like EM is the best solution.

When EM is applied to our model, the E-step corresponds to calculating the probability for all the ITG trees, and the M-step corresponds to re-estimate the feature values. As it is intractable to handle all possible ITG trees, instead we use the Viterbi parse to update the feature values. In other words, the training is a kind of approximate EM rather than EM.

Word pairs are collected over Viterbi alignment and their conditional probabilities are estimated by MLE. As to phrase/h-phrase, if they are handled in a similar way, then there will be data sparseness (as there are much fewer phrase/h-phrase pairs in Viterbi parse tree than needed for reliable parameter estimation). Thus, we collect all phrase/h-phrase pairs which are consistent with the alignment links. The conditional probabilities are then estimated by MLE.

4.4 Semi-supervised training

Algorithm EMD (semi-supervised training)

input development data *dev*, test data *test*, training data with initial alignment (*train, align_train*)

output feature weights λ and features Ψ.

estimate initial features Ψ^0 with (*train, align_train*)
get an initial weights λ^0 by MERT with the initial features Ψ^0 on *dev*.
get the F-Measure e^0 for λ^0 and Ψ^0 on *test*.
for(t=1;; t++)
    get the Viterbi alignment *align_train* for *train* using λ^{t-1} and Ψ^{t-1}
    estimate Ψ^t with (*train, align_train*)
    get new feature weights λ^t by MERT with Ψ^t on *dev*.
    get the F-Measure e^t for λ^t and Ψ^t on *test*.
    if $e^t \leq e^{t-1} + 0.1$ **then**
        break
end for
return λ^{t-1} and Ψ^{t-1}

Figure 5. Semi-supervised training for HP-DITG.

The discriminative training (error minimization) of feature weights λ and the approximate EM learning of feature values Ψ are integrated in a single semi-supervised framework. Given an initial estimation of Ψ (estimated from an initial alignment matrix by some simpler word alignment model) and an initial estimation of λ, the discriminative training process and the approximate EM learning process are alternatively iterated until there is no more improvement. The sketch of the semi-supervised training is shown in Figure 5.

4.5 Features for word pairs

The following features about alignment link are used in W-DITG:

1) Word pair translation probabilities trained from HMM model (Vogel et al., 1996) and IBM model 4 (Brown et al., 1993).
2) Conditional link probability (Moore, 2006).
3) Association score rank features (Moore et al., 2006).
4) Distortion features: counts of inversion and concatenation.

4.6 Features for phrase/h-phrase pairs

For our HP-DITG model, the rule probabilities in both English-to-foreign and foreign-to-English directions are estimated and taken as features, in addition to those features in W-DITG, in the discriminative model of alignment hypothesis selection:

1) $P(\bar{e}_i/\bar{f}_i)$: The conditional probability of English phrase/h-phrase given foreign phrase/h-phrase.
2) $P(\bar{f}_i/\bar{e}_i)$: The conditional probability of foreign phrase/h-phrase given English phrase/h-phrase.

The features are calculated as described in section 4.3.

5 Evaluation

Our experiments evaluate the performance of HP-DITG in both word alignment and translation in a Chinese-English setting, taking GIZA++, BerkeleyAligner (henceforth BERK) (Haghighi, et al., 2009), W-ITG as baselines. Word alignment quality is evaluated by recall, precision, and F-measure, while translation performance is evaluated by case-insensitive BLEU4.

5.1 Experiment Data

The small human annotated alignment set for discriminative training of feature weights is the same as that in Haghighi et al. (2009). The 491

sentence pairs in this dataset are adapted to our own Chinese word segmentation standard. 250 sentence pairs are used as training data and the other 241 are test data. The large, un-annotated bilingual corpus for approximate EM learning of feature values is FBIS, which is also the training set for our SMT systems.

In SMT experiments, our 5-gram language model is trained from the Xinhua section of the Gigaword corpus. The NIST'03 test set is used as our development corpus and the NIST'05 and NIST'08 test sets are our test sets. We use two kinds of state-of-the-art SMT systems. One is a phrase-based decoder (PBSMT) with a MaxEnt-based distortion model (Xiong, et al., 2006), and the other is an implementation of hierarchical phrase-based model (HPBSMT) (Chiang, 2007). The phrase/rule table for these two systems is not generated from the terminal node of HP-DITG tree directly, but extracted from word alignment matrix (HP-DITG generated) using the same criterion as most phrase-based systems (Chiang, 2007).

5.2 HP-DITG without EMD

Our first experiment isolates the contribution of the various DITG alignment models from that of semi-supervised training. The feature values of the DITG models are estimated simply from IBM Model 4 using GIZA++. Apart from DITG, P-ITG, and HP-ITG as introduced in Section 2, we also include a variation, known as H-DITG, which covers h-phrase pairs but no simple phrase pairs at all. The experiment results are shown in Table 1.

	Precision	Recall	F-Measure
GIZA++	0.826	0.807	0.816
BERK	**0.917**	**0.814**	**0.862**
W-DITG	0.912	0.745	0.820
P-DITG	0.913	0.788	0.846
H-DITG	0.913	0.781	0.842
HP-DITG	**0.915**	**0.795**	**0.851**

Table 1. Performance gains with features for HP-DITG.

It is obvious that any form of ITG achieves better F-Measure than GIZA++. Without semi-supervised training, however, our various DITG models cannot compete with BERK. Among the DITG models, it can be seen that precision is roughly the same in all cases, while W-ITG has the lowest recall, due to the limitation of one-to-one matching. The improvement by (simple) phrase pairs is roughly the same as that by h-phrase pairs. And it is not surprising that the combination of both kinds of phrases achieve the best result.

Even HP-DITG does not achieve as high recall as BERK, it does produce promising alignment patterns that BERK fails to produce. For instance, for the following sentence pair:

自 上周末 以来， 我 一直 在 生病 .

I have been ill ***since*** last weekend .

Both GIZA++ and BERK produce the pattern in Figure 6(a), while HP-DITG produces the better pattern in Figure 6(b) as it learns the h-phrase pair "since *X*"/"自 *X* 以来".

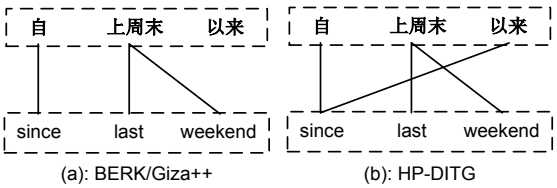

Figure 6. Partial alignment results.

5.3 Alignment Quality of HP-DITG with EMD

	Precision	Recall	F- Measure
GIZA++	0.826	0.807	0.816
BERK	0.917	0.814	0.862
EMD0	0.915	0.795	0.851
EMD1	0.923	0.814	0.865
EMD2	0.930	**0.821**	0.872
EMD3	**0.935**	0.819	**0.873**

Table 2. Semi-supervised Training Task on F-Measure.

The second experiment evaluates how the semi-supervised method of EMD improves HP-DITG with respect to word alignment quality. The results are shown in Table 2. In the table, EMD0 refers to the HP-DITG model before any EMD training; EMD1 refers to the model after the first iteration of training, and so on. It is empirically found that F-Measure is not improved after the third EMD iteration.

It can be observed that EMD succeeds to help HP-DITG improves feature value and weight estimation iteratively. When semi-supervised

training converges, the new HP-DITG model is better than before training by 2%, and better than BERK by 1%.

5.4 Translation Quality of HP-DITG with EMD

The third experiment evaluates the same alignment models in the last experiment but with respect to translation quality, measured by case-insensitive BLEU4. The results are shown in Table 3. Note that the differences between EMD3 and the two baselines are statistically significant.

	PBSMT		HPBSMT	
	05	08	05	08
GIZA++	33.43	23.89	33.59	24.39
BERK	33.76	24.92	34.22	25.18
EMD0	34.02	24.50	34.30	24.90
EMD1	34.29	24.80	34.77	25.25
EMD2	*34.25*	25.01	**35.04**	25.43
EMD3	**34.42**	**25.19**	*34.82*	**25.56**

Table 3. Semi-supervised Training Task on BLEU.

It can be observed that EMD improves SMT performance in most iterations in most cases. EMD does not always improve BLEU score because the objective function of the discriminative training in EMD is about alignment F-Measure rather than BLEU. And it is well known that the correlation between F-Measure and BLEU (Fraser and Marcu, 2007) is itself an intriguing problem.

The best HP-DITG leads to more than 1 BLEU point gain compared with GIZA++ on all datasets/MT models. Compared with BERK, EMD3 improves SMT performance significantly on NIST05 and slightly on NIST08.

6 Conclusion and Future Work

In this paper, we propose an ITG formalism which employs the notion of phrase/h-phrase pairs, in order to remove the limitation of one-to-one matching. The formalism is proved to enable an alignment model to capture the linguistic fact that a single concept is expressed in several non-contiguous words. Based on the formalism, we also propose a semi-supervised training method to optimize feature values and feature weights, which does not only improve the alignment quality but also machine translation performance significantly. Combining the formalism and semi-supervised training, we obtain better alignment and translation than the baselines of GIZA++ and BERK.

A fundamental problem of our current framework is that we fail to obtain monotonic increment of BLEU score during the course of semi-supervised training. In the future, therefore, we will try to take the BLEU score as our objective function in discriminative training. That is to certain extent inspired by Deng et al. (2008).

Appendix A. The Normal Form Grammar

Table 4 lists the ITG rules in normal form as used in this paper, which extend the normal form in Wu (1997) so as to handle the case of alignment to null.

1	$S \rightarrow A \mid B \mid C$
2	$A \rightarrow [A\ B] \mid [A\ C] \mid [B\ B] \mid [BC] \mid [C\ B] \mid [C\ C]$
3	$B \rightarrow \langle A\ A\rangle \mid \langle A\ C\rangle \mid \langle B\ A\rangle \mid \langle B\ C\rangle$
	$B \rightarrow \langle C\ A\rangle \mid \langle C\ C\rangle$
4	$C \rightarrow C_w \mid C_{fw} \mid C_{ew}$
5	$C \rightarrow [C_{ew}\ C_{fw}]$
6	$C_w \rightarrow u/v$
7	$C_e \rightarrow \varepsilon/v;\ C_f \rightarrow u/\varepsilon$
8	$C_{em} \rightarrow C_e \mid [C_{em}\ C_e];\ C_{fm} \rightarrow C_f \mid [C_{fm}\ C_f]$
9	$C_{ew} \rightarrow [C_{em}\ C_w];\ C_{fw} \rightarrow [C_{fm}\ C_w]$

Table 4. ITG Rules in Normal Form.

In these rules, S is the Start symbol; A is the category for concatenating combination whereas B for inverted combination. Rules (2) and (3) are inherited from Wu (1997). Rules (4) divide the terminal category C into subcategories. Rule schema (6) subsumes all terminal unary rules for some English word u and foreign word v, and rule schemas (7) are unary rules for alignment to null. Rules (8) ensure all words linked to null are combined in left branching manner, while rules (9) ensure those words linked to null combine with some following, rather than preceding, word pair. (Note: Accordingly, all sentences must be ended by a special token $\langle end\rangle$, otherwise the last word(s) of a sentence cannot be linked to null.) If there are both English and foreign words linked to null, rule (5) ensures that those English words linked to null precede those foreign words linked to null.

References

Birch, Alexandra, Chris Callison-Burch, Miles Osborne and Phillipp Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. *Proceedings of the Workshop on Statistical Machine Translation*.

Brown, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Peitra, Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics,* 19(2):263-311.

Cherry, Colin and Dekang Lin. 2006. Soft Syntactic Constraints for Word Alignment through Discriminative Training. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics.

Cherry, Colin and Dekang Lin. 2007. Inversion Transduction Grammar for Joint Phrasal Translation Modeling. *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, Pages:17-24.

Chiang, David. 2007. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2).

Deng, Yonggang, Jia Xu and Yuqing Gao. 2008. Phrase Table Training For Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair?. *Proceedings of the 7th International Conference on Human Language Technology Research and 46th Annual Meeting of the Association for Computational Linguistics*, Pages:1017-1026.

Fraser, Alexander, Daniel Marcu. 2006. Semi-Supervised Training for StatisticalWord Alignment. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Pages:769-776.

Fraser, Alexander, Daniel Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3).

Haghighi, Aria, John Blitzer, John DeNero, and Dan Klein. 2009. Better Word Alignments with Supervised ITG Models. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language, Pages: 923-931.

Klein, Dan and Christopher D. Manning. 2001. Parsing and Hypergraphs. *Proceedings of the 7th International Workshop on Parsing Technologies*, Pages:17-19

Liu, Yang, Qun Liu and Shouxun Lin. 2005. Log-linear models for word alignment. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Pages: 81-88.

Marcu, Daniel, William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing*, Pages:133-139.

Moore, Robert, Wen-tau Yih, and Andreas Bode. 2006. Improved Discriminative Bilingual Word Alignment. *Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics*, Pages: 513-520.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. *Proceedings of the 41rd Annual Meeting of the Association for Computational Linguistics*, Pages:160-167.

Och, Franz Josef and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4) : 417-449.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. *Proceedings of 16th International Conference on Computational Linguistics*, Pages: 836-841.

Wu, Dekai. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3).

Xiong, Deyi, Qun Liu and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. *Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics*, Pages: 521-528.

Zhang, Hao and Daniel Gildea. 2005. Stochastic Lexicalized Inversion Transduction Grammar for Alignment. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.

Zhang, Hao, Chris Quirk, Robert Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. *Proceedings of the 46rd Annual Meeting of the Association for Computational Linguistics*, Pages: 314-323.