Improving Phrase-Based Translation via Word Alignments from Stochastic Inversion Transduction Grammars

Markus SAERS

Dept. of Linguistics and Philology
Uppsala University
Sweden
markus.saers@lingfil.uu.se

Dekai WU
Human Language Technology Center
Dept. of Computer Science & Engineering
HKUST
Hong Kong
dekai@cs.ust.hk

Abstract

We argue that learning word alignments through a compositionally-structured, joint process yields higher phrase-based translation accuracy than the conventional heuristic of intersecting conditional models. Flawed word alignments can lead to flawed phrase translations that damage translation accuracy. Yet the IBM word alignments usually used today are known to be flawed, in large part because IBM models (1) model reordering by allowing unrestricted movement of words, rather than constrained movement of compositional units, and therefore must (2) attempt to compensate via directed, asymmetric distortion and fertility models. The conventional heuristics for attempting to recover from the resulting alignment errors involve estimating two directed models in opposite directions and then intersecting their alignments – to make up for the fact that, in reality, word alignment is an inherently joint relation. A natural alternative is provided by Inversion Transduction Grammars, which estimate the joint word alignment relation directly, eliminating the need for any of the conventional heuristics. We show that this alignment ultimately produces superior translation accuracy on BLEU, NIST, and METEOR metrics over three distinct language pairs.

1 Introduction

In this paper we argue that word alignments learned through a compositionally-structured, joint process are able to significantly improve the training of phrase-based translation systems, leading to higher translation accuracy than the conventional heuristic of intersecting conditional models. Today, statistical machine translation (SMT) systems perform at state-of-the-art levels; their ability to weigh different translation hypotheses against each other to find an optimal solution has proven to be a great asset. What sets various SMT systems apart are the models employed to determine what to consider optimal. The most common systems today consist of phrase-based models, where chunks of texts are substituted and rearranged to produce the output sentence.

Our premise is that certain flawed word alignments can lead to flawed phrase translations that in turn damage translation accuracy, since word alignment is the basis for learning phrase translations in phrase-based SMT systems. A critical part of such systems is the word-level translation model, which is estimated from aligned data. Currently, the standard way of computing a word alignment is to estimate a function linking words in one of the languages to words in the other. Functions can only define many-to-one relations, but word alignment is a many-to-many relation. The solution is to combine two functions, one in each direction, and harmonize them by means of some heuristic. After that, phrases can be extracted from the word alignments.

The problem is that the starting point for word alignments is usually the IBM models (Brown *et al*., 1993), which are known to produce flawed alignments, in large part because they (1) model reordering by allowing unrestricted movement of words, rather than constrained movement of compositional units, and therefore must (2) attempt to compensate via directed, asymmetric distortion and fertility models.

Proceedings of SSST-3, Third Workshop on Syntax and Structure in Statistical Translation, pages 28–36,
Boulder, Colorado, June 2009.

The conventional heuristics for attempting to recover from the resulting alignment errors is to estimate two directed models in opposite directions and then intersect their alignments – to make up for the fact that, in reality, word alignment is an inherently joint relation. It is unfortunate that such a critical stage in the training process of an SMT system relies on inaccurate heuristics, which have been largely motivated by historical implementation factors, rather than principles explaining language phenomena.

Inversion Transduction Grammar (ITG) models provide a natural, alternative approach, by estimating the joint word alignment relation directly, eliminating the need for any of the conventional heuristics. A transduction grammar is a grammar that generates sentences in two languages (L_0 and L_1) simultaneously; i.e., one start symbol expands into two strings, as for example in Figure 1(b). A transduction grammar explains two languages simultaneously. ITGs model a class of transductions (sets of sentence translations) with expressive power and computational complexity falling between (a) finite-state transducers or FSTs and (b) syntax-directed transduction grammars[1] or SDTGs. An ITG produces both a common structural form for a sentence pairs, as well as relating the words – aligning them. This could actually work as the joint word alignment that is usually constructed by heuristic function combination.

Yet despite the substantial body of literature on word alignment, ITG based models, and phrase-based SMT, the existing work has not assessed the potential for improving phrase-based translation quality by using joint ITG based word alignments to replace the error-prone conditional IBM model based word alignments and associated heuristics for intersecting bidirectional IBM alignments.

On one hand, word alignment work is usually evaluated not on actual translation quality, but rather on artificial metrics like alignment error rate (AER, Och & Ney, 2003), which relies on a manually annotated gold standard word alignment. There are some indications that ITG produces better alignment then the standard method (Zhao & Vogel, 2003, Zhang & Gildea 2005, Chao & Li, 2007). There is, however, little inherent utility in alignments – their value is determined by the SMT systems one can build from them. In fact, recent studies have discredited the earlier assumption that lower AER is correlated with improved translation quality – the opposite can very well occur (Ayan & Dorr, 2006). Therefore it is essential to evaluate the quality of the word alignment not in terms of AER, but rather in terms of actual translation quality in a system built from it.

On the other hand, ITG models have been employed to improve translation quality as measured by BLEU (Papineni *et al.*, 2002), but still without directly addressing the problem of dependence on inaccurate IBM alignments. Sánchez & Benedí (2006) construct an ITG from word alignments computed by the conventional IBM model, which does little to alleviate the problems. Sima'an & Mylonakis (2008) use an ITG to structure a prior distribution to a phrase extraction system, which is an altogether different approach. Cherry & Lin (2007) do use ITG to build word alignments, but blur the lines by still mixing in the conventional IBM method, and focus on phrase extraction.

The present work clearly demonstrates, for the first time to our knowledge, that replacing the widely-used heuristic of intersecting IBM word alignments from two directed conditional models instead with a single ITG alignment from a joint model produces superior translation accuracy. The experiments are performed on three distinct language pairs: German–English, Spanish–English, and French–English. Translation accuracy is reported in terms of BLEU, NIST, and METEOR metrics.

2 Background

Statistical Machine Translation is a paradigm where translation is considered as a code-breaking problem. The goal is to find the most likely output sentence (clear text message) given the supplied input sentence (coded message), according to some model.

To get a probabilistic model, large amounts of training data are used. These data have to be aligned so that an understanding of correspondences between the languages is there to be learnt from. Even if the data is assumed to be aligned at sentence level, sub-sentence alignment is also needed. This is usually carried out by training some statistical model of a word-to-word function (Brown *et al*., 1993), or a hidden Markov model consuming input words and emitting output words

[1] Which "synchronous CFGs" are essentially identical to.

(Vogel, Ney & Tillmann, 1996). The toolkit GIZA++ (Och & Ney, 2000) is freely available and widely used to compute such word alignments.

All these models learn a directed translation function that maps input words to output words. Since these functions focus solely on surface phenomena, they have no mechanisms for dealing with the kind of structured reordering between languages that could account for, e.g., the difference between SVO languages and SOV languages.

What emerges is in fact a rather flawed model of how one language is rewritten into another. The conventional way to alleviate this flaw is to train an equally flawed model in the other direction, and then intersect the two. This practice certainly alleviates some of the problems, but far from all.

To build a phrase-based SMT system, the word alignment is used as a starting point to try to account for the entire sentence. This means that the word alignment is gradually expanded, so that all words in both sentences are accounted for, either by words in the other language, or by the *null* empty word ε. This process is called grow-diag-final (Koehn, Och & Marcu, 2003).

The grow-diag-final process does smooth over some of the flaws still left in the word alignment, but error analysis gives reason to doubt that it repairs enough of the errors to avoid damaging translation accuracy. Thus, we are motivated to investigate a completely different approach that attempts to avoid the noisy directed alignments in the first place.

2.1 Inversion Transduction Grammars

A **transduction** is a set of sentence translation pairs – just as a language is a set of sentences. The set defines a relation between the input and output languages.

In the *generative* view, a **transduction grammar** generates a transduction, i.e., a set of sentence translation pairs or **bisentences** – just as an ordinary (monolingual) language grammar generates a language, i.e., a set of sentences. In the *recognition* view, alternatively, a transduction grammar **biparses** or accepts all sentence pairs of a transduction – just as a language grammar parses or accepts all sentences of a language. And in the *transduction* view, a transduction grammar **transduces** (translates) input sentences to output sentences.

Two familiar classes of transductions have been in widespread use for decades in many areas of computer science and linguistics:

> A **syntax-directed transduction** is a set of bisentences generated by some **syntax-directed transduction grammar** or SDTG (Lewis & Stearns, 1968; Aho & Ullman, 1969, 1972). A "synchronous CFG" is equivalent to an SDTG.

> A **finite-state transduction** is a set of bisentences generated by some **finite-state transducer** or FST. It is possible to describe finite-state transductions using SDTGs (or synchronous CFGs) by restricting them alternatively to the special cases of either "right regular SDTGs" or "left regular SDTGs". However, such characterizations rather misleadingly overlook the key point – by severely limiting expressive power, finite-state transductions are orders of magnitude cheaper to biparse, train, and induce than syntax-directed transductions – and are often even more accurate to induce.

More recently, an intermediate equivalence class of transductions whose generative capacity and computational complexity falls in between these two has become widely used in state-of-the-art MT systems – due to numerous empirical results indicating significantly better fit to modeling translation between many human language pairs:

> An **inversion transduction** is a set of bisentences generated by some **inversion transduction grammar** or ITG (Wu, 1995a, 1995b, 1997). As above with finite-state transductions, it is possible to describe inversion transductions using SDTGs (or synchronous CFGs) by restricting them alternatively to the special cases of "binary SDTGs", "ternary SDTGs", or "SDTGs whose transduction rules are restricted to straight and inverted permutations only". Again however, as above, such characterizations rather misleadingly overlook the key point – by severely limiting expressive power, inversion transductions are orders of magnitude cheaper to biparse, train, and induce than syntax-directed transductions – and are often even more accurate to induce.

Any SDTG (or synchronous CFG) of binary rank – i.e., that has at most two nonterminals on the right-hand-side of any rule – is an ITG. (Similarly, any SDTG (or synchronous CFG) that is right regular is a finite-state transduction grammar.) Thus, for example, any grammar computed by the binarization algorithm of Zhang *et al.*

(2006) is an ITG. Similarly, any grammar induced following the hierarchical phrase-based translation method, which always yields a binary transduction grammar (Chiang 2005), is an ITG.

Moreover, any SDTG (or synchronous CFG) of ternary rank – i.e., that has at most three nonterminals on the right-hand-side of any rule – is still equivalent to an ITG. Of course, this does not hold for SDTGs (or synchronous CFGs) in general, which allow arbitrary rank (possibly exceeding three) at the price of exponential complexity, as summarized in Table 1.

monolingual		*bilingual*	
regular or finite-state languages **FSA** *or* *CFG that is right regular or left regular*	$O(n^2)$	regular or finite-state transductions **FST** *or* *SDTG (or synchronous CFG) that is right regular or left regular*	$O(n^4)$
context-free languages **CFG**	$O(n^3)$	inversion transductions **ITG** *or* *SDTG (or synchronous CFG) that is binary or ternary or inverting*	$O(n^6)$
		syntax-directed transductions **SDTG** *(or synchronous CFG)*	$O(n^{2n+2})$

Table 1: Summary comparison of computational complexity for Viterbi and chart (bi)parsing, and EM training algorithms for both monolingual and bilingual hierarchies.

Without loss of generality, any ITG can be conveniently written in a **2-normal form** (Wu, 1995a, 1997). This cannot be done for SDTGs (or synchronous CFGs) – unlike the monolingual case of CFGs, which form an equivalence class of context-free languages that can all be written in Chomsky's 2-normal form. In the bilingual case, only ITGs form an equivalence class of inversion transductions that can all be written in a 2-normal form.

Formally, an ITG in this 2-normal form, which segregates syntactic versus lexical rules, consists of a tuple $\langle N, V_0, V_1, R, S \rangle$ where N is a set of nonterminal symbols, V_0 and V_1 are the vocabularies of L_0 and L_1 respectively, R is a set of transduction rules, and $S \in N$ is the start symbol. Each **transduction rule** takes one of the following forms:

$S \rightarrow X$
$X \rightarrow [Y\ Z]$
$X \rightarrow \langle Y\ Z \rangle$
$X \rightarrow \text{segment}_{L0}/\varepsilon$
$X \rightarrow \varepsilon/\text{segment}_{L1}$
$X \rightarrow \text{segment}_{L0}/\text{segment}_{L1}$

where X, Y and Z may be any nonterminal.

Aside from the start rule, there are two kinds of **syntactic transduction rules**, namely **straight** and **inverted**. In the above notation, straight transduction rules $X \rightarrow [Y\ Z]$ use square brackets, whereas inverted rules $X \rightarrow \langle Y\ Z \rangle$ use angled brackets. The transductions generated by straight nodes have the same order in both languages, whereas the transduction generated by the inverted nodes are inverted in one of the languages, meaning that the children are read left-to-right in L_0 and right-to-left in L_1. In Figure 1(b) for example, the parse tree node instantiating an inverted transduction rule is marked with a horizontal bar. This mechanism allows for a minimal amount of reordering, while keeping the complexity down.

The last three forms are for **lexical transduction rules**. Each segment comes from the vocabulary of one of the languages, indicated by the subscript. In the simplest case, the two ε-rule forms define **singletons**, which insert "spurious" segments into either language. Spurious segments lack any correspondence in the other language – they are "aligned to *null*" – and singletons are lexical rules that associate a *null*-aligned segment in one of the languages with an empty segment (ε) in the other.

On the other hand, the last rule form defines a **lexical translation pair** that aligns the word/phrase segment_{L0} to its translation segment_{L1}. Such rules can also be written compositionally as a pair of singletons, although it reads less transparently:

$X \rightarrow \text{segment}_{L0}/\varepsilon\ \ \varepsilon/\text{segment}_{L1}$

Note that **segments** typically consist of multiple **tokens**. Common examples include:

- Chinese word/phrase segments consisting of multiple unsegmented character tokens
- Chinese word/phrase segments consisting of multiple smaller, presegmented multi-character word/phrase tokens
- English phrase/collocation segments consisting of multiple word tokens (*roller coaster*)

ITGs inherently model phrasal translation – linguistically speaking, ITGs assume the set of lexical translation pairs constitutes a **phrasal lexicon** (just as lexicographers assume in building ordinary everyday dictionaries). An advantage of this is that the ITG biparsing and decoding algorithms perform integrated **translation-driven segmentation** simultaneously with optimizing the parse (Wu, 1997; Wu & Wong, 1998).

These properties allow an ITG to (1) insert and delete words/phrases, which matches the ability of the conventional methods for word alignment as well as phrase alignment, and (2) account for the reordering in a more principled and restricted way than conventional alignment methods.

A **stochastic ITG** or SITG is an ITG where every rule is associated with a probability. As with a stochastic CFG (SCFG), the probabilities are conditioned on the left-hand-side symbol, so that the probability of rule $X \rightarrow \chi$ is $p(\chi|X)$.

A **bracketing ITG** or BITG or BTG (Wu, 1995a) contains only one nonterminal symbol, with syntactic transduction rules $X \rightarrow [X\ X]$ and $X \rightarrow \langle X\ X \rangle$, which means that it produces a bracketing rather than a labeled tree. With a **stochastic BITG** (SBITG or SBTG) it is still possible to determine an optimal tree, since inversion and alignment are coupled: where inversions are needed is decided by the translations, and vice versa.

In Wu (1995b) algorithms for training a SITG using expectation maximization, as well as finding the optimal parse of a sentence pair given a SITG are presented. These are polynomial time $O(n^6)$, as seen in Table 1. Further pruning methods can also be added, especially for longer sentences.

2.2 Previous uses of ITG in alignment

There have been several attempts to use various forms of ITGs in an alignment setting.

Zhao & Vogel (2003) and Sánchez & Benedí (2006) both use GIZA++ to establish their SITG. Since they use GIZA++ to create their ITG, little light is shed on the question of whether an ITG produces better alignments than GIZA++.

Zhang & Gildea (2005) compare lexicalized and standard ITGs on an alignment task, and conclude that both are superior to IBM models 1 and 4, and that lexicalization helps. They also employ some pruning techniques to speed up training. Chao & Li (2007) incorporate the reordering constraints imposed by an ITG to their discriminative word aligner, and also note a lower alignment error rate in their system. Since neither work evaluates results on a translation task, it is hard to know whether better AER would translate into improved translation quality, in light of Ayan & Dorr (2006).

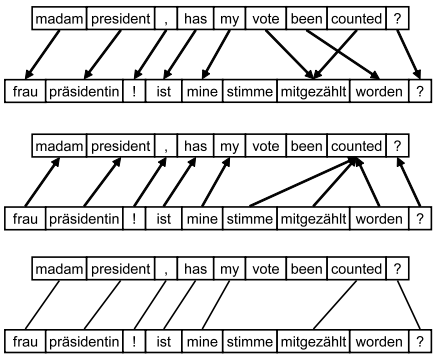
Sima'an & Mylonakis (2008) use an ITG as the basis of a prior distribution in their system that extracts all possible phrases rather than employing a length cut-off, and report an increase in translation quality as measured by the BLEU score (Papineni *et al.*, 2002). In this paper, it is not primarily pure ITG that is being evaluated, but it lends some credibility to our assumption that the ITG structure helps when aligning.

Cherry & Lin (2007) use an ITG to produce phrase tables that are then used in a translation system. However, to make their system outperform GIZA++, they blend in a non-compositionality constraint that is still based on GIZA++ word alignments. We would very much like to clearly see and understand the difference between ITG and GIZA++ alignments, and the lines are somewhat blurred in their work.

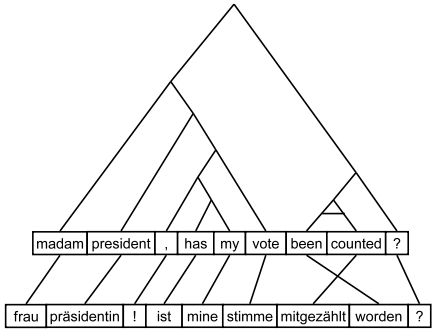
3 Model

First, the lexicon of the SBITG is initialized, by extracting lexical transduction rules from cooccurrence data from the corpus. Each pair of tokens in each sentence pair is initially considered equally likely to be a lexical translation pair. Each token is also considered to be a possible singleton. The two syntactic transduction rules $X \rightarrow [X\ X]$ and $X \rightarrow \langle X\ X \rangle$ are initially assumed to be equally likely.

Then full expectation-maximization training (Wu, 1995b) is carried out on the training data. Instead of waiting for full convergence, the process is halted when the increase in the training data's probability starts to decline.

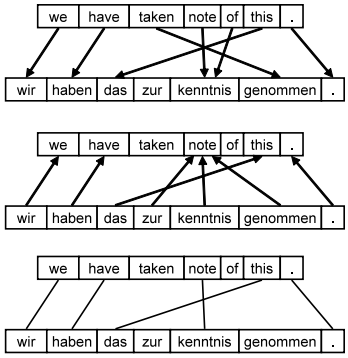


(a)

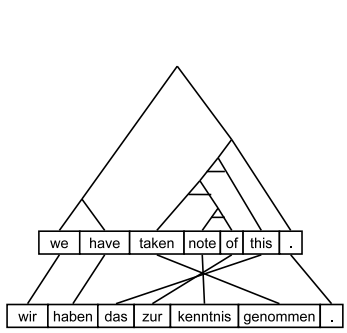


(b)

Figure 1: (a) Bidirectional IBM alignments and their intersection and (b) ITG alignments.

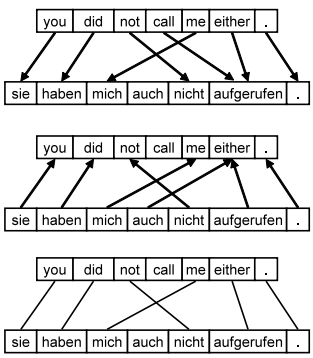


(a)

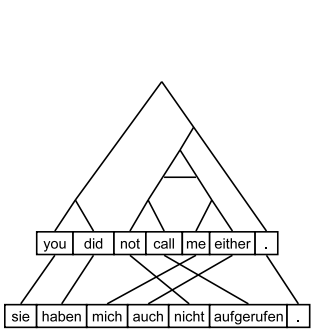


(b)

Figure 2: (a) Bidirectional IBM alignments and their intersection and (b) ITG alignments.



(a)



(b)

Figure 3: (a) Bidirectional IBM alignments and their intersection and (b) ITG alignments.

At this point, we extract the optimal parses from the training data, and use the word alignment imposed by the ITG instead of the one computed by GIZA++ (Och & Ney, 2000). Training after this point is carried out according to the guidelines for the WSMT08 baseline system (see section 4.2). In Figure 1(a) is an example of a sentence aligned with GIZA++, and in Figure 1(b) is the same sentence, aligned with ITG. In this case it is clearly visible how the structured reordering constraints that the ITG enforces results in a clear alignment, whereas GIZA++ is unable to sort it out.

	sentence pairs	tokens
de-en	115,323	1,602,781
es-en	108,073	1,466,132
fr-en	95,990	1,340,718

Table 2: Summary of training data.

4 Experimental setup

4.1 Data

We used a subset of the data provided for the Second Workshop on Statistical Machine Translation[2], which consists mainly of texts from the Europarl corpus (Koehn, 2005). We used the Europarl part for the translation tasks: German–English (de-en), Spanish–English (es-en), and French–English (fr-en). Table 2 summarizes the datasets used for training. For tuning and testing, the tuning and development test sets provided for the workshop were used – each measuring 2,000 sentence pairs.

4.2 Baseline system

For baseline system we trained phrase-based SMT models with GIZA++ (Och & Ney, 2000), the training scripts supplied with Moses (Koehn *et al.*,

[2] *www.statmt.org/wmt08*

2007), and minimum error rate training (MERT, Och, 2003), all according to the WSMT08-guidelines for baseline systems. This means that 5 iterations are carried out with IBM model 1 training, 5 iterations with HMM training, 3 iterations of IBM model 3 training, and finally 3 iterations of IBM model 4 training. After GIZA++ training, the Moses training script extracts and scores phrases, and establishes a lexicalized reordering model.

The WSMT08 guidelines call for the combination heuristic "grow-diag-final-and" (GDFA). We also tried the "intersect" combination heuristic, which simply calculates the intersection of alignment points in the two directed alignments provided by GIZA++.

4.3 SBITG system

Since imposing an SBITG biparse on a sentence pair forces a word alignment on the sentence pair, word alignment under SBITG models is identical to biparsing.

Expectation-maximization training was used to induce a SBITG from the training data. Training is halted when the EM-process started to converge. In our experience, convergence typically requires no more than 3 iterations or so. When EM training is finished, we extracted the optimal biparses from the training data, which then constitute the optimal alignment given the grammar. This alignment was then output in GIZA++ format. All singletons from the SBITG alignment were converted to be *null*-alignments in the GIZA++ formatted file. These files could then be used instead of GIZA++ in the remainder of the training process for the phrase-based translation system.

Although the results from the ITG are interpreted as two directed alignments, they are identical, both with each other and the intersection. Trying different combination heuristics for these results always yields the same results.

The training process was identical save for the fact that the word alignments were produced by SBITGs rather than by GIZA++.

5 Experimental results

We trained a total of nine systems (three tasks and three different alignments), which we evaluated with three different measures: BLEU (Papineni *et al.*, 2002), NIST (Doddington, 2002), and METEOR (Lavie & Agarwal 2007).

Figure 2 shows a sentence pair as it was aligned with the two different models. Figure 2(a) shows the GIZA++ alignment in both directions, and the intersection between them, whereas Figure 2(b) shows the SBITG alignment with its common structure. The asymmetric reordering mechanism of the IBM models is simply unable to relate the two halves to one another. The segment *zur kenntnis genommen* could certainly be said to mean *note*, but as a verb, and not as a noun, which is the current usage of the word. This is an inherent problem of the asymmetry of the IBM models, which is rectified by simultaneous alignment.

Figure 3 shows another sentence pair. Again, Figure 3(a) was aligned with GIZA++ and Figure 3(b) with the SITG model. This shows a case with perhaps even more structured reordering, where a notion of constituency is definitely needed to get it right. SITG handles constituency, and gets this issue right. The IBM models do not, resulting in the error of aligning *either* to *aufgerufen*.

As mentioned before, the GDFA heuristic is applied after the word alignment process, and it does fix some of these problems. Therefore we opted to evaluate this, not on alignments, but rather on translation quality of phrase based SMT systems derived from the alignments. Our empirical results confirm that SBITG alignments do indeed lead to better translation quality, as shown in Table 2.

We also tried the intersect combination heuristic, and depending on language pair and evaluation metric, the GDFA and intersect heuristics come out on top. The ITG approach is, however, consistently better than either of the heuristics applied to GIZA++ output.

6 Discussion

There are of course fundamental differences between ITG and IBM models. The main difference is that IBM models are directed and surface oriented, whereas the ITG model is joint and structured. The directedness means that the IBM models are unable to produce a word alignment that is optimal for a sentence pair; they can only produce word alignments that are optimal when translating from one language into the other. An ITG on the other hand is capable of producing the optimal alignment that explains both sentences in the pair. We see this phenomenon clearly in Figures 1–3.

	BLEU			NIST			METEOR		
	GIZA++		SBITG	GIZA++		SBITG	GIZA++		SBITG
	GDFA	inters.		GDFA	inters.		GDFA	inters.	
de-en	20.59	20.69	**21.13**	5.8668	5.8623	**5.9380**	0.4969	0.4953	**0.5029**
es-en	25.97	26.33	**26.63**	6.6352	6.6793	**6.7407**	0.5599	0.5582	**0.5612**
fr-en	26.03	26.17	**26.63**	6.6907	6.7071	**6.8151**	0.5544	0.5560	**0.5635**

Table 2: Results. The best result on each task/metric combination is in bold digits.
(The identical results for SBITG on Spanish–English and French–English are not typos.)

IBM models are also built to allow for fairly "whimsical" reorderings, which are not modeled very well to begin with. This allows for far too many degrees of freedom to fit the model to the data. Because natural languages are inherently structural, this excess degree of freedom could hurt performance. Some restraints are needed. ITGs on the other hand only allow for compositionally structured reordering, which corresponds better to the reorderings between natural languages. There are some issues with ITG as well, one of them being that all permutations are actually not allowed, even if structured. This has led to some problems when an a prior alignment or structure is forced upon a sentence pair, but using unrestricted expectation-maximization means that the sentence pair is fitted to the grammar, and what the grammar cannot express is not applied to the data. Even if ITG proves to be too restrictive in the future, the fact that it bases reordering on structure, rather than unrestricted lexical movement, gives it an edge over the IBM models. The benefits of structured reordering as opposed to unrestricted are clearly visible in Figures 1–3.

An argument to continue using IBM models is that two directed alignments can be intersected and heuristically grown to build a joint alignment, thus compensating for the flaws in the original models. But as we have seen in Figure 3, even the combination of two models contains errors that should have been avoided. This approach is not able to smooth over the flaws of the IBM models.

The results in this paper give credibility to the claim that these limitations of the IBM models are so serious that they hurt translation quality of systems built upon them; even after the phrase building heuristic has been applied. Systems built on ITG alignment on the other hand fare better, on all three evaluation metrics.

There is still more to be done. So far we have only employed bracketing SITGs, which are not able to distinguish one structure form another. The structural changes that the SBITG is capable of are dictated by the alignment of the leaves in the tree. This seems impressive, given the information at hand, but is really a logical conclusion of the fact that the grammar can leverage different alignment probabilities against each other, and as the alignment is coupled to the structure of the ITG parse, the structure is constrained to the alignment. The reverse is also true: the alignment is constrained by the structure. This coupling is essential to the training of SITGs. For a SBITG, there is very little information in the structure, only the decision to read the node as straight or inverted. This is not an inherent property of ITGs in general; more information can be carried higher up in the tree by labeling the nonterminals. There is great hope that adding more information to the structuring, even better alignments could be gained.

In this paper we have extracted the word alignments from ITG biparses, and inserted them into the conventional phrase-based SMT pipeline. It is feasible to extract phrases directly from the grammar, as demonstrated by Cherry & Lin (2007). Our results suggest that augmenting other portions of the phrase-based SMT framework with ITG structures might also be worth exploring, in particular decoding. Recall that in the transduction view of transduction grammars (as opposed to generative or recognition views), an output translation can be determined by parsing an input sentence with a transduction grammar (Wu 1996; Wu & Wong 1998). This kind of translation would also entail the notion of structure that we have just witnessed helping alignment. Phrase-based SMT currently relies on unrestricted phrasal movement, which is a lot better than unrestricted lexical movement, but could probably use some structure as well.

7 Conclusion

We have shown that learning word alignments through a compositionally-structured, joint process yields higher phrase-based translation accuracy than the conventional heuristic of intersecting conditional models.

The conventional method with IBM-models suffers from their directionality. The asymmetry causes bad alignments. We have instead introduced an automatically induced ITG alignment that does not suffer from this asymmetry, and is able to explain the two sentences simultaneously rather than one in terms of the other. The IBM-models also suffers from a simplified reordering model, which relies on moving individual words. The hierarchical structure of ITGs means that even a BITG has enough structural information to outperform the IBM models. Previous work shows that these advantages translate into better alignments as measured against a manually annotated gold standard using alignment error rate (AER). Previous work also shows that AER is a poor indicator of whether translation quality is increased. We have showed that the increase in alignment quality actually translates into an increase in translation quality in this case, as measured by BLEU, NIST and METEOR across three different language pairs.

Acknowledgments

This material is based upon work supported in part by the Swedish National Graduate School of Language Technology, the Olof Gjerdmans Travel Grant, the Defense Advanced Research Projects Agency (DARPA) under GALE Contract No. HR0011-06-C-0023, and the Hong Kong Research Grants Council (RGC) under research grants GRF621008, DAG03/04.EG09, RGC6256/00E, and RGC6083/99E. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency.

References

AHO, Alfred V. & Jeffrey D. ULLMAN (1969) "Syntax-directed translations and the pushdown assembler" in *Journal of Computer and System Sciences* **3**: 37–56.

AHO, Alfred V. & Jeffrey D. ULLMAN (1972) *The Theory of Parsing, Translation, and Compiling* (Volumes 1 and 2). Englewood Cliffs, NJ: Prentice-Hall.

AYAN, Necip Fazil & Bonnie J. DORR (2006) "Going Beyond AER: An Extensive Analysis of Word Alignments and Their Impact on MT" in *COLING-ACL'06*, pp. 9–16, Sydney, Australia, July 2006.

BROWN, Peter F., Stephen A. DELLA PIETRA, Vincent J. DELLA PIETRA & Robert L. MERCER (1993) "The Mathematics of Statistical Machine Translation" in *Computational Linguistics* **19**(2): 263–311.

CHAO, Wen-Han & Zhou-Jun LI (2007) "Incorporating Constituent Structure Constraint into Discriminative Word Alignment" in *MT Summit XI*, pp. 97–103, Copenhagen, Denmark.

CHERRY, Colin & Dekang LIN (2007) "Inversion Transduction Grammar for Joint Phrasal Translation Modeling" in *Proceedings of SSST*, pp. 17–24, Rochester, New York, April 2007.

CHIANG, David (2005) "A Hierarchical Phrase-Based Model for Statistical Machine Translation" in *ACL-2005*, pp. 263–270, Ann Arbor, MI, June 2005.

KOEHN, Philipp, Franz Josef OCH & Daniel MARCU (2003) "Statistical Phrase-based Translation" in *HLT-NAACL'03*, pp. 127–133.

KOEHN, Philipp (2005) "Europarl: A Parallel Corpus for Statistical Machine Translation" in *MT Summit X*, Phuket, Thailand, September 2005.

DODDINGTON, George (2002) "Automatic Evaluation of Machine Translation Quality using n-gram Co-occurrence Statistics" in *HLT-2002*. San Diego, California.

KOEHN, Philipp, Hieu HOANG, Alexandra BIRCH, Chris CALLISON-BURCH, Marcello FEDERICO, Nicola BERTOLDI, Brooke COWAN, Wade SHEN, Christine MORAN, Richard ZENS, Chris DYER, Ondrej BOJAR, Alexandra CONSTANTIN & Evan HERBST (2007) "Moses: Open Source Toolkit for Statistical Machine Translation" in *ACL'07*, Prague, Czech Republic, June 2007.

LAVIE, Alon & Abhaya AGARWAL (2007) "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgment" in *WSMT*. Prague, Czech Republic, June 2007.

LEWIS, Philip M. & Richard E. STEARNS. (1968) "Syntax-directed transduction" in *Journal of the ACM* **15**: 465–488.

OCH, Franz Josef & Hermann NEY (2000) "Improved Statistical Alignment Models" in *ACL-2000*, pp. 440–447, Hong Kong, October 2000.

OCH, Franz Josef (2003) "Minimum error rate training in statistical machine translation" in *ACL'03*.

OCH, Franz Josef & Hermann NEY (2003) "A Systematic Comparison of Various Statistical Alignment Models" in *Computational Linguistics* 29(1), pp. 19–52.

PAPINENI, Kishore, Salim ROUKOS, Todd WARD & Wei-Jing ZHU (2002) "BLEU: a Method for Automatic Evaluation of Machine Translation" in *ACL'02*, pp 311-318. Philadelphia, Pennsylvania.

SÁNCHEZ, J. A., J.M. BENEDÍ (2006) "Stochastic Inversion Transduction Grammars for Obtaining Word Phrases for Phrase-based Statistical Machine Translation" in *WSMT*, pp. 130–133, New York City, June 2006.

SIMA'AN, Khalil & Markos MYLONAKIS (2008) "Better Statistical Estimation Can Benefit all Phrases in Phrase-based Statistical Machine Translation" in *SLT 2008*, pp. 237–240, Goa, India, December 2008.

VOGEL, Stephan, Hermann NEY & Christoph TILLMANN (1996) "HMM-based Word Alignment in Statistical Translation" in *COLING'96*, pp. 836–841.

WU, Dekai (1995a) "An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words" in *ACL'95*, pp. 244–251, Cambridge, Massachusetts, June 1995.

WU, Dekai (1995b) "Trainable Coarse Bilingual Grammars for Parallel Text Bracketing" in *WVLC-3*, pp. 69–82, Cambridge, Massachusetts, June 1995.

WU, Dekai (1996) "A polynomial-time algorithm for statistical machine translation" in *ACL-96*, Santa Cruz, CA: June 1996.

WU, Dekai (1997) "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora" in *Computational Linguistics* 23(3), pp 377–403.

WU, Dekai & Hongsing WONG (1998) "Machine Translation with a Stochastic Grammatical Channel" in *COLING-ACL'98*, Montreal, August 1998.

ZHANG, Hao & Daniel GILDEA (2005) "Stochastic Lexicalized Inversion Transduction Grammar for Alignment" in *ACL'05*, pp. 475–482, Ann Arbor, June 2005.

ZHANG, Hao, Liang HUANG, Dan GILDEA & Kevin KNIGHT (2006) "Synchronous Binarization for Machine Translation" in *HLT/NAACL-2006*, pp. 256–263, New York, June 2006.

ZHAO, Bing & Stephan VOGEL (2003) "Word Alignment Based on Bilingual Bracketing" in *HLT-NAACL Workshop: Building and Using Parallel Texts*, pp. 15–18, Edmonton, May–June 2003.