Inversion Transduction Grammar for Joint Phrasal Translation Modeling

Colin Cherry
Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 2E8
colinc@cs.ualberta.ca

Dekang Lin
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA, USA, 9403
lindek@google.com

Abstract

We present a phrasal inversion transduction grammar as an alternative to joint phrasal translation models. This syntactic model is similar to its flat-string phrasal predecessors, but admits polynomial-time algorithms for Viterbi alignment and EM training. We demonstrate that the consistency constraints that allow flat phrasal models to scale also help ITG algorithms, producing an 80-times faster inside-outside algorithm. We also show that the phrasal translation tables produced by the ITG are superior to those of the flat joint phrasal model, producing up to a 2.5 point improvement in BLEU score. Finally, we explore, for the first time, the utility of a joint phrasal translation model as a word alignment method.

1 Introduction

Statistical machine translation benefits greatly from considering more than one word at a time. One can put forward any number of non-compositional translations to support this point, such as the colloquial Canadian French-English pair, (*Wo les moteurs*, *Hold your horses*), where no clear word-to-word connection can be drawn. Nearly all current decoding methods have shifted to phrasal representations, gaining the ability to handle non-compositional translations, but also allowing the decoder to memorize phenomena such as monolingual agreement and short-range movement, taking pressure off of language and distortion models.

Despite the success of phrasal decoders, knowledge acquisition for translation generally begins with a word-level analysis of the training text, taking the form of a word alignment. Attempts to apply the same statistical analysis used at the word level in a phrasal setting have met with limited success, held back by the sheer size of phrasal alignment space. Hybrid methods that combine well-founded statistical analysis with high-confidence word-level alignments have made some headway (Birch et al., 2006), but suffer from the daunting task of heuristically exploring a still very large alignment space. In the meantime, synchronous parsing methods efficiently process the same bitext phrases while building their bilingual constituents, but continue to be employed primarily for word-to-word analysis (Wu, 1997). In this paper we unify the probability models for phrasal translation with the algorithms for synchronous parsing, harnessing the benefits of both to create a statistically and algorithmically well-founded method for phrasal analysis of bitext.

Section 2 begins by outlining the phrase extraction system we intend to replace and the two methods we combine to do so: the joint phrasal translation model (JPTM) and inversion transduction grammar (ITG). Section 3 describes our proposed solution, a phrasal ITG. Section 4 describes how to apply our phrasal ITG, both as a translation model and as a phrasal word-aligner. Section 5 tests our system in both these capacities, while Section 6 concludes.

2 Background

2.1 Phrase Table Extraction

Phrasal decoders require a phrase table (Koehn et al., 2003), which contains bilingual phrase pairs and

Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation, pages 17–24,
Rochester, New York, April 2007. ©2007 Association for Computational Linguistics

scores indicating their utility. The **surface heuristic** is the most popular method for phrase-table construction. It extracts all consistent phrase pairs from word-aligned bitext (Koehn et al., 2003). The word alignment provides bilingual links, indicating translation relationships between words. **Consistency** is defined so that alignment links are never broken by phrase boundaries. For each token w in a consistent phrase pair $\bar{p}$, all tokens linked to w by the alignment must also be included in $\bar{p}$. Each consistent phrase pair is counted as occurring once per sentence pair. The scores for the extracted phrase pairs are provided by normalizing these flat counts according to common English or Foreign components, producing the conditional distributions $p(\bar{f}|\bar{e})$ and $p(\bar{e}|\bar{f})$.

The surface heuristic can define consistency according to any word alignment; but most often, the alignment is provided by GIZA++ (Och and Ney, 2003). This alignment system is powered by the IBM translation models (Brown et al., 1993), in which one sentence generates the other. These models produce only one-to-many alignments: each generated token can participate in at most one link. Many-to-many alignments can be created by combining two GIZA++ alignments, one where English generates Foreign and another with those roles reversed (Och and Ney, 2003). Combination approaches begin with the intersection of the two alignments, and add links from the union heuristically. The grow-diag-final (GDF) combination heuristic (Koehn et al., 2003) adds links so that each new link connects a previously unlinked token.

2.2 Joint phrasal translation model

The IBM models that power GIZA++ are trained with Expectation Maximization (Dempster et al., 1977), or EM, on sentence-aligned bitext. A translation model assigns probabilities to alignments; these alignment distributions are used to count translation events, which are then used to estimate new parameters for the translation model. Sampling is employed when the alignment distributions cannot be calculated efficiently. This statistically-motivated process is much more appealing than the flat counting described in Section 2.1, but it does not directly include phrases.

The joint phrasal translation model (Marcu and Wong, 2002), or JPTM, applies the same statistical techniques from the IBM models in a phrasal setting. The JPTM is designed according to a generative process where both languages are generated simultaneously. First, a bag of concepts, or cepts, C is generated. Each $c_i \in C$ corresponds to a bilingual phrase pair, $c_i = (\bar{e}_i, \bar{f}_i)$. These contiguous phrases are permuted in each language to create two sequences of phrases. Initially, Marcu and Wong assume that the number of cepts, as well as the phrase orderings, are drawn from uniform distributions. That leaves a joint translation distribution $p(\bar{e}_i, \bar{f}_i)$ to determine which phrase pairs are selected. Given a lexicon of possible cepts and a predicate $L(E, F, C)$ that determines if a bag of cepts C can be bilingually permuted to create the sentence pair (E, F), the probability of a sentence pair is:

$$p(E, F) \propto \sum_{\{C|L(E,F,C)\}} \left[\prod_{c_i \in C} p(\bar{e}_i, \bar{f}_i) \right] \qquad (1)$$

If left unconstrained, (1) will consider every phrasal segmentation of E and F, and every alignment between those phrases. Later, a distortion model based on absolute token positions is added to (1).

The JPTM faces several problems when scaling up to large training sets:

1. The alignment space enumerated by the sum in (1) is huge, far larger than the one-to-many space explored by GIZA++.
2. The translation distribution $p(\bar{e}, \bar{f})$ will cover all co-occurring phrases observed in the bitext. This is far too large to fit in main memory, and can be unwieldly for storage on disk.
3. Given a non-uniform $p(\bar{e}, \bar{f})$, there is no efficient algorithm to compute the expectation of phrase pair counts required for EM, or to find the most likely phrasal alignment.

Marcu and Wong (2002) address point 2 with a **lexicon constraint**; monolingual phrases that are above a length threshold or below a frequency threshold are excluded from the lexicon. Point 3 is handled by hill-climbing to a likely phrasal alignment and sampling around it. However, point 1 remains unaddressed, which prevents the model from scaling to large data sets.

Birch et al. (2006) handle point 1 directly by reducing the size of the alignment space. This is accomplished by constraining the JPTM to only use phrase pairs that are consistent with a high-confidence word alignment, which is provided by GIZA++ intersection. We refer to this constrained JPTM as a C-JPTM. This strikes an interesting middle ground between the surface heuristic described in Section 2.1 and the JPTM. Like the surface heuristic, a word alignment is used to limit the phrase pairs considered, but the C-JPTM reasons about distributions over phrasal alignments, instead of taking flat counts. The consistency constraint allows them to scale their C-JPTM up to 700,000 sentence pairs. With this constraint in place, the use of hill-climbing and sampling during EM training becomes one of the largest remaining weaknesses of the C-JPTM.

2.3 Inversion Transduction Grammar

Like the JPTM, stochastic synchronous grammars provide a generative process to produce a sentence and its translation simultaneously. Inversion transduction grammar (Wu, 1997), or ITG, is a well-studied synchronous grammar formalism. Terminal productions of the form $A \rightarrow e/f$ produce a token in each stream, or a token in one stream with the null symbol $\emptyset$ in the other. To allow for movement during translation, non-terminal productions can be either straight or inverted. Straight productions, with their non-terminals inside square brackets $[\ldots]$, produce their symbols in the given order in both streams. Inverted productions, indicated by angled brackets $\langle\ldots\rangle$, are output in reverse order in the Foreign stream only.

The work described here uses the binary bracketing ITG, which has a single non-terminal:

$$A \rightarrow [AA] \mid \langle AA \rangle \mid e/f \tag{2}$$

This grammar admits an efficient bitext parsing algorithm, and holds no language-specific biases.

(2) cannot represent all possible permutations of concepts that may occur during translation, because some permutations will require discontinuous constituents (Melamed, 2003). This **ITG constraint** is characterized by the two forbidden structures shown in Figure 1 (Wu, 1997). Empirical studies suggest that only a small percentage of human translations violate these constraints (Cherry and Lin, 2006).

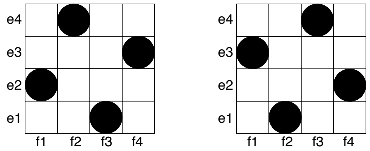
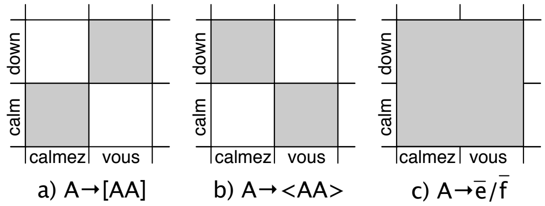


Figure 1: The two ITG forbidden structures.



a) A→[AA] b) A→<AA> c) A→$\bar{e}/\bar{f}$

Figure 2: Three ways in which a phrasal ITG can analyze a multi-word span or phrase.

Stochastic ITGs are parameterized like their PCFG counterparts (Wu, 1997); productions $A \rightarrow \mathcal{X}$ are assigned probability $\Pr(\mathcal{X}|A)$. These parameters can be learned from sentence-aligned bitext using the EM algorithm. The expectation task of counting productions weighted by their probability is handled with dynamic programming, using the inside-outside algorithm extended to bitext (Zhang and Gildea, 2004).

3 ITG as a Phrasal Translation Model

This paper introduces a phrasal ITG; in doing so, we combine ITG with the JPTM. ITG parsing algorithms consider every possible two-dimensional span of bitext, each corresponding to a bilingual phrase pair. Each multi-token span is analyzed in terms of how it could be built from smaller spans using a straight or inverted production, as is illustrated in Figures 2 (a) and (b). To extend ITG to a phrasal setting, we add a third option for span analysis: that the span under consideration might have been drawn directly from the lexicon. This option can be added to our grammar by altering the definition of a terminal production to include phrases: $A \rightarrow \bar{e}/\bar{f}$. This third option is shown in Figure 2 (c). The model implied by this extended grammar is trained using inside-outside and EM.

Our approach differs from previous attempts to use ITGs for phrasal bitext analysis. Wu (1997) used a binary bracketing ITG to segment a sentence while simultaneously word-aligning it to its translation, but the model was trained heuristically with a fixed segmentation. Vilar and Vidal (2005) used ITG-like dynamic programming to drive both training and alignment for their recursive translation model, but they employed a conditional model that did not maintain a phrasal lexicon. Instead, they scored phrase pairs using IBM Model 1.

Our phrasal ITG is quite similar to the JPTM. Both models are trained with EM, and both employ generative stories that create a sentence and its translation simultaneously. The similarities become more apparent when we consider the canonical-form binary-bracketing ITG (Wu, 1997) shown here:

$$\begin{array}{ll} S \rightarrow & A \mid B \mid C \\ A \rightarrow & [AB] \mid [BB] \mid [CB] \mid \\ & [AC] \mid [BC] \mid [CC] \\ B \rightarrow & \langle AA \rangle \mid \langle BA \rangle \mid \langle CA \rangle \mid \\ & \langle AC \rangle \mid \langle BC \rangle \mid \langle CC \rangle \\ C \rightarrow & \bar{e}/\bar{f} \end{array} \tag{3}$$

(3) is employed in place of (2) to reduce redundant alignments and clean up EM expectations.[1] More importantly for our purposes, it introduces a preterminal C, which generates all phrase pairs or cepts. When (3) is parameterized as a stochastic ITG, the conditional distribution $p(\bar{e}/\bar{f}|C)$ is equivalent to the JPTM's $p(\bar{e}, \bar{f})$; both are joint distributions over all possible phrase pairs. The distributions conditioned on the remaining three non-terminals assign probability to concept movement by tracking inversions. Like the JPTM's distortion model, these parameters grade each movement decision independently. With terminal productions producing cepts, and inversions measuring distortion, our phrasal ITG is essentially a variation on the JPTM with an alternate distortion model.

Our phrasal ITG has two main advantages over the JPTM. Most significantly, we gain polynomial-time algorithms for both Viterbi alignment and EM expectation, through the use of ITG parsing and inside-outside algorithms. These phrasal ITG algorithms are no more expensive asymptotically than their word-to-word counterparts, since each potential phrase needs to be analyzed anyway during constituent construction. We hypothesize that using these methods in place of heuristic search and sampling will improve the phrasal translation model learned by EM. Also, we can easily incorporate links to $\emptyset$ by including the symbol among our terminals. To minimize redundancy, we allow only single tokens, not phrases, to align to $\emptyset$. The JPTM does not allow links to $\emptyset$.

The phrasal ITG also introduces two new complications. ITG Viterbi and inside-outside algorithms have polynomial complexity, but that polynomial is $O(n^6)$, where n is the length of the longer sentence in the pair. This is too slow to train on large data sets without massive parallelization. Also, ITG algorithms explore their alignment space perfectly, but that space has been reduced by the ITG constraint described in Section 2.3. We will address each of these issues in the following two subsections.

3.1 Pruning Spans

First, we address the problem of scaling ITG to large data. ITG dynamic programming algorithms work by analyzing each bitext span only once, storing its value in a table for future use. There are $O(n^4)$ of these spans, and each analysis takes $O(n^2)$ time. An effective approach to speeding up ITG algorithms is to eliminate unlikely spans as a preprocessing step, assigning them 0 probability and saving the time spent processing them. Past approaches have pruned spans using IBM Model 1 probability estimates (Zhang and Gildea, 2005) or using agreement with an existing parse tree (Cherry and Lin, 2006). The former is referred to as tic-tac-toe pruning because it uses both inside and outside estimates.

We propose a new ITG pruning method that leverages high-confidence links by pruning all spans that are inconsistent with a provided alignment. This is similar to the constraint used in the C-JPTM, but we do not just eliminate those spans as potential phrase-to-phrase links: we never consider any ITG parse that builds a non-terminal over a pruned span.[2] This **fixed-link pruning** will speed up both Viterbi alignment and EM training by reducing the number of analyzed spans, and so long as we trust

[1] If the null symbol $\emptyset$ is included among the terminals, then redundant parses will still occur, but far less frequently.

[2] Birch et al. (2006) re-introduce inconsistent phrase-pairs in cases where the sentence pair could not be aligned otherwise. We allow links to $\emptyset$ to handle these situations, completely eliminating the pruned spans from our alignment space.

our high-confidence links, it will do so harmlessly. We demonstrate the effectiveness of this pruning method experimentally in Section 5.1.

3.2 Handling the ITG Constraint

Our remaining concern is the ITG constraint. There are some alignments that we just cannot build, and sentence pairs requiring those alignments will occur. These could potentially pollute our training data; if the system is unable to build the right alignment, the counts it will collect from that pair must be wrong. Furthermore, if our high-confidence links are not ITG-compatible, our fixed-link pruning will prevent the aligner from forming any alignments at all.

However, these two potential problems cancel each other out. Sentence pairs containing non-ITG translations will tend to have high-confidence links that are also not ITG-compatible. Our EM learner will simply skip these sentence pairs during training, avoiding pollution of our training data. We can use a linear-time algorithm (Zhang et al., 2006) to detect non-ITG movement in our high-confidence links, and remove the offending sentence pairs from our training corpus. This results in only a minor reduction in training data; in our French-English training set, we lose less than 1%. In the experiments described in Section 5, all systems that do not use ITG will take advantage of the complete training set.

4 Applying the model

Any phrasal translation model can be used for two tasks: translation modeling and phrasal word alignment. Previous work on JPTM has focused on only the first task. We are interested in phrasal alignment because it may be better suited to heuristic phrase-extraction than word-based models. This section describes how to use our phrasal ITG first as a translation model, and then as a phrasal aligner.

4.1 Translation Modeling

We can test our model's utility for translation by transforming its parameters into a phrase table for the phrasal decoder Pharaoh (Koehn et al., 2003). Any joint model can produce the necessary conditional probabilities by conditionalizing the joint table in both directions. We use our $p(\bar{e}/\bar{f}|C)$ distribution from our stochastic grammar to produce $p(\bar{e}|\bar{f})$ and $p(\bar{f}|\bar{e})$ values for its phrasal lexicon.

Pharaoh also includes lexical weighting parameters that are derived from the alignments used to induce its phrase pairs (Koehn et al., 2003). Using the phrasal ITG as a direct translation model, we do not produce alignments for individual sentence pairs. Instead, we provide a lexical preference with an IBM Model 1 feature p_{M1} that penalizes unmatched words (Vogel et al., 2003). We include both $p_{\mathrm{M1}}(\bar{e}|\bar{f})$ and $p_{\mathrm{M1}}(\bar{f}|\bar{e})$.

4.2 Phrasal Word Alignment

We can produce a translation model using inside-outside, without ever creating a Viterbi parse. However, we can also examine the maximum likelihood phrasal alignments predicted by the trained model.

Despite its strengths derived from using phrases throughout training, the alignments predicted by our phrasal ITG are usually unsatisfying. For example, the fragment pair (*order of business*, *ordre des travaux*) is aligned as a phrase pair by our system, linking every English word to every French word. This is frustrating, since there is a clear compositional relationship between the fragment's component words. This happens because the system seeks only to maximize the likelihood of its training corpus, and phrases are far more efficient than word-to-word connections. When aligning text, annotators are told to resort to many-to-many links only when no clear compositional relationship exists (Melamed, 1998). If we could tell our phrasal aligner the same thing, we could greatly improve the intuitive appeal of our alignments. Again, we can leverage high-confidence links for help.

In the high-confidence alignments provided by GIZA++ intersection, each token participates in at most one link. Links only appear when two word-based IBM translation models can agree. Therefore, they occur at points of high compositionality: the two words clearly account for one another. We adopt an alignment-driven definition of compositionality: any phrase pair containing two or more high-confidence links is compositional, and can be separated into at least two non-compositional phrases. By removing any phrase pairs that are compositional by this definition from our terminal productions, we can ensure that our aligner never creates such phrases during training or alignment. Doing so produces far more intuitive alignments. Aligned with a model trained using this **non-compositional constraint** (NCC), our example now forms three word-to-word connections, rather than a single phrasal one. The phrases produced with this constraint are very small, and include only non-compositional context. Therefore, we use the constraint only to train models intended for Viterbi alignment, and not when generating phrase tables directly as in Section 4.1.

5 Experiments and Results

In this section, we first verify the effectiveness of fixed-link pruning, and then test our phrasal ITG, both as an aligner and as a translation model. We train all translation models with a French-English Europarl corpus obtained by applying a 25 token sentence-length limit to the training set provided for the HLT-NAACL SMT Workshop Shared Task (Koehn and Monz, 2006). The resulting corpus has 393,132 sentence pairs. 3,376 of these are omitted for ITG methods because their high-confidence alignments have ITG-incompatible constructions. Like our predecessors (Marcu and Wong, 2002; Birch et al., 2006), we apply a lexicon constraint: no monolingual phrase can be used by any phrasal model unless it occurs at least five times. High-confidence alignments are provided by intersecting GIZA++ alignments trained in each direction with 5 iterations each of Model 1, HMM, and Model 4. All GIZA++ alignments are trained with no sentence-length limit, using the full 688K corpus.

5.1 Pruning Speed Experiments

To measure the speed-up provided by fixed-link pruning, we timed our phrasal inside-outside algorithm on the first 100 sentence pairs in our training set, with and without pruning. The results are shown in Table 1. Tic-tac-toe pruning is included for comparison. With fixed-link pruning, on average 95% of the possible spans are pruned, reducing running time by two orders of magnitude. This improvement makes ITG training feasible, even with large bitexts.

5.2 Alignment Experiments

The goal of this experiment is to compare the Viterbi alignments from the phrasal ITG to gold standard human alignments. We do this to validate our non-compositional constraint and to select good alignments for use with the surface heuristic.

Table 1: Inside-outside run-time comparison.

Method	Seconds	Avg. Spans Pruned
No Prune	415	-
Tic-tac-toe	37	68%
Fixed link	5	95%

Table 2: Alignment Comparison.

Method	Prec	Rec	F-measure
GIZA++ Intersect	**96.7**	53.0	68.5
GIZA++ Union	82.5	69.0	75.1
GIZA++ GDF	84.0	68.2	75.2
Phrasal ITG	50.7	**80.3**	62.2
Phrasal ITG + NCC	75.4	78.0	**76.7**

Following the lead of (Fraser and Marcu, 2006), we hand-aligned the first 100 sentence pairs of our training set according to the Blinker annotation guidelines (Melamed, 1998). We did not differentiate between sure and possible links. We report precision, recall and balanced F-measure (Och and Ney, 2003). For comparison purposes, we include the results of three types of GIZA++ combination, including the grow-diag-final heuristic (GDF). We tested our phrasal ITG with fixed link pruning, and then added the non-compositional constraint (NCC). During development we determined that performance levels off for both of the ITG models after 3 EM iterations. The results are shown in Table 2.

The first thing to note is that GIZA++ Intersection is indeed very high precision. Our confidence in it as a constraint is not misplaced. We also see that both phrasal models have significantly higher recall than any of the GIZA++ alignments, even higher than the permissive GIZA++ union. One factor contributing to this is the phrasal model's use of cepts: it completely interconnects any phrase pair, while GIZA++ union and GDF may not. Its global view of phrases also helps in this regard: evidence for a phrase can be built up over multiple sentences. Finally, we note that in terms of alignment quality, the non-compositional constraint is an unqualified success for the phrasal ITG. It produces a 25 point improvement in precision, at the cost of 2 points of recall. This produces the highest balanced F-measure observed on our test set, but the utility of its alignments will depend largely on one's desired precision-recall trade-off.

5.3 Translation Experiments

In this section, we compare a number of different methods for phrase table generation in a French to English translation task. We are interested in answering three questions:

1. Does the phrasal ITG improve on the C-JPTM?
2. Can phrasal translation models outperform the surface heuristic?
3. Do Viterbi phrasal alignments provide better input for the surface heuristic?

With this in mind, we test five phrase tables. Two are conditionalized phrasal models, each EM trained until performance degrades:

- C-JPTM[3] as described in (Birch et al., 2006)
- Phrasal ITG as described in Section 4.1

Three provide alignments for the surface heuristic:

- GIZA++ with grow-diag-final (GDF)
- Viterbi Phrasal ITG with and without the non-compositional constraint

We use the Pharaoh decoder (Koehn et al., 2003) with the SMT Shared Task baseline system (Koehn and Monz, 2006). Weights for the log-linear model are set using the 500-sentence tuning set provided for the shared task with minimum error rate training (Och, 2003) as implemented by Venugopal and Vogel (2005). Results on the provided 2000-sentence development set are reported using the BLEU metric (Papineni et al., 2002). For all methods, we report performance with and without IBM Model 1 features (M1), along with the size of the resulting tables in millions of phrase pairs. The results of all experiments are shown in Table 3.

Table 3: Translation Comparison.

Method	BLEU	+M1	Size
Conditionalized Phrasal Model			
C-JPTM	26.27	28.98	1.3M
Phrasal ITG	28.85	30.24	2.2M
Alignment with Surface Heuristic			
GIZA++ GDF	30.46	30.61	9.8M
Phrasal ITG	30.31	30.39	5.8M
Phrasal ITG + NCC	**30.66**	**30.80**	9.0M

We see that the Phrasal ITG surpasses the C-JPTM by more than 2.5 BLEU points. A large component of this improvement is due to the ITG's use of inside-outside for expectation calculation, though there are other differences between the two systems.[4] This improvement over search and sampling is demonstrated by the ITG's larger table size; by exploring more thoroughly, it is extracting more phrase pairs from the same amount of data. Both systems improve drastically with the addition of IBM Model 1 features for lexical preference. These features also narrow the gap between the two systems. To help calibrate the contribution of these features, we parameterized the ITG's phrase table using only Model 1 features, which scores 27.17.

Although ITG+M1 comes close, neither phrasal model matches the performance of the surface heuristic. Whatever the surface heuristic lacks in sophistication, it makes up for in sheer coverage, as demonstrated by its huge table sizes. Even the Phrasal ITG Viterbi alignments, which over-commit wildly and have horrible precision, score slightly higher than the best phrasal model. The surface heuristic benefits from capturing as much context as possible, while still covering smaller translation events with its flat counts. It is not held back by any lexicon constraints. When GIZA++ GDF+M1 is forced to conform to a lexicon constraint by dropping any phrase with a frequency lower than 5 from its table, it scores only 29.26, for a reduction of 1.35 BLEU points.

Phrases extracted from our non-compositional Viterbi alignments receive the highest BLEU score, but they are not significantly better than GIZA++ GDF. The two methods also produce similarly-sized tables, despite the ITG's higher recall.

[3] Supplied by personal communication. Run with default parameters, but with maximum phrase length increased to 5.

[4] Unlike our system, the Birch implementation does table smoothing and internal lexical weighting, both of which should help improve their results. The systems also differ in distortion modeling and $\emptyset$ handling, as described in Section 3.

6 Conclusion

We have presented a phrasal ITG as an alternative to the joint phrasal translation model. This syntactic solution to phrase modeling admits polynomial-time training and alignment algorithms. We demonstrate that the same consistency constraints that allow joint phrasal models to scale also dramatically speed up ITGs, producing an 80-times faster inside-outside algorithm. We show that when used to learn phrase tables for the Pharaoh decoder, the phrasal ITG is superior to the constrained joint phrasal model, producing tables that result in a 2.5 point improvement in BLEU when used alone, and a 1 point improvement when used with IBM Model 1 features. This suggests that ITG's perfect expectation counting does matter; other phrasal models could benefit from either adopting the ITG formalism, or improving their sampling heuristics.

We have explored, for the first time, the utility of a joint phrasal model as a word alignment method. We present a non-compositional constraint that turns the phrasal ITG into a high-recall phrasal aligner with an F-measure that is comparable to GIZA++.

With search and sampling no longer a concern, the remaining weaknesses of the system seem to lie with the model itself. Phrases are just too efficient probabilistically: were we to remove all lexicon constraints, EM would always align entire sentences to entire sentences. This pressure to always build the longest phrase possible may be overwhelming otherwise strong correlations in our training data. A promising next step would be to develop a prior over lexicon size or phrase size, allowing EM to introduce large phrases at a penalty, and removing the need for artificial constraints on the lexicon.

Acknowledgments Special thanks to Alexandra Birch for the use of her code, and to our reviewers for their comments. The first author is funded by Alberta Ingenuity and iCORE studentships.

References

A. Birch, C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *HLT-NAACL Workshop on Statistical Machine Translation*, pages 154–157.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

C. Cherry and D. Lin. 2006. A comparison of syntactically motivated word alignment spaces. In *EACL*, pages 145–152.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *ACL*, pages 769–776.

P. Koehn and C. Monz. 2006. Manual and automatic evaluation of machine translation. In *HLT-NACCL Workshop on Statistical Machine Translation*, pages 102–121.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*, pages 127–133.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistic machine translation. In *EMNLP*, pages 133–139.

I. D. Melamed. 1998. Manual annotation of translational equivalence: The blinker project. Technical Report 98-07, Institute for Research in Cognitive Science.

I. D. Melamed. 2003. Multitext grammars and synchronous parsers. In *HLT-NAACL*, pages 158–165.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

A. Venugopal and S. Vogel. 2005. Considerations in maximum mutual information and minimum classification error training for statistical machine translation. In *EAMT*.

J. M. Vilar and E. Vidal. 2005. A recursive statistical translation model. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 199–207.

S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhang, and A. Waibel. 2003. The CMU statistical machine translation system. In *MT Summmit*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

H. Zhang and D. Gildea. 2004. Syntax-based alignment: Supervised or unsupervised? In *COLING*, pages 418–424.

H. Zhang and D. Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *ACL*, pages 475–482.

H. Zhang, L. Huang, D. Gildea, and K. Knight. 2006. Synchronous binarization for machine translation. In *HLT-NAACL*, pages 256–263.