Learning Translations for Tagged Words: Extending the Translation Lexicon of an ITG for Low Resource Languages

Markus Saers and **Dekai Wu**
Human Language Technology Center
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
HKUST, Clear Water Bay, Kowloon, Hong Kong
{masaers|dekai}@cs.ust.hk

Abstract

We tackle the challenge of learning part-of-speech classified translations as part of an inversion transduction grammar, by learning translations for English words with known part-of-speech tags, both from existing translation lexica and from parallel corpora. When translating from a low resource language into English, we can expect to have rich resources for English, such as treebanks, and small amounts of bilingual resources, such as translation lexica and parallel corpora. We solve the problem of integrating these heterogeneous resources into a single model using stochastic Inversion Transduction Grammars, which we augment with wildcards to handle unknown translations.

1 Introduction

We introduce an augmentation to Inversion Transduction Grammars, or ITGs (Wu, 1997), that allow us to under specify the translation of lexical rules, and defer to observed usage to decide, within the syntactic context of the lexical rule, how the wildcard should be instantiated. Having specific wildcard rules instead of just instantiating all possible translations for the sentence pair at hand (a) allows us to use it as a back-off: we can limit the use of these spurious rules to the circumstances when we have no other choice, and (b) allows us to explicitly reserve some probability mass for the unknown translations of a lexical unit, which also gives us a hint about how certain we are about the known translations. This allows us to say things like "we know that twelve is a cardinal number but it's not in our translation lexicon, let's see what it is translated to in the parallel corpus". Whith small amounts of data, it is imperative to get the structural generalizations right in order to get adequate statistics; there simply aren't enough examples of longer chunks to get reliable counts. This approach allows us to make good use of the limited parallel resources that are available in a way that traditional statistical machine translation systems are unable to; and to make good use of the human translation examples in a way that traditional rule-based machine translation systems are unable to. Even statistical tree-based models typically require word alignments as input, which again requires large amounts of parallel data to learn well. In contrast, Inversion Transduction Grammars learn the translation model and the word alignments simultaneously, effectively integrating over all possible alignments during training. This is possible because of the strong modeling bias, that limits the search space of possible compositions to make the seemingly intractable problem of bilingual composition tractable, without resorting to heuristics. The modeling bias has been empirically shown to still allows the model to express most structural differences that have been observed between natural languages. Taken together, this lets us mostly rely on what we already know, and incorporate new knowledge as we encounter the need to. This is ideal in the low resource language setting, where we want to stick to presumably hard earned prior knowledge when possible, but still have the option of adding to this knowledge base when needed.

The low-resource language translation setting presumes that we have small amounts of resources in

Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP, pages 55–64,
San Diego, California, June 17, 2016.

the input language, and large amounts of resources in the output language (English). The job of a translation system is to produce fluent output that adequately represents the meaning of the input, so a translation system should be biased towards the output language. We do this by basing our ITG model on an English treebank, which allows us to (a) extract a binarized context-free grammar, CFG, and (b) estimate initial probabilities for the structural rules. The stochastic CFG can then be mirrored to form a grammatical channel model (Wu and Wong, 1998).

Conventional statistical machine translation, or SMT, systems such as phrase-based SMT rely on large amounts of parallel data to collect statistics over how large chunks translate between two languages. These models are highly specific, and may have two different rules for example, for a long and complicated noun phrase with the determiner and for the very same noun phrase without it. Needless to say this kind of modeling is too wasteful to be of much use when there is very small amounts of parallel data available. Tree-based models, models that allow for chunks containing general categories as opposed to fully lexicalized chunks, are better able to generalize, but make poor use of the data by disregarding translations that fall outside of the single monolingually motivated parse tree that has been committed to. A forest-based system would alleviate this problem, but two problems still remain: The conventional systems all take a preexisting word alignment as input, disregarding anything that does not conform to it. And none of them make use of forests in the *output* language.

In contrast to conventional systems, our proposed model jointly models the parse forest of the output language, the lexical alignment to the input language, and thus also the projection of the output language parse forest onto the input language sentence, as well as the structural differences needed to make this projection happen. All of this is possible because of ITGs: Their inductive bias allows us to disregard huge swathes of the search space when explaining the structural differences. Their similarity with CFGs allows us to use an English context-free grammar as a starting point for induction. And their computational complexity allows us to efficiently collect rule expectations.

The presented model is capable of exploring the entire space of structural differences between, in our case, English and Chinese, that conform to (a) the English CFG, and (b) the ITG-constraints. It is also capable of expanding the translation dictionary beyond what we initialize it with to cover the small parallel corpus that we assume is available. Although Chinese is by no means a low resource language, it is vastly different from English, meaning that there are plenty of structural differences to learn. We also allow very limited Chinese resources in building our model, simulating a low resource language setting.

Inversion transductions are formally a family of transductions, where a transduction is the bilingual version of a formal language, such that it relates two formal languages to each other. Inversion transductions are generated by Inversion Transduction Grammars, or ITGs (Wu, 1997), which share several traits with CFGs in that the transduction rules have single non-terminals on the left-hand side, and in that there is always a 2-normal form equivalence for every ITG. The latter is quite rare for transduction grammars, and limits the structural differences that can be generated between the languages. These limits in structural differences have been empirically shown to include most of the differences found between natural languages (Søgaard and Wu, 2009), and make efficient processing possible.

Formally, an ITG is a tuple $\langle \mathcal{N}, \mathcal{W}_0, \mathcal{W}_1, \mathcal{R}, S \rangle$, where $\mathcal{N}$ is a finite nonempty set of nonterminals, $\mathcal{W}_0$ is a finite set of terminals in the output language L_0, $\mathcal{W}_1$ is a finite set of terminals in the input language L_1, $\mathcal{R}$ is a finite nonempty set of inversion transduction rules and $S \in \mathcal{N}$ is a designated start symbol. An inversion transduction rule is restricted to take one of the following forms:

$$S \to [A]\,, \quad A \to [\varphi^+]\,, \quad A \to \langle \varphi^+ \rangle$$

where $S \in \mathcal{N}$ is the start symbol, $A \in \mathcal{N}$ is a nonterminal, and φ^+ is a nonempty sequence of nonterminals and biterminals. A biterminal is a pair: $\mathcal{W}_0^* \times \mathcal{W}_1^*$, where at least one of the strings have to be nonempty. The square and angled brackets signal straight and inverted order respectively. The brackets are frequently left out when there is only one element on the right-hand side.

The ITG 2-normal form is analogous to the Chomsky normal form for CFGs, where the rules are fur-

ther restricted to only the following forms:

$$S \rightarrow A, \quad A \rightarrow [BC], \quad A \rightarrow \langle BC \rangle,$$
$$A \rightarrow e/f, \quad A \rightarrow e/\epsilon, \quad A \rightarrow \epsilon/f$$

where $S \in \mathcal{N}$ is the start symbol, $A, B, C \in \mathcal{N}$ are nonterminals, $e \in \mathcal{W}_0$ is an L_0 token, $f \in \mathcal{W}_1$ is an L_1 token, and ϵ is the empty token.

A bracketing ITG, or BITG, has only one nonterminal symbol (other than the dedicated start symbol), which means that the nonterminals carry no information at all other than the fact that their yields are discrete unit.

2 Related work

The use of structure as input to the training of a statistical model in machine translation was pioneered by Yamada and Knight (2001), where they extend the IBM model 1 (Brown *et al.*, 1993) to incorporate syntactic features derived from a parse tree on the output language (the input to the noisy channel, but the output of the decoder). The generative story of the model is that an English parse tree has the children of its nodes reordered, gets the option to insert a foreign token to the left or right of any node, and finally have all the English leaf nodes translated. Reading the leaf nodes of the tree in order yields the generated foreign sentence. Being a generative model, it is straight forward to train using EM, which they do. Manual evaluation shows that the word alignments corresponding to the one-best derivation are better than those of IBM model 5, and that they managed to get 10 out of the evaluated 50 sentence pairs perfectly aligned, whereas IBM model 5 got none. There are two key differences between Yamada and Knight (2001) and out model: (a) Their model describes how a foreign sentence is generated from an English parse tree, our model describes how sentence pairs are jointly generated. And (b) their model requires committing to a single English parse tree, our model jointly parses and aligns the sentences, effectively integrating out all parse trees that our grammar allows for the English sentence.

Perhaps the most prolific translation model that involves trees is Galley *et al.* (2004), which learns very complex rules such as $(\text{ne VB pas}) \rightarrow (\text{VP } (\text{AUX does}) \ (\text{RB not}) \ x_2)$ where x_2 is a variable binding to the second element in the left-hand side. The method takes a parallel sentence pair where one of the sentences has been parsed, and a word alignment, and produces, for each observed word aligned sentence pair, the minimal set of rules to explain it. This method allows complicated rules to be extracted, and different feature scores to be calculated for the extracted rules. This does, however, come at the cost of not being able to optimize the model in any meaningful way. Instead, one has to resort to tuning the feature weights used by the decoder. Any mistakes made by the parser or the automatic word aligner are incorporated into the model without any recourse. In contrast, our model jointly parses the output language and aligns the sentences, within a generative model that can be optimized globally across the entire training data.

It is possible to alleviate the mismatch between given alignments and parse trees. Riesa and Marcu (2010) formulate a discriminative k-best alignment model with hundreds of features that can be used to choose the best alignment that matches a given tree. The method as presented require some analysis on both input and output language for the feature model, as well as a set of hand aligned sentences for training; neither of which are readily available for low resource languages. Similarly, DeNero and Klein (2007) describe a model to tailor word alignments to existing syntactic trees; in our model, we instead consider all possible trees allowed by a CFG. Burkett *et al.* (2010) and Burkett and Klein (2012) take the opposite approach and alter the trees to fit the alignments, as presented, their approach requires parallel treebanks to train on, which cannot be expected to be available when translating to or from a low resource language.

It is also possible to learn translation rules focusing on the syntax of the input language rather than output language. Huang *et al.* (2006) turn the tables, and learn translation rules with the input language parsed. Their approach forces the decoder to commit to a single input tree and then build possible output *strings* rather than building possible output *trees*. The model has also been generalized so that rules can be extracted from input forests rather than single trees (Mi and Huang, 2008), and the decoder has been extended to accept forests as input (Mi *et al.*, 2008). The rule extraction process still requires a word alignment to be provided. Since our model is

based on a grammar rather than parse trees, it essentially integrates out all trees as well as the alignments needed to perform the rule extraction. There is a difference in that we have an output language grammar, but we still implicitly build input language forests.

Our model is related to the grammatical channel model of Wu and Wong (1998) in that the way we set up our initial grammar is similar, but where they then proceed to translate directly with it, induction has just started with our model. Key differences is that they merely mirror structural rules, allowing them to be completely straight or completely inverted, and that they make no effort to extend the translation dictionary. In contrast, we allow long rules to be broken down, which allows for more differences in word order, and induce additional lexical translations that are helpful in explaining the training data.

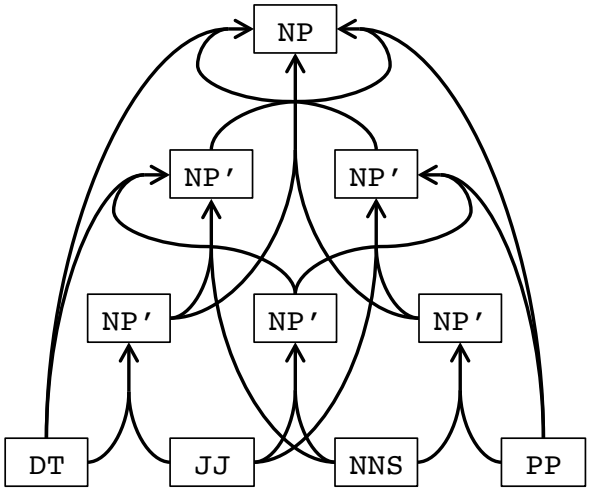

Figure 1: All possible binarization of a structural rule with four constituents. Any valid hyperpath from the four leaves to the root constitute a valid binarization.

3 Induction Algorithm

The main focus of this paper is to learn translations of known English lexical units. Our method relies on being able to realize when it lacks the translation needed to use an English lexical unit in a specific syntactic context, and to find viable candidate translations from the observed sentence pair. The model knows which English tokens go where (from the treebank), and can be expected to have some translation vocabulary, mainly of content words (from the translation lexicon). It can easily realize when it lacks an applicable translation, and if there is an applicable wildcard-rule we allow it to hypothesize a realization of that wildcard from the current situation. This is a good mechanism for extending the translation lexicon for content words, but it requires the biparser to determine the syntactic context that this hypothesizing takes place in. This in turn requires the English structural rules to be adapted so that they can handle Chinese as well. To do this, we hypothesize that there are two main differences between English and Chinese: word order and function words. It makes sense for most content words to have some kind of correspondence in the other language, but they rarely occur in the same order. The function words, on the other hand are sometimes realized in both languages, but frequently one language will use a function word, whereas the other language will ignore that distinction, or realize it through word order or in some other syntactic way. An example of the former is the Chinese question particle (吗), which is realized in English through the syntactic question construction. An example of the latter is determinate case in English, which is simply not realized in Chinese, meaning that the word the needs to be intelligently inserted when translating from Chinese to English.

To account for the word order differences, we binarize the structural rules, allowing each binary rule to be either straight or inverted. This allows the model to account for any word order difference within the ITG-constraints. Since the model is already heavily biased towards English, allowing English words to not translate into anything takes care of the case where there is no Chinese equivalence of an English function word. For the Chinese function words, there either is an English equivalence, in which case the mechanism for content words will find it, or there isn't, in which case we simply allow the biparser to algorithmically skip these words.

3.1 Step 1

Like Wu and Wong (1998) we start with an English grammar and a translation dictionary. But, where they allow structural rules to be either completely straight or completely inverted, we allow *any binarization* of the structural rules to be either straight or inverted. And, where they allow any English word

Table 1: The result of converting the two English CFG rules JJ → *nice* and NP → DT JJ NNS PP to ITG rules using our proposed method and that of Wu and Wong (1998). Clubs are wild, and NP′ is the auxiliary symbol for NP.

Our method			**Wu and Wong (1998)**
JJ → *nice*/和藹	NP → [DT NP′]	NP → ⟨DT NP′⟩	
JJ → *nice*/可亲	NP → [NP′ NP′]	NP → ⟨NP′ NP′⟩	
JJ → *nice*/♣	NP → [NP′ PP]	NP → ⟨NP′ PP⟩	
	NP′ → [DT NP′]	NP′ → ⟨DT NP′⟩	JJ → *nice*/和藹
	NP′ → [JJ NP′]	NP′ → ⟨JJ NP′⟩	JJ → *nice*/可亲
	NP′ → [NP′ PP]	NP′ → ⟨NP′ PP⟩	NP → [DT JJ NNS PP]
	NP′ → [NP′ NNS]	NP′ → ⟨NP′ NNS⟩	NP → ⟨DT JJ NNS PP⟩
	NP′ → [DT JJ]	NP′ → ⟨DT JJ⟩	
	NP′ → [JJ NNS]	NP′ → ⟨JJ NNS⟩	
	NP′ → [NNS PP]	NP′ → ⟨NNS PP⟩	

to translate into whatever the translation dictionary mandates, we additionally assume that any English word can translate into a wildcard token that represents the translations not in the dictionary. When binarizing rules, there are many approaches one could take. It is possible to "box" nonterminals to produce an exactly equivalent binary grammar. Boxing means replacing any right-hand side occurrence of a sequence, eg. DT NN with a single boxed nonterminal $\overline{\text{DT NN}}$, which deterministically expand to the original sequence with the rule $\overline{\text{DT NN}}$ → DT NN. This tends to expand the set of nonterminals needlessly, and differentiating, for example, between the case when a noun phrase contains three versus four nouns is counter-productively specific. Instead, we opt to have each left-hand side symbol be associated with one auxiliary symbol that handles the binarization of that category, and represents a fragment of it. We essentially generate the entire parse forest rooted in the left-hand side, with the right-hand side symbols as the leaves, and every internal node labeled with the auxiliary symbol (see Figure 1). From this parse forest of the rule, we can extract all binary rules, and add a straight and an inverted version of them to our ITG. This allows us to account for any permutation of the right-hand sides that fall within the ITG-constraints. Table 1 shows the result of applying our method as opposed to applying the method of Wu and Wong (1998) to the two English CFG-rules NP → DT JJ NNS PP and JJ → *nice*.

3.2 Step 2

At this point our model has too many structural rules and too few lexical rules, so the next step is to root out the superfluous structural rules and build out the coverage of the lexical rules. We do this by reestimating the ITG using a small parallel corpus and variational Bayes; to weed out unnecessary structural rules we use a sparse prior, and to build out the lexical coverage, we hypothesis that the wild card will be binding predominantly to valid translations of the associated English word, and instantiate the observed bindings into the ITG.

Variational Bayes is mechanically similar to expectation maximization (Dempster *et al.*, 1977) with inside-outside (Lari and Young, 1990), but optimizes the maximum *a posteriori* probability of the model given the data and a prior, rather than the maximum likelihood of the model given the data. Intuitively it can be understood as collecting the fractional counts, or expectations, from the data and discounting them before maximizing. Given that we have rule expectations from inside-outside $E(A \to \varphi)$, the reestimated probability of a rule is:

$$p(A \to \varphi) = \frac{e^{\psi\left(E(A\to\varphi)+\alpha_{A\to\varphi}\right)}}{\sum_{\phi} e^{\psi\left(E(A\to\phi)+\alpha_{A\to\phi}\right)}}$$

where ψ is the digamma function, and $\alpha_{A\to\varphi}$ is the prior of the rule $A \to \varphi$ (Kurihara and Sato, 2006). It has previously been used for ITG learning (Zhang *et al.*, 2008; Saers and Wu, 2013), but only with uniform priors. In this paper, we distinguish three different classes of rules, and assign them different pri-

ors: *preexisting lexical rules*, *lexical rules with wildcards*, and *other rules*. We want the preexisting lexical rules to have a relatively high prior, since we trust our translation dictionary. We may or may not want to keep the wildcard rules, and having a separate prior for them allows us to easily choose their fate. All other rules need to earn their keep, and are thus subjected to a sparse prior.

The lexical coverage is extended by hypothesizing that what the wildcard binds to in the expectation collection phase constitutes a valid translation. This is an iterative step, where we first biparse the parallel corpus with the previous ITG. The wildcard is allowed to match a preset number of Chinese tokens when no known translation applies, and we also allow skipping both English and Chinese tokens as a backoff. Once a sentence pair has been processed, we inspect the parse forest and extract all bindings for all wildcards and instantiate the corresponding rules with half of the fractional counts it would have had if it existed. The other half of the fractional counts is retained with the wildcard rule.

The biparser we have is based on Saers *et al.* (2009), with the addition of wildcard matching and skipping. The algorithm is very suitable for our needs, as it approximates the search to bring down the time complexity of collecting the fractional counts from $O(n^6)$to $O(bn^3)$, where n is proportional to the length of the sentences in the pair, and b is the width of the search beam (the wider the beam, the more accurate the approximation). We implement skipping as implicit, low probability rules on the following forms:

$$A \to [A\ \epsilon/f]\,, \quad A \to [\epsilon/f\ A]\,,$$
$$A \to [A\ e/\epsilon]\,, \quad A \to [e/\epsilon\ A]$$

This allows the parser to maintain the category, but consume a foreign or English token adjacent to a known constituent. The low probability makes the parser avoid skipping if possible.

4 Experimental Setup

To test the induction algorithm, we empirically compare the results of our proposed system to a bracketing ITG induced from the same parallel corpus, but without any prior knowledge of English. To extract a CFG over English, we use the Penn treebank (Marcus *et al.*, 1993), with relative frequencies of the productions as the rule probabilities. As translation dictionary, we use the *Chinese–English Translation Lexicon*(Huang and Graff, 2002). When transforming the CFG into an ITG (Section 3.1) we divide the probability mass of the CFG-rules uniformly among the ITG rules they spawn.

As the small parallel data set we use the IWSLT07 Chinese–English data set (Fordyce, 2007), which contains 46,867 sentence pairs. Chinese sentence are typically written without spaces, so we use a tool (Wu, 1999) to segment it into more "word like" units. We allow the wildcard to match zero or one such Chinese tokens. To make use of the parallel data, we perform 10 iterations of reestimation (Section 3.2), with a beam width $b = 100$, and the following priors: preexisting lexical rules: 10^{-2}, lexical rules with wildcards: 10^{-5}, other rules: 10^{-10}. We qualitatively evaluate the resulting ITG by looking at how it explains some of the parallel sentences, as opposed to how the baseline bracketing ITG explains them.

5 Results

We evaluate the results qualitatively, by comparing the output of our induced system with that of the baseline BITG. Figures 2 and 4 contain the output of our system, and Figures 3 and 5 contain the output of the baseline BITG. All Figures have the same structure: the English sentence on the top; the Chinese sentence on the left; a compositional alignment matrix between them where black boxes represent correct terminals, gray boxes represent incorrect terminals and outlined boxes represent the compositions; and the corresponding ITG parse tree over the English sentence.

Figure 2 shows a single noun phrase utterance (room fifty-six twelve. translating into 五六一二房。 'five six one two room.') taken from our training data, as biparsed by our induced ITG. The tokens align well, which would not have been possible if the Chinese segmenter hadn't grouped 一 'one' and 二 'two' together as a single token for twelve to align with. It also aligns fifty to 五 'five', which is technically wrong, but may be useful when different languages prefer to group numbers differently. Notice that the noun phrase needs to separate the number series from the noun (room) in order to flip the or-

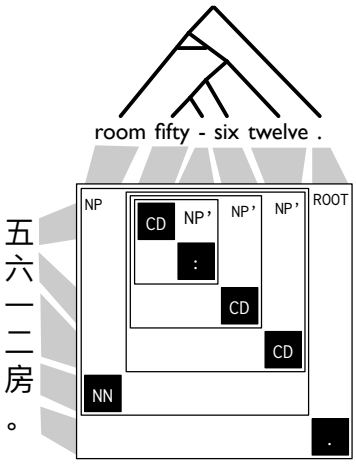

Figure 2: The tree structure imposed on one of the training sentences by our induced ITG.

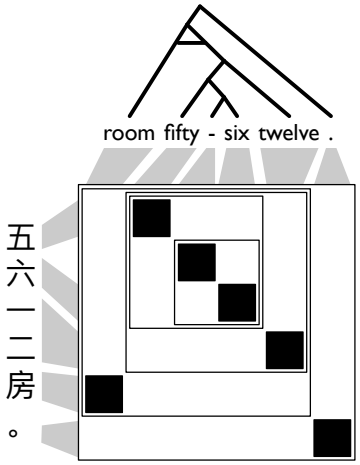

Figure 3: The tree structure imposed on one of the training sentences by the baseline bracketing ITG.

der in Chinese. It has also chosen to make the NP′ a right-heavy tree, when it could very well have made it left-heavy or balanced. This is desirable, since it appears to have generalized arbitrarily long chains of numbers, so what it has learned is that a noun-phrase fragment can consist of a noun-phrase fragment and a number. Inspecting the grammar, we find the following relevant rules:

$$\text{NP}' \xrightarrow{6.4\%} [\text{CD NP}'] \quad \text{NP}' \xrightarrow{4.8\%} \langle \text{CD NP}' \rangle$$
$$\text{NP}' \xrightarrow{1.4\%} [\text{NP}' \text{ CD}] \quad \text{NP}' \xrightarrow{1.2\%} \langle \text{NP}' \text{ CD} \rangle$$

There is a clear bias in favor of right-heavy trees (the first two rules account for about 11.2% of noun-phrase fragments, as opposed to 2.6% which are left-heavy), as well as a bias for maintaining the order of number sequences in noun-phrase fragments (7.8% as opposed to 6%).

The bracketing ITG does not know what to do in these circumstances. As Figure 3 shows, the number sequence is arbitrarily nested. It is possible to force an ITG to always prefer a canonical left- or right-heavy sequence of straight (or inverted) rule applications, but that is an external constraint put on it manually. We managed to induce that preference automatically.

The second sentence pair is the question how long does it take to reach Japan? with its Chinese translation 到日本要花多长时间? 'reach Japan need spend how long time?'. Notice that this is a very hard translation to recreate for an automatic system. English mandates a subject in well-formed clauses, Chinese does not; English requires an auxiliary verb in questions, Chinese does not; English requires an infinitive marker, Chinese does not; English allows time to be implicit from how long, Chinese requires it to be explicit.

Figure 4 shows our system, and Figure 5 shows the BITG. Our system has learned that English has a mandatory subject, which Chinese lacks, as is evident from it begin correctly classified as a pronoun, and translated into nothing, whereas the BITG has it incorrectly translated as 花 'spend'. Our system has incorrectly aligned does to 多 'how', whereas the BITG has it incorrectly aligned it to 时间 'time'. Our system has correctly classified to as an infinitive marker as opposed to a preposition, but aligned it with 到 'reach', whereas the BITG has is correctly aligned to nothing. It is, however worth noting that our system has managed to compose to reach Japan as a unit, whereas the BITG has no concept of an infinitive verb at all. The treebank we extract our grammar from has no specific category for infinitive verb phrases either, but our system has repurposed the S-7 category (that occurs in the treebank, but not it's annotation guidelines) to represent infinitive verb phrases. Both systems have incorrectly aligned 时间 'time' to something.

Looking at the nestings, it becomes clear that the BITG has done a good job of nailing the token-to-token correspondences, but that the structure is all wrong. It is a left-heavy tree where one would ex-

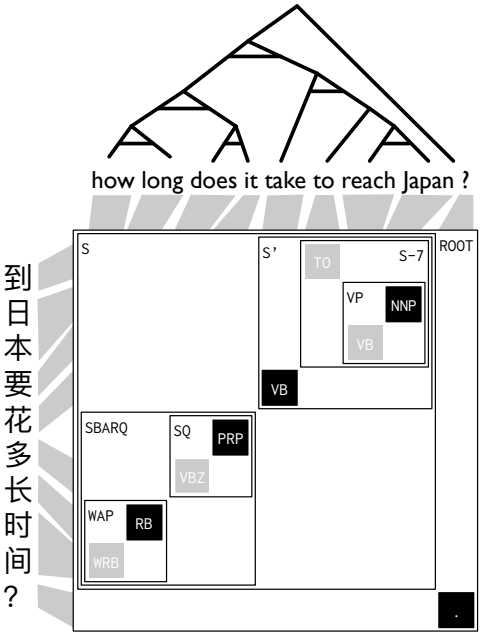

Figure 4: The tree structure imposed on one of the training sentences by our induced ITG.

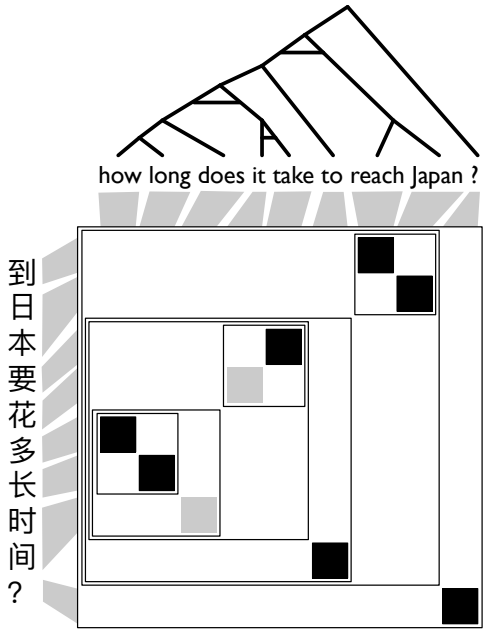

Figure 5: The tree structure imposed on one of the training sentences by the baseline bracketing ITG.

pect a mostly right-heavy tree. Our system does a better job with the nesting, although one would have wished to see take to reach Japan be dominated by how long does it, rather than being on the same level. The spans are also reasonably well labeled.

6 Conclusions

We have presented a novel approach to low-resource language translation that relies on extending the lexical coverage of a linguistically informed inversion transduction grammar. The linguistic information is derived from an output language treebank, and extending the translation lexicon requires the correct translation for a word with a specific part-of-speech tag to be identified. We have shown a way of doing precisely that through setting aside part of the probability mass for unknown translations, represented as wildcards in the ITG. Staying within the formalism of ITGs is desirable, as they can be trained directly on parallel data, without the need to first induce, and commit to, a single word alignment. The inductive bias of the ITG formalism allows us to integrate over all relevant word alignments in polynomial time, while still being able to capture most of the structural variation observed between human languages. It also allows us to combine multiple heterogenous sources of knowledge, in this paper we used (a) a context-free grammar derived from an English treebank, (b) a Chinese–English translation dictionary, and (c) a small Chinese–English parallel corpus, which is imperative in a low-resource language setting, where all available resources need to be utilized.

Acknowledgements

This material based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under LORELEI contract HR0011-15-C-0114, BOLT contracts HR0011-12-C-0014 and HR0011-12-C-0016, and GALE contracts HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the Horizon 2020 grant agreement 645452 (QT21) and FP7 grant agreement 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF16210714, GRF16214315, GRF620811 and GRF621008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

References

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

David Burkett and Dan Klein. Transforming trees to improve syntactic convergence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 863–872, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

David Burkett, John Blitzer, and Dan Klein. Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, pages 127–135, Los Angeles, California, June 2010.

Arthur Pentland Dempster, Nan M. Laird, and Donald Bruce Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

John DeNero and Dan Klein. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

C. S. Fordyce. Overview of the IWSLT 2007 evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT 2007)*, pages 1–12, 2007.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, May 2004.

Shudong Huang and David Graff. Chinese–English translation lexicon version 3.0 LDC2002L27, 2002.

Liang Huang, Kevin Knight, and Aravind Joshi. Statistical syntax-directed translation with extended domain of locality. In *7th Biennial Conference Association for Machine Translation in the Americas (AMTA 2006)*, pages 66–73, Boston, Massachusetts, 2006.

Kenichi Kurihara and Taisuke Sato. Variational bayesian grammar induction for natural language. In *Proceedings of the 8th International Conference on Grammatical Inference: Algorithms and Applications*, ICGI'06, pages 84–96, Berlin, Heidelberg, 2006. Springer-Verlag.

Karim Lari and Steve J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 4(1):35–56, 1990.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

Haitao Mi and Liang Huang. Forest-based translation rule extraction. In *2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 206–214, Honolulu, Hawaii, October 2008.

Haitao Mi, Liang Huang, and Qun Liu. Forest-based translation. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 192–199, Columbus, Ohio, June 2008.

Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 157–166, Uppsala, Sweden, July 2010.

Markus Saers and Dekai Wu. Bayesian induction of bracketing inversion transduction grammars. In *Sixth International Joint Conference on Natural Language Processing (IJCNLP2013)*, pages 1158–1166, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

Markus Saers, Joakim Nivre, and Dekai Wu. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *11th International Conference on Parsing Technologies (IWPT'09)*, pages 29–32, Paris, France, October 2009.

Anders Søgaard and Dekai Wu. Empirical lower bounds on translation unit error rate for the full

class of inversion transduction grammars. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 33–36, Paris, France, October 2009.

Dekai Wu and Hongsing Wong. Machine translation with a stochastic grammatical channel. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL '98)*, volume 2, pages 1408–1415, Montreal, Quebec, August 1998.

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

Zhibiao Wu. LDC Chinese segmenter, 1999.

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *39th Annual Meeting of the Association for Computational Linguistics and 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July 2001.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 97–105, Columbus, Ohio, June 2008.