Learning Hierarchical Translation Structure with Linguistic Annotations

Markos Mylonakis
ILLC
University of Amsterdam
m.mylonakis@uva.nl

Khalil Sima'an
ILLC
University of Amsterdam
k.simaan@uva.nl

Abstract

While it is generally accepted that many translation phenomena are correlated with linguistic structures, employing linguistic syntax for translation has proven a highly non-trivial task. The key assumption behind many approaches is that translation is guided by the source and/or target language parse, employing rules extracted from the parse tree or performing tree transformations. These approaches enforce strict constraints and might overlook important translation phenomena that cross linguistic constituents. We propose a novel flexible modelling approach to introduce linguistic information of varying granularity from the source side. Our method induces joint probability synchronous grammars and estimates their parameters, by selecting and weighing together linguistically motivated rules according to an objective function directly targeting generalisation over future data. We obtain statistically significant improvements across 4 different language pairs with English as source, mounting up to +1.92 BLEU for Chinese as target.

1 Introduction

Recent advances in Statistical Machine Translation (SMT) are widely centred around two concepts: (a) hierarchical translation processes, frequently employing Synchronous Context Free Grammars (SCFGs) and (b) transduction or synchronous rewrite processes over a linguistic syntactic tree. SCFGs in the form of the Inversion-Transduction Grammar (ITG) were first introduced by (Wu, 1997) as a formalism to recursively describe the translation process. The Hiero system (Chiang, 2005) utilised an ITG-flavour which focused on hierarchical phrase-pairs to capture context-driven translation and reordering patterns with 'gaps', offering competitive performance particularly for language pairs with extensive reordering. As Hiero uses a single non-terminal and concentrates on overcoming translation lexicon sparsity, it barely explores the recursive nature of translation past the lexical level. Nevertheless, the successful employment of SCFGs for phrase-based SMT brought translation models assuming latent syntactic structure to the spotlight.

Simultaneously, mounting efforts have been directed towards SMT models employing linguistic syntax on the source side (Yamada and Knight, 2001; Quirk et al., 2005; Liu et al., 2006), target side (Galley et al., 2004; Galley et al., 2006) or both (Zhang et al., 2008; Liu et al., 2009; Chiang, 2010). Hierarchical translation was combined with target side linguistic annotation in (Zollmann and Venugopal, 2006). Interestingly, early on (Koehn et al., 2003) exemplified the difficulties of integrating linguistic information in translation systems. Syntax-based MT often suffers from inadequate constraints in the translation rules extracted, or from striving to combine these rules together towards a full derivation. Recent research tries to address these issues, by re-structuring training data parse trees to better suit syntax-based SMT training (Wang et al., 2010), or by moving from linguistically motivated synchronous grammars to systems where linguistic plausibility of the translation is assessed through additional features in a phrase-based system (Venugopal et al., 2009; Chiang et al., 2009), obscuring the impact of higher level syntactic processes.

While it is assumed that linguistic structure does correlate with some translation phenomena, in this

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 642–652,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

work we do not employ it as the backbone of translation. In place of linguistically *constrained* translation imposing syntactic parse structure, we opt for linguistically *motivated* translation. We learn latent hierarchical structure, taking advantage of linguistic annotations but shaped and trained for translation.

We start by labelling each phrase-pair span in the word-aligned training data with multiple linguistically motivated categories, offering multi-grained abstractions from its lexical content. These phrase-pair label charts are the input of our learning algorithm, which extracts the linguistically motivated rules and estimates the probabilities for a stochastic SCFG, without arbitrary constraints such as phrase or span sizes. Estimating such grammars under a Maximum Likelihood criterion is known to be plagued by strong overfitting leading to degenerate estimates (DeNero et al., 2006). In contrast, our learning objective not only avoids overfitting the training data but, most importantly, learns joint stochastic synchronous grammars which directly aim at generalisation towards yet unseen instances.

By advancing from structures which mimic linguistic syntax, to learning linguistically aware latent recursive structures targeting translation, we achieve significant improvements in translation quality for 4 different language pairs in comparison with a strong hierarchical translation baseline.

Our key contributions are presented in the following sections. Section 2 discusses the weak independence assumptions of SCFGs and introduces a joint translation model which addresses these issues and separates hierarchical translation structure from phrase-pair emission. In section 3 we consider a chart over phrase-pair spans filled with source-language linguistically motivated labels. We show how we can employ this crucial input to extract and train a hierarchical translation structure model with millions of rules. Section 4 demonstrates decoding with the model by constraining derivations to linguistic hints of the source sentence and presents our empirical results. We close with a discussion of related work and our conclusions.

2 Joint Translation Model

Our model is based on a probabilistic Synchronous CFG (Wu, 1997; Chiang, 2005). SCFGs define a language over string pairs, which are generated beginning from a start symbol S and recursively expanding pairs of linked non-terminals across the two strings using the grammar's rule set. By crossing the links between the non-terminals of the two sides reordering phenomena are captured. We employ binary SCFGs, i.e. grammars with a maximum of two non-terminals on the right-hand side. Also, for this work we only used grammars with either purely lexical or purely abstract rules involving one or two non-terminal pairs. An example can be seen in Figure 1, using an ITG-style notation and assuming the same non-terminal labels for both sides.

$$\begin{aligned}
\mathrm{SBAR} &\rightarrow [\mathrm{WHNP}\ \ \mathrm{SBAR}\backslash\mathrm{WHNP}] && \text{(a)}\\
\mathrm{SBAR}\backslash\mathrm{WHNP} &\rightarrow \langle \mathrm{VP/NP}^{\mathbf{L}}\ \ \mathrm{NP}^{\mathbf{R}} \rangle && \text{(b)}\\
\mathrm{NP}^{\mathbf{R}} &\rightarrow [\mathrm{NP}\ \ \mathrm{PP}] && \text{(c)}\\
\mathrm{WHNP} &\rightarrow \mathrm{WHNP}_{\mathbf{P}} && \text{(d)}\\
\mathrm{WHNP}_{\mathbf{P}} &\rightarrow \text{which} \ /\ \text{der} && \text{(e)}\\
\mathrm{VP/NP}^{\mathbf{L}} &\rightarrow \mathrm{VP/NP}^{\mathbf{L}}_{\mathbf{P}} && \text{(f)}\\
\mathrm{VP/NP}^{\mathbf{L}}_{\mathbf{P}} &\rightarrow \text{is} \ /\ \text{ist} && \text{(g)}\\
\mathrm{NP}^{\mathbf{R}} &\rightarrow \mathrm{NP}^{\mathbf{R}}_{\mathbf{P}} && \text{(h)}\\
\mathrm{NP}^{\mathbf{R}}_{\mathbf{P}} &\rightarrow \text{the solution} \ /\ \text{die Lösung} && \text{(i)}\\
\mathrm{NP} &\rightarrow \mathrm{NP}_{\mathbf{P}} && \text{(j)}\\
\mathrm{NP}_{\mathbf{P}} &\rightarrow \text{the solution} \ /\ \text{die Lösung} && \text{(k)}\\
\mathrm{PP} &\rightarrow \mathrm{PP}_{\mathbf{P}} && \text{(l)}\\
\mathrm{PP}_{\mathbf{P}} &\rightarrow \text{to the problem} \ /\ \text{für das Problem} && \text{(m)}
\end{aligned}$$

Figure 1: English-German SCFG rules for the relative clause(s) 'which is the solution (to the problem) / der die Lösung (für das Problem) ist', [ ] signify monotone translation, ⟨ ⟩ a swap reordering.

We utilise *probabilistic* SCFGs, where each rule is assigned a conditional probability of expanding the left-hand side symbol with the rule's right-hand side. Phrase-pairs are emitted jointly and the overall probabilistic SCFG is a *joint* model over parallel strings.

2.1 SCFG Reordering Weaknesses

An interesting feature of all probabilistic SCFGs (i.e. not only binary ones), which has received surprisingly little attention, is that the reordering pattern between the non-terminal pairs (or in the case of ITGs the choice between monotone and swap expansion) are not conditioned on any other part of a derivation. The result is that, the reordering pattern with the highest probability will *always* be preferred (e.g. in the Viterbi derivation) over the rest, irrespective of lexical or abstract context. As an example, a probabilistic SCFG will always assign a higher probability to derivations swapping or monotonically translating nouns and adjectives between English and French, only depending on which of the two rules $NP \rightarrow \ [NN \ \ JJ]$, $NP \rightarrow \ \langle NN \ \ JJ\rangle$ has a higher probability. The rest of the (sometimes thousands of) rule-specific features usually added to SCFG translation models do not directly help either, leaving reordering decisions disconnected from the rest of the derivation.

While in a decoder this is somehow mitigated by the use of a language model, we believe that the weakness of straightforward applications of SCFGs to model reordering structure at the sentence level misses a chance to learn this crucial part of the translation process during grammar induction. As (Mylonakis and Sima'an, 2010) note, 'plain' SCFGs seem to perform worse than the grammars described next, mainly due to wrong long-range reordering decisions for which the language model can hardly help.

2.2 Hierarchical Reordering SCFG

We address the weaknesses mentioned above by relying on an SCFG grammar design that is similar to the 'Lexicalised Reordering' grammar of (Mylonakis and Sima'an, 2010). As in the rules of Figure 1, we separate non-terminals according to the reordering patterns in which they participate. Non-terminals such as $\mathrm{B}^{\mathbf{L}}$, $\mathrm{C}^{\mathbf{R}}$ take part only in swapping right-hand sides $\langle \mathrm{B}^{\mathbf{L}} \quad \mathrm{C}^{\mathbf{R}}\rangle$ (with $\mathrm{B}^{\mathbf{L}}$ swapping from the source side's left to the target side's right, $\mathrm{C}^{\mathbf{R}}$ swapping in the opposite direction), while non-terminals such as B, C take part solely in monotone right-hand side expansions $[\mathrm{B} \quad \mathrm{C}]$. These non-terminal categories can appear also on the left-hand side of a rule, as in rule (c) of Figure 1.

In contrast with (Mylonakis and Sima'an, 2010), monotone and swapping non-terminals do not emit phrase-pairs themselves. Rather, each non-terminal NT is expanded to a dedicated phrase-pair emitting non-terminal $\mathrm{NT}_{\mathbf{P}}$, which generates all phrase-pairs for it and nothing more. In this way, the preference of non-terminals to either expand towards a (long) phrase-pair or be further analysed recursively is explicitly modelled. Furthermore, this set of *pre-terminals* allows us to separate the higher order translation structure from the process that emits phrase-pairs, a feature we employ next.

$$\begin{array}{rr}
\mathrm{A} \rightarrow [\mathrm{B} \ \ \mathrm{C}] & \mathrm{A} \rightarrow \langle \mathrm{B}^{\mathbf{L}} \ \ \mathrm{C}^{\mathbf{R}}\rangle \\
\mathrm{A}^{\mathbf{L}} \rightarrow [\mathrm{B} \ \ \mathrm{C}] & \mathrm{A}^{\mathbf{L}} \rightarrow \langle \mathrm{B}^{\mathbf{L}} \ \ \mathrm{C}^{\mathbf{R}}\rangle \\
\mathrm{A}^{\mathbf{R}} \rightarrow [\mathrm{B} \ \ \mathrm{C}] & \mathrm{A}^{\mathbf{R}} \rightarrow \langle \mathrm{B}^{\mathbf{L}} \ \ \mathrm{C}^{\mathbf{R}}\rangle \\
\mathrm{A} \rightarrow \mathrm{A}_{\mathbf{P}} & \mathrm{A}_{\mathbf{P}} \rightarrow \alpha \ / \ \beta \\
\mathrm{A}^{\mathbf{L}} \rightarrow \mathrm{A}^{\mathbf{L}}_{\mathbf{P}} & \mathrm{A}^{\mathbf{L}}_{\mathbf{P}} \rightarrow \alpha \ / \ \beta \\
\mathrm{A}^{\mathbf{R}} \rightarrow \mathrm{A}^{\mathbf{R}}_{\mathbf{P}} & \mathrm{A}^{\mathbf{R}}_{\mathbf{P}} \rightarrow \alpha \ / \ \beta
\end{array}$$

Figure 2: Recursive Reordering Grammar rule categories; A, B, C non-terminals; α, β source and target strings respectively.

In (Mylonakis and Sima'an, 2010) this grammar design mainly contributed to model lexical reordering preferences. While we retain this function, for the rich linguistically-motivated grammars used in this work this design effectively propagates reordering preferences above and below the current rule application (e.g. Figure 1, rules (a)-(c)), allowing to learn and apply complex reordering patterns.

The different types of grammar rules are summarised in abstract form in Figure 2. We will subsequently refer to this grammar structure as Hierarchical Reordering SCFG (HR-SCFG).

2.3 Generative Model

We arrive at a probabilistic SCFG model which jointly generates source $\mathbf{e}$ and target $\mathbf{f}$ strings, by augmenting each grammar rule with a probability, summing up to one for every left-hand side. The probability of a derivation D of tuple $\langle \mathbf{e}, \mathbf{f}\rangle$ beginning from start symbol S is equal to the product of the probabilities of the rules used to recursively generate it.

We separate the structural part of the derivation D, down to the pre-terminals $\mathrm{NT}_{\mathbf{P}}$, from the phrase-emission part. The grammar rules pertaining to the

<table>
<tr><td colspan="4">X, SBAR, WHNP+VP, WHNP+VBZ+NP</td></tr>
<tr><td></td><td colspan="3">X, VBZ+NP, VP, SBAR\WHNP</td></tr>
<tr><td colspan="3">X, SBAR/NN, WHNP+VBZ+DT</td><td></td></tr>
<tr><td></td><td colspan="2">X, VBZ+DT, VP/NN</td><td></td></tr>
<tr><td colspan="2">X, WHNP+VBZ,
SBAR/NP</td><td colspan="2">X, NP,
VP\VBZ</td></tr>
<tr><td>X, WHNP,
SBAR/VP</td><td>X, VBZ,
VP/NP</td><td>X, DT,
NP/NN</td><td>X, NN,
NP\DT</td></tr>
<tr><td><b>which</b></td><td><b>is</b></td><td><b>the</b></td><td><b>problem</b></td></tr>
</table>

Figure 3: The label chart for the source fragment 'which is the problem'. Only a sample of the entries is listed.

structural part and their associated probabilities define a model $p(\sigma)$ over the latent variable σ determining the recursive, reordering and phrase-pair segmenting structure of translation, as in Figure 4. Given σ, the phrase-pair emission part merely generates the phrase-pairs utilising distributions from every $\text{NT}_{\mathbf{P}}$ to the phrase-pairs that it covers, thereby defining a model over all sentence-pairs generated given each translation structure. The probabilities of a derivation and of a sentence-pair are then as follows:

$$p(D) = p(\sigma)p(\mathbf{e}, \mathbf{f}|\sigma) \quad (1)$$

$$p(\mathbf{e}, \mathbf{f}) = \sum_{D: D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle} p(D) \quad (2)$$

By splitting the joint model in a hierarchical structure model and a lexical emission one we facilitate estimating the two models separately. The following section discusses this.

3 Learning Translation Structure

3.1 Phrase-Pair Label Chart

The input to our learning algorithm is a word-aligned parallel corpus. We consider as phrase-pair spans those that obey the word-alignment constraints of (Koehn et al., 2003). For every training sentence-pair, we also input a chart containing one or more labels for every synchronous span, such as that of Figure 3. Each label describes different properties of the phrase pair (syntactic, semantic etc.), possibly in relation to its context, or supplying varying levels of abstraction (phrase-pair, determiner with noun, noun-phrase, sentence etc.). We aim to induce a recursive translation structure explaining the joint generation of the source and target sentence taking advantage of these phrase-pair span labels.

For this work we employ the linguistically motivated labels of (Zollmann and Venugopal, 2006), albeit for the source language. Given a parse of the source sentence, each span is assigned the following kind of labels:

Phrase-Pair All phrase-pairs are assigned the X label

Constituent Source phrase is a constituent A

Concatenation of Constituents Source phrase labelled A+B as a concatenation of constituents A and B, similarly for 3 constituents.

Partial Constituents Categorial grammar (Bar-Hillel, 1953) inspired labels A/B, A\B, indicating a partial constituent A missing constituent B right or left respectively.

An important point is that we assign all applicable labels to every span. In this way, each label set captures the features of the source side's parse-tree without being bounded by the actual parse structure, as well as provides a coarse to fine-grained view of the source phrase.

3.2 Grammar Extraction

From every word-aligned sentence-pair and its label chart, we extract SCFG rules as those of Figure 2. Binary rules are extracted from adjoining synchronous spans up to the whole sentence-pair level, with the non-terminals of both left and right-hand side derived from the label names plus their reordering function (monotone, left/right swapping) in the span examined. A single unary rule per non-terminal NT generates the phrase-pair emitting $\text{NT}_{\mathbf{P}}$. Unary rules $\text{NT}_{\mathbf{P}} \rightarrow \alpha \;/\; \beta$ generating the phrase-pair are created for all the labels covering it.

While we label the phrase-pairs similarly to (Zollmann and Venugopal, 2006), the extracted grammar is rather different. We do not employ rules that are grounded to lexical context ('gap' rules), relying instead on the reordering-aware non-terminal set and related unary and binary rules. The result is a grammar which can both capture a rich array of translation phenomena based on linguistic and lexical grounds and explicitly model the balance between

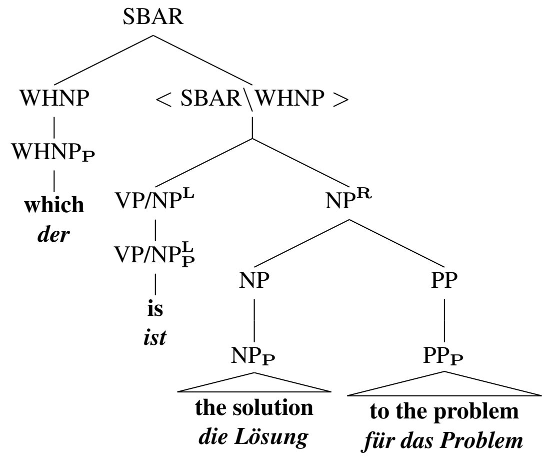

Figure 4: A derivation of a sentence fragment with the grammar of Figure 1.

memorising long phrase-pairs and generalising over yet unseen ones, as shown in the next example.

The derivation in Figure 4 illustrates some of the formalism's features. A preference to reorder based on lexical *content* is applied for is / ist. Noun phrase NP$^{\mathbf{R}}$ is recursively constructed with a preference to constitute the right branch of an order swapping nonterminal expansion. This is matched with VP/NP$^{\mathbf{L}}$ which reorders in the opposite direction. The labels VP/NP and SBAR\WHNP allow linguistic syntax *context* to influence the lexical and reordering translation choices. Crucially, *all* these lexical, attachment and reordering preferences (as encoded in the model's rules and probabilities) must be matched together to arrive at the analysis in Figure 4.

3.3 Parameter Estimation

We estimate the parameters for the phrase-emission model $p(\mathbf{e}, \mathbf{f}|\sigma)$ using Relative Frequency Estimation (RFE) on the *label charts* induced for the training sentence-pairs, after the labels have been augmented by the reordering indications. In the RFE estimate, every rule NT$_{\mathbf{P}} \rightarrow \alpha \;/\; \beta$ receives a probability in proportion with the times that $\alpha \;/\; \beta$ was covered by the NT label.

On the other hand, estimating the parameters under Maximum-Likelihood Estimation (MLE) for the latent translation structure model $p(\sigma)$ is bound to overfit towards memorising whole sentence-pairs as discussed in (Mylonakis and Sima'an, 2010), with the resulting grammar estimate not being able to generalise past the training data. However, apart from overfitting towards long phrase-pairs, a grammar with millions of structural rules is also liable to overfit towards degenerate latent structures which, while fitting the training data well, have limited applicability to unseen sentences.

We avoid both pitfalls by estimating the grammar probabilities with the Cross-Validating Expectation-Maximization algorithm (CV-EM) (Mylonakis and Sima'an, 2008; Mylonakis and Sima'an, 2010). CV-EM is a cross-validating instance of the well known EM algorithm (Dempster et al., 1977). It works iteratively on a partition of the training data, climbing the likelihood of the training data while cross-validating the latent variable values, considering for every training data point only those which can be produced by models built from the rest of the data excluding the current part. As a result, the estimation process simulates maximising future data likelihood, using the training data to directly aim towards strong generalisation of the estimate.

For our probabilistic SCFG-based translation structure variable σ, implementing CV-EM boils down to a synchronous version of the Inside-Outside algorithm, modified to enforce the CV criterion. In this way we arrive at cross-validated ML estimate of the σ parameters while keeping the phrase-emission parameters of $p(\mathbf{e}, \mathbf{f}|\sigma)$ fixed. The CV-criterion, apart from avoiding overfitting, results in discarding the structural rules which are only found in a single part of the training corpus, leading to a more compact grammar while still retaining millions of structural rules that are more hopeful to generalise.

Unravelling the joint generative process, by modelling latent hierarchical structure separately from phrase-pair emission, allows us to concentrate our inference efforts towards the hidden, higher-level translation mechanism.

4 Experiments

4.1 Decoding Model

The induced joint translation model can be used to recover $\arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f})$, as it is equal to $\arg\max_{\mathbf{e}} p(\mathbf{e}, \mathbf{f})$. We employ the induced probabilistic HR-SCFG $\mathbf{G}$ as the backbone of a log-linear, feature based translation model, with the derivation probability $p(D)$ under the grammar estimate being one of the features. This is augmented with a small number n of additional smoothing features ϕ_i for derivation rules r: (a) conditional phrase translation probabilities, (b) lexical phrase translation probabilities, (c) word generation penalty, and (d) a count of swapping reordering operations. Features (a), (b) and (c) are applicable to phrase-pair emission rules and features for both translation directions are used, while (d) is only triggered by structural rules.

These extra features assess translation quality past the synchronous grammar derivation and learning general reordering or word emission preferences for the language pair. As an example, while our probabilistic HR-SCFG maintains a separate joint phrase-pair emission distribution per non-terminal, the smoothing features (a) above assess the conditional translation of surface phrases irrespective of any notion of recursive translation structure.

The final feature is the language model score for the target sentence, mounting up to the following model used at decoding time, with the feature weights λ trained by Minimum Error Rate Training (MERT) (Och, 2003) on a development corpus.

$$p(D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle) \propto p(\mathbf{e})^{\lambda_{lm}} p_{\mathbf{G}}(D)^{\lambda_{\mathbf{G}}} \prod_{i=1}^{n} \prod_{r \in D} \phi_i(r)^{\lambda_i}$$

4.2 Decoding Modifications

We use a customised version of the Joshua SCFG decoder (Li et al., 2009) to translate, with the following modifications:

Source Labels Constraints As for this work the phrase-pair labels used to extract the grammar are based on the linguistic analysis of the source side, we can construct the label chart for every input sentence from its parse. We subsequently use it to consider only derivations with synchronous spans which are covered by non-terminals matching one of the labels for those spans. This applies both for the non-terminals covering phrase-pairs as well as the higher level parts of the derivation.

In this manner we not only constrain the translation hypotheses resulting in faster decoding time, but, more importantly, we may ground the hypotheses more closely to the available linguistic information of the source sentence. This is of particular interest as we move up the derivation tree, where an initial wrong choice below could propagate towards hypotheses wildly diverging from the input sentence's linguistic annotation.

Per Non-Terminal Pruning The decoder uses a combination of beam and cube-pruning (Huang and Chiang, 2007). As our grammar uses non-terminals in the hundreds of thousands, it is important not to prune away prematurely non-terminals covering smaller spans and to leave more options to be considered as we move up the derivation tree.

For this, for every cell in the decoder's chart, we keep a separate bin per non-terminal and prune together hypotheses leading to the same non-terminal covering a cell. This allows full derivations to be found for all input sentences, as well as avoids aggressive pruning at an early stage. Given the source label constraint discussed above, this does not increase running times or memory demands considerably as we allow only up to a few tens of non-terminals per span.

Expected Counts Rule Pruning To compact the hierarchical structure part of the grammar prior to decoding, we prune rules that fail to accumulate 10^{-8} expected counts during the last CV-EM iteration. For English to German, this brings the structural rules from 15M down to 1.2M. Note that we do not prune the phrase-pair emitting rules. Overall, we consider this a much more informed pruning criterion than those based on probability values (that are not comparable across left-hand sides) or right-hand side counts (frequent symbols need many more expansions than a highly specialised one).

4.3 Experimental Setting & Baseline

We evaluate our method on four different language pairs with English as the source language and French, German, Dutch and Chinese as target. The data for the first three language pairs are derived from parliament proceedings sourced from the Europarl corpus (Koehn, 2005), with WMT-07 development and test data for French and German. The data for the English to Chinese task is composed of parliament proceedings and news articles. For all language pairs we employ 200K and 400K sentence pairs for training, 2K for development and 2K for testing (single reference per source sentence). Both the baseline and our method decode

Training set size	English to	French		German		Dutch		Chinese	
		BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
200K	`josh-base`	29.20	7.2123	18.65	5.8047	21.97	6.2469	22.34	6.5540
	`lts`	29.43	7.2611**	19.10**	5.8714**	22.31*	6.2903*	23.67**	6.6595**
400K	`josh-base`	29.58	7.3033	18.86	5.8818	22.25	6.2949	23.24	6.7402
	`lts`	29.83	7.4000**	19.49**	5.9374**	22.92**	6.3727**	25.16**	6.9005**

Table 1: Experimental results for training sets of 200K and 400K sentence pairs. Statistically significant score improvements from the baseline at the 95% confidence level are labelled with a single star, at the 99% level with two.

with a 3-gram language model smoothed with modified Knesser-Ney discounting (Chen and Goodman, 1998), trained on around 1M sentences per target language. The parses of the source sentences employed by our system during training and decoding are created with the Charniak parser (Charniak, 2000).

We compare against a state-of-the-art hierarchical translation (Chiang, 2005) baseline, based on the Joshua translation system under the default training and decoding settings (`josh-base`). Apart of evaluating against a state-of-the-art system, especially on the English-Chinese language pair, the comparison has an added interesting aspect. The heuristically trained baseline takes advantage of ‘gap rules’ to reorder based on lexical context cues, but makes very limited use of the hierarchical structure above the lexical surface. In contrast, our method induces a grammar with no such rules, relying on lexical content and the strength of a higher level translation structure instead.

4.4 Training & Decoding Details

To train our Latent Translation Structure (LTS) system, we used the following settings. CV-EM cross-validated on a 10-part partition of the training data and performed 10 iterations. The structural rule probabilities were initialised to uniform per left-hand side.

The decoder does not employ any ‘glue grammar’ as is usual with hierarchical translation systems to limit reordering up to a certain cut-off length. Instead, we rely on our LTS grammar to reorder and construct the translation output up to the full sentence length.

In summary, our system’s experimental pipeline is as follows. All input sentences are parsed and label charts are created from these parses. The Hierarchical Reordering SCFG is extracted and its parameters are estimated employing CV-EM. The structural rules of the estimate are pruned according to their expected counts and smoothing features are added to all rules. We train the feature weights under MERT and decode with the resulting log-linear model.

The overall training and decoding setup is appealing also regarding computational demands. On an 8-core 2.3GHz system, training on 200K sentence-pairs demands 4.5 hours while decoding runs on 25 sentences per minute.

4.5 Results

Table 1 presents the results for the baseline and our method for the 4 language pairs, for training sets of both 200K and 400K sentence pairs. Our system (`lts`) outperforms the baseline for all 4 language pairs for both BLEU and NIST scores, by a margin which scales up to +1.92 BLEU points for English to Chinese translation when training on the 400K set. In addition, increasing the size of the training data from 200K to 400K sentence pairs widens the performance margin between the baseline and our system, in some cases considerably. All but one of the performance improvements are found to be statistically significant (Koehn, 2004) at the 95% confidence level, most of them also at the 99% level.

We selected an array of target languages of increasing reordering complexity with English as source. Examining the results across the target languages, LTS performance gains increase the more challenging the sentence structure of the target language is in relation to the source’s, highlighted when translating to Chinese. Even for Dutch and German, which pose additional challenges such as compound words and morphology which we do not explicitly treat in the current system, LTS still delivers significant improvements in performance. Additionally,

	System	200K	400K
(a)	`lts-nolabels`	22.50	24.24
	`lts`	**23.67****	**25.16****
(b)	`josh-base-lm4`	23.81	24.77
	`lts-lm4`	**24.48****	**26.35****

Table 2: Additional experiments for English to Chinese translation examining (a) the impact of the linguistic annotations in the LTS system (`lts`), when compared with an instance not employing such annotations (`lts-nolabels`) and (b) decoding with a 4th-order language model (`-lm4`). BLEU scores for 200K and 400K training sentence pairs.

the robustness of our system is exemplified by delivering significant performance increases for all language pairs.

For the English to Chinese translation task, we performed further experiments along two axes. We first investigate the contribution of the linguistic annotations, by comparing our complete system (`lts`) with an otherwise identical implementation (`lts-nolabels`) which does not employ any linguistically motivated labels. The latter system then uses a labels chart as that of Figure 3, which however labels all phrase-pair spans solely with the generic X label. The results in Table 2(a) indicate that a large part of the performance improvement can be attributed to the use of the linguistic annotations extracted from the source parse trees, indicating the potential of the LTS system to take advantage of such additional annotations to deliver better translations.

The second additional experiment relates to the impact of employing a stronger language model during decoding, which may increase performance but slows down decoding speed. Notably, as can be seen in Table 2(b), switching to a 4-gram LM results in performance gains for both the baseline and our system and while the margin between the two systems decreases, our system continues to deliver a considerable and significant improvement in translation BLEU scores.

5 Related Work

In this work, we focus on the combination of learning latent structure with syntax and linguistic annotations, exploring the crossroads of machine learning, linguistic syntax and machine translation. Training a joint probability model was first discussed in (Marcu and Wong, 2002). We show that a translation system based on such a joint model can perform competitively in comparison with conditional probability models, when it is augmented with a rich latent hierarchical structure trained adequately to avoid overfitting.

Earlier approaches for linguistic syntax-based translation such as (Yamada and Knight, 2001; Galley et al., 2006; Huang et al., 2006; Liu et al., 2006) focus on memorising and reusing parts of the structure of the source and/or target parse trees and constraining decoding by the input parse tree. In contrast to this approach, we choose to employ linguistic annotations in the form of unambiguous synchronous span labels, while discovering ambiguous translation structure taking advantage of them.

Later work (Marton and Resnik, 2008; Venugopal et al., 2009; Chiang et al., 2009) takes a more flexible approach, influencing translation output using linguistically motivated features, or features based on source-side linguistically-guided latent syntactic categories (Huang et al., 2010). A feature-based approach and ours are not mutually exclusive, as we also employ a limited set of features next to our trained model during decoding. We find augmenting our system with a more extensive feature set an interesting research direction for the future.

An array of recent work (Chiang, 2010; Zhang et al., 2008; Liu et al., 2009) sets off to utilise source *and* target syntax for translation. While for this work we constrain ourselves to source language syntax annotations, our method can be directly applied to employ labels taking advantage of linguistic annotations from both sides of translation. The decoding constraints of section 4.2 can then still be applied on the source part of hybrid source-target labels.

For the experiments in this paper we employ a label set similar to the non-terminals set of (Zollmann and Venugopal, 2006). However, the synchronous grammars we learn share few similarities with those that they heuristically extract. The HR-SCFG we adopt allows capturing more complex reordering phenomena and, in contrast to both (Chiang, 2005; Zollmann and Venugopal, 2006), is not exposed to the issues highlighted in section 2.1. Nevertheless, our results underline the capacity of linguistic annotations similar to those of (Zollmann and Venugopal, 2006) as part of latent translation variables.

Most of the aforementioned work does concentrate on *learning* hierarchical, linguistically motivated translation models. Cohn and Blunsom (2009) sample rules of the form proposed in (Galley et al., 2004) from a Bayesian model, employing Dirichlet Process priors favouring smaller rules to avoid overfitting. Their grammar is however also based on the target parse-tree structure, with their system surpassing a weak baseline by a small margin. In contrast to the Bayesian approach which imposes external priors to lead estimation away from degenerate solutions, we take a *data-driven* approach to arrive to estimates which generalise well. The rich linguistically motivated latent variable learnt by our method delivers translation performance that compares favourably to a state-of-the-art system.

Mylonakis and Sima'an (2010) also employ the CV-EM algorithm to estimate the parameters of an SCFG, albeit a much simpler one based on a handful of non-terminals. In this work we employ some of their grammar design principles for an immensely more complex grammar with millions of hierarchical latent structure rules and show how such grammar can be learnt and applied taking advantage of source language linguistic annotations.

6 Conclusions

In this work we contribute a method to learn and apply latent hierarchical translation structure. To this end, we take advantage of source-language linguistic annotations to motivate instead of constrain the translation process. An input chart over phrase-pair spans, with each cell filled with multiple linguistically motivated labels, is coupled with the HR-SCFG design to arrive at a rich synchronous grammar with millions of structural rules and the capacity to capture complex linguistically conditioned translation phenomena. We address overfitting issues by cross-validating climbing the likelihood of the training data and propose solutions to increase the efficiency and accuracy of decoding.

An interesting aspect of our work is delivering competitive performance for difficult language pairs such as English-Chinese with a joint probability generative model and an SCFG without 'gap rules'. Instead of employing hierarchical phrase-pairs, we invest in learning the higher-order hierarchical synchronous structure behind translation, up to the full sentence length. While these choices and the related results challenge current MT research trends, they are not mutually exclusive with them. Future work directions include investigating the impact of hierarchical phrases for our models as well as any gains from additional features in the log-linear decoding model.

Smoothing the HR-SCFG grammar estimates could prove a possible source of further performance improvements. Learning translation and reordering behaviour with respect to linguistic cues is facilitated in our approach by keeping separate phrase-pair emission distributions per emitting non-terminal and reordering pattern, while the employment of the generic X non-terminals already allows backing off to more coarse-grained rules. Nevertheless, we still believe that further smoothing of these sparse distributions, e.g. by interpolating them with less sparse ones, could in the future lead to an additional increase in translation quality.

Finally, we discuss in this work how our method can already utilise hundreds of thousands of phrase-pair labels and millions of structural rules. A further promising direction is broadening this set with labels taking advantage of both source and target-language linguistic annotation or categories exploring additional phrase-pair properties past the parse trees such as semantic annotations.

Acknowledgments

Both authors are supported by a VIDI grant (nr. 639.022.604) from The Netherlands Organization for Scientific Research (NWO). The authors would like to thank Maxim Khalilov for helping with experimental data and Andreas Zollmann and the anonymous reviewers for their valuable comments.

References

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North American Association for Computational Linguistics (HLT/NAACL)*, Seattle, Washington, USA, April.

Stanley Chen and Joshua Goodman. 1998. *An empirical study of smoothing techniques for language modeling*. Technical Report TR-10-98, Harvard University, August.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL 2005*, pages 263–270.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

Trevor Cohn and Phil Blunsom. 2009. A Bayesian model of syntax-directed tree to string grammar induction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 352–361, Singapore, August. Association for Computational Linguistics.

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, Series B*, 39(1):1–38.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Zhongqiang Huang, Martin Cmejrek, and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 138–147, Cambridge, MA, October. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL 2003*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit 2005*.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 558–566, Suntec, Singapore, August. Association for Computational Linguistics.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of Empirical methods in natural language processing*, pages 133–139. Association for Computational Linguistics.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011, Columbus, Ohio, June. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2008. Phrase translation probabilities with ITG priors and smoothing as learning objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 630–639, Honolulu, USA, October.

Markos Mylonakis and Khalil Sima'an. 2010. Learning probabilistic synchronous CFGs for phrase-based translation. In *Fourteenth Conference on Computational Natural Language Learning*, Uppsala, Sweden, July.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Michigan, USA, June.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2009. Preference grammars: Softening syntactic constraints to improve statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–244, Boulder, Colorado, June. Association for Computational Linguistics.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36(2):247–277.

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

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

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.