Incremental Syntactic Language Models for Phrase-based Translation

Lane Schwartz
Air Force Research Laboratory
Wright-Patterson AFB, OH USA
lane.schwartz@wpafb.af.mil

Chris Callison-Burch
Johns Hopkins University
Baltimore, MD USA
ccb@cs.jhu.edu

William Schuler
Ohio State University
Columbus, OH USA
schuler@ling.ohio-state.edu

Stephen Wu
Mayo Clinic
Rochester, MN USA
wu.stephen@mayo.edu

Abstract

This paper describes a novel technique for incorporating syntactic knowledge into phrase-based machine translation through incremental syntactic parsing. Bottom-up and top-down parsers typically require a completed string as input. This requirement makes it difficult to incorporate them into phrase-based translation, which generates partial hypothesized translations from left-to-right. Incremental syntactic language models score sentences in a similar left-to-right fashion, and are therefore a good mechanism for incorporating syntax into phrase-based translation. We give a formal definition of one such linear-time syntactic language model, detail its relation to phrase-based decoding, and integrate the model with the Moses phrase-based translation system. We present empirical results on a constrained Urdu-English translation task that demonstrate a significant BLEU score improvement and a large decrease in perplexity.

1 Introduction

Early work in statistical machine translation viewed translation as a noisy channel process comprised of a translation model, which functioned to posit adequate translations of source language words, and a target language model, which guided the fluency of generated target language strings (Brown et al., 1990). Drawing on earlier successes in speech recognition, research in statistical machine translation has effectively used n-gram word sequence models as language models.

Modern phrase-based translation using large scale n-gram language models generally performs well in terms of lexical choice, but still often produces ungrammatical output. Syntactic parsing may help produce more grammatical output by better modeling structural relationships and long-distance dependencies. Bottom-up and top-down parsers typically require a completed string as input; this requirement makes it difficult to incorporate these parsers into phrase-based translation, which generates hypothesized translations incrementally, from left-to-right.[1] As a workaround, parsers can rerank the translated output of translation systems (Och et al., 2004).

On the other hand, incremental parsers (Roark, 2001; Henderson, 2004; Schuler et al., 2010; Huang and Sagae, 2010) process input in a straightforward left-to-right manner. We observe that incremental parsers, used as structured language models, provide an appropriate algorithmic match to incremental phrase-based decoding. We directly integrate incremental syntactic parsing into phrase-based translation. This approach re-exerts the role of the language model as a mechanism for encouraging syntactically fluent translations.

The contributions of this work are as follows:

- A novel method for integrating syntactic LMs into phrase-based translation (§3)
- A formal definition of an incremental parser for

This research was supported by NSF CAREER/PECASE award 0447685, NSF grant IIS-0713448, and the European Commission through the EuroMatrixPlus project. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the sponsors or the United States Air Force. Cleared for public release (Case Number 88ABW-2010-6489) on 10 Dec 2010.

[1] While not all languages are written left-to-right, we will refer to incremental processing which proceeds from the beginning of a sentence as left-to-right.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 620–631,
Portland, Oregon, June 19-24, 2011.

statistical MT that can run in linear-time (§4)

- Integration with Moses (§5) along with empirical results for perplexity and significant translation score improvement on a constrained Urdu-English task (§6)

2 Related Work

Neither phrase-based (Koehn et al., 2003) nor hierarchical phrase-based translation (Chiang, 2005) take explicit advantage of the syntactic structure of either source or target language. The translation models in these techniques define *phrases* as contiguous word sequences (with gaps allowed in the case of hierarchical phrases) which may or may not correspond to any linguistic constituent. Early work in statistical phrase-based translation considered whether restricting translation models to use only syntactically well-formed constituents might improve translation quality (Koehn et al., 2003) but found such restrictions failed to improve translation quality.

Significant research has examined the extent to which syntax can be usefully incorporated into statistical tree-based translation models: string-to-tree (Yamada and Knight, 2001; Gildea, 2003; Imamura et al., 2004; Galley et al., 2004; Graehl and Knight, 2004; Melamed, 2004; Galley et al., 2006; Huang et al., 2006; Shen et al., 2008), tree-to-string (Liu et al., 2006; Liu et al., 2007; Mi et al., 2008; Mi and Huang, 2008; Huang and Mi, 2010), tree-to-tree (Abeillé et al., 1990; Shieber and Schabes, 1990; Poutsma, 1998; Eisner, 2003; Shieber, 2004; Cowan et al., 2006; Nesson et al., 2006; Zhang et al., 2007; DeNeefe et al., 2007; DeNeefe and Knight, 2009; Liu et al., 2009; Chiang, 2010), and treelet (Ding and Palmer, 2005; Quirk et al., 2005) techniques use syntactic information to inform the translation model. Recent work has shown that parsing-based machine translation using syntax-augmented (Zollmann and Venugopal, 2006) hierarchical translation grammars with rich nonterminal sets can demonstrate substantial gains over hierarchical grammars for certain language pairs (Baker et al., 2009). In contrast to the above tree-based translation models, our approach maintains a standard (non-syntactic) phrase-based translation model. Instead, we incorporate syntax into the language model.

Traditional approaches to language models in speech recognition and statistical machine translation focus on the use of n-grams, which provide a simple finite-state model approximation of the target language. Chelba and Jelinek (1998) proposed that syntactic structure could be used as an alternative technique in language modeling. This insight has been explored in the context of speech recognition (Chelba and Jelinek, 2000; Collins et al., 2005). Hassan et al. (2007) and Birch et al. (2007) use supertag n-gram LMs. Syntactic language models have also been explored with tree-based translation models. Charniak et al. (2003) use syntactic language models to rescore the output of a tree-based translation system. Post and Gildea (2008) investigate the integration of parsers as syntactic language models during binary bracketing transduction translation (Wu, 1997); under these conditions, both syntactic phrase-structure and dependency parsing language models were found to improve oracle-best translations, but did not improve actual translation results. Post and Gildea (2009) use tree substitution grammar parsing for language modeling, but do not use this language model in a translation system. Our work, in contrast to the above approaches, explores the use of incremental syntactic language models in conjunction with phrase-based translation models.

Our syntactic language model fits into the family of linear-time dynamic programming parsers described in (Huang and Sagae, 2010). Like (Galley and Manning, 2009) our work implements an incremental syntactic language model; our approach differs by calculating syntactic LM scores over all available phrase-structure parses at each hypothesis instead of the 1-best dependency parse.

The syntax-driven reordering model of Ge (2010) uses syntax-driven features to influence word order within standard phrase-based translation. The syntactic cohesion features of Cherry (2008) encourages the use of syntactically well-formed translation phrases. These approaches are fully orthogonal to our proposed incremental syntactic language model, and could be applied in concert with our work.

3 Parser as Syntactic Language Model in Phrase-Based Translation

Parsing is the task of selecting the representation $\hat{\tau}$ (typically a tree) that best models the structure of

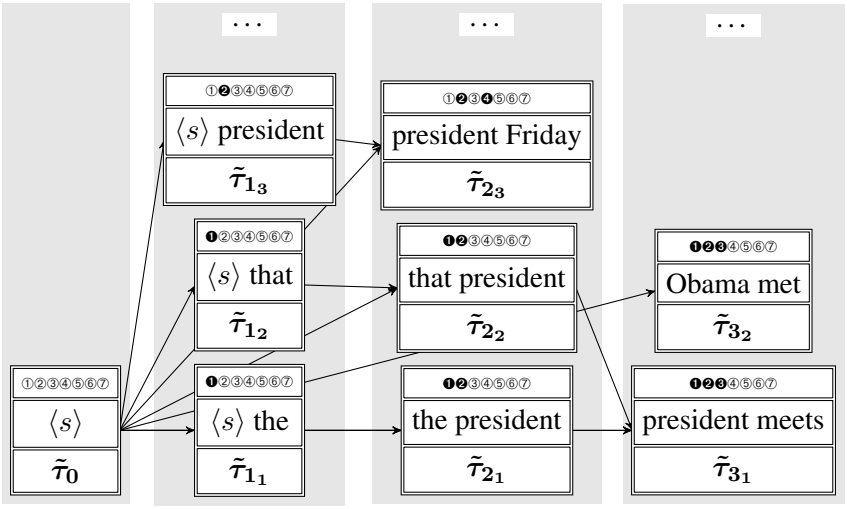

Figure 1: Partial decoding lattice for standard phrase-based decoding stack algorithm translating the German sentence *Der Präsident trifft am Freitag den Vorstand*. Each node h in decoding stack t represents the application of a translation option, and includes the source sentence coverage vector, target language n-gram state, and syntactic language model state $\tilde{\boldsymbol{\tau}}_{\boldsymbol{t_h}}$. Hypothesis combination is also shown, indicating where lattice paths with identical n-gram histories converge. We use the English translation *The president meets the board on Friday* as a running example throughout all Figures.

sentence $\mathbf{e}$, out of all such possible representations $\boldsymbol{\tau}$. This set of representations may be all phrase structure trees or all dependency trees allowed by the parsing model. Typically, tree $\hat{\tau}$ is taken to be:

$$\hat{\tau} = \underset{\tau}{\operatorname{argmax}}\, \mathsf{P}(\tau \,|\, \mathbf{e}) \tag{1}$$

We define a syntactic language model $\mathsf{P}(\mathbf{e})$ based on the total probability mass over all possible trees for string $\mathbf{e}$. This is shown in Equation 2 and decomposed in Equation 3.

$$\mathsf{P}(\mathbf{e}) = \sum_{\tau \in \boldsymbol{\tau}} \mathsf{P}(\tau, \mathbf{e}) \tag{2}$$

$$\mathsf{P}(\mathbf{e}) = \sum_{\tau \in \boldsymbol{\tau}} \mathsf{P}(\mathbf{e} \,|\, \tau)\mathsf{P}(\tau) \tag{3}$$

3.1 Incremental syntactic language model

An incremental parser processes each token of input sequentially from the beginning of a sentence to the end, rather than processing input in a top-down (Earley, 1968) or bottom-up (Cocke and Schwartz, 1970; Kasami, 1965; Younger, 1967) fashion. After processing the tth token in string $\mathbf{e}$, an incremental parser has some internal representation of possible hypothesized (incomplete) trees, $\boldsymbol{\tau_t}$. The syntactic language model probability of a partial sentence $e_1 ... e_t$ is defined:

$$\mathsf{P}(e_1 ... e_t) = \sum_{\tau \in \boldsymbol{\tau_t}} \mathsf{P}(e_1 ... e_t \,|\, \tau)\mathsf{P}(\tau) \tag{4}$$

In practice, a parser may constrain the set of trees under consideration to $\tilde{\boldsymbol{\tau}}_{\boldsymbol{t}}$, that subset of analyses or partial analyses that remains after any pruning is performed. An incremental syntactic language model can then be defined by a probability mass function (Equation 5) and a transition function δ (Equation 6). The role of δ is explained in §3.3 below. Any parser which implements these two functions can serve as a syntactic language model.

$$\mathsf{P}(e_1 ... e_t) \approx \mathsf{P}(\tilde{\boldsymbol{\tau}}_t) = \sum_{\tau \in \tilde{\boldsymbol{\tau}}_t} \mathsf{P}(e_1 ... e_t \,|\, \tau)\mathsf{P}(\tau) \tag{5}$$

$$\delta(e_t, \tilde{\boldsymbol{\tau}}_{t-1}) \rightarrow \tilde{\boldsymbol{\tau}}_t \tag{6}$$

3.2 Decoding in phrase-based translation

Given a source language input sentence $\mathbf{f}$, a trained source-to-target translation model, and a target language model, the task of translation is to find the maximally probable translation $\hat{\boldsymbol{e}}$ using a linear combination of j feature functions $\boldsymbol{h}$ weighted according to tuned parameters $\boldsymbol{\lambda}$ (Och and Ney, 2002).

$$\hat{\boldsymbol{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \exp(\sum_{j} \lambda_j h_j(\boldsymbol{e}, \boldsymbol{f})) \quad (7)$$

Phrase-based translation constructs a set of translation options — hypothesized translations for contiguous portions of the source sentence — from a trained phrase table, then incrementally constructs a lattice of partial target translations (Koehn, 2010). To prune the search space, lattice nodes are organized into beam stacks (Jelinek, 1969) according to the number of source words translated. An n-gram language model history is also maintained at each node in the translation lattice. The search space is further trimmed with hypothesis recombination, which collapses lattice nodes that share a common coverage vector and n-gram state.

3.3 Incorporating a Syntactic Language Model

Phrase-based translation produces target language words in an incremental left-to-right fashion, generating words at the beginning of a translation first and words at the end of a translation last. Similarly, incremental parsers process sentences in an incremental fashion, analyzing words at the beginning of a sentence first and words at the end of a sentence last. As such, an incremental parser with transition function δ can be incorporated into the phrase-based decoding process in a straightforward manner. Each node in the translation lattice is augmented with a syntactic language model state $\tilde{\boldsymbol{\tau}}_t$.

The hypothesis at the root of the translation lattice is initialized with $\tilde{\boldsymbol{\tau}}_0$, representing the internal state of the incremental parser before any input words are processed. The phrase-based translation decoding process adds nodes to the lattice; each new node contains one or more target language words. Each node contains a backpointer to its parent node, in which $\tilde{\boldsymbol{\tau}}_{t-1}$ is stored. Given a new target language word e_t and $\tilde{\boldsymbol{\tau}}_{t-1}$, the incremental parser's transition function δ calculates $\tilde{\boldsymbol{\tau}}_t$. Figure 1 illustrates

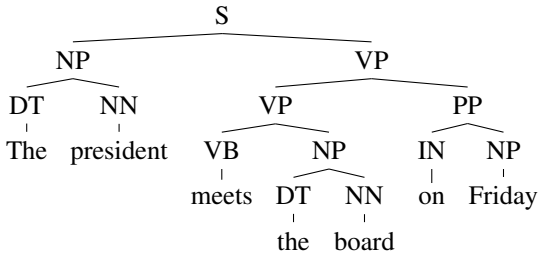

Figure 2: Sample binarized phrase structure tree.

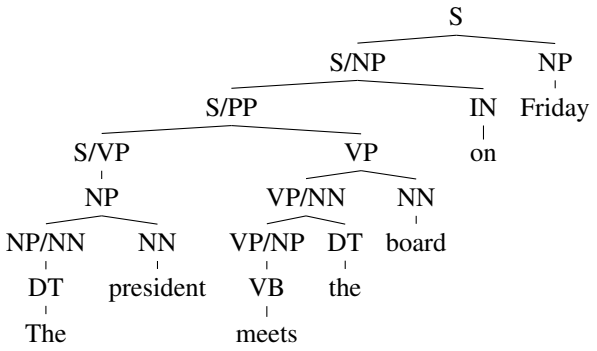

Figure 3: Sample binarized phrase structure tree after application of right-corner transform.

a sample phrase-based decoding lattice where each translation lattice node is augmented with syntactic language model state $\tilde{\boldsymbol{\tau}}_t$.

In phrase-based translation, many translation lattice nodes represent multi-word target language phrases. For such translation lattice nodes, δ will be called once for each newly hypothesized target language word in the node. Only the final syntactic language model state in such sequences need be stored in the translation lattice node.

4 Incremental Bounded-Memory Parsing with a Time Series Model

Having defined the framework by which any incremental parser may be incorporated into phrase-based translation, we now formally define a specific incremental parser for use in our experiments.

The parser must process target language words incrementally as the phrase-based decoder adds hypotheses to the translation lattice. To facilitate this incremental processing, ordinary phrase-structure trees can be transformed into right-corner recur-

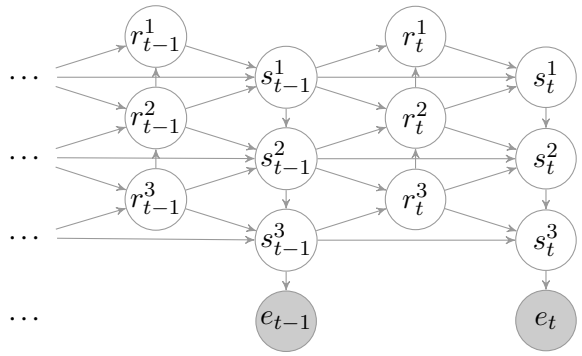

Figure 4: Graphical representation of the dependency structure in a standard Hierarchic Hidden Markov Model with $D = 3$ hidden levels that can be used to parse syntax. Circles denote random variables, and edges denote conditional dependencies. Shaded circles denote variables with observed values.

sive phrase structure trees using the tree transforms in Schuler et al. (2010). Constituent nonterminals in right-corner transformed trees take the form of *incomplete constituents* $c_\eta/c_{\eta\iota}$ consisting of an 'active' constituent c_η lacking an 'awaited' constituent $c_{\eta\iota}$ yet to come, similar to non-constituent categories in a Combinatory Categorial Grammar (Ades and Steedman, 1982; Steedman, 2000). As an example, the parser might consider VP/NN as a possible category for input "meets the".

A sample phrase structure tree is shown before and after the right-corner transform in Figures 2 and 3. Our parser operates over a right-corner transformed probabilistic context-free grammar (PCFG). Parsing runs in linear time on the length of the input. This model of incremental parsing is implemented as a Hierarchical Hidden Markov Model (HHMM) (Murphy and Paskin, 2001), and is equivalent to a probabilistic pushdown automaton with a bounded pushdown store. The parser runs in O(n) time, where n is the number of words in the input. This model is shown graphically in Figure 4 and formally defined in §4.1 below.

The incremental parser assigns a probability (Eq. 5) for a partial target language hypothesis, using a bounded store of incomplete constituents $c_\eta/c_{\eta\iota}$. The phrase-based decoder uses this probability value as the syntactic language model feature score.

4.1 Formal Parsing Model: Scoring Partial Translation Hypotheses

This model is essentially an extension of an HHMM, which obtains a most likely sequence of hidden store states, $\hat{s}^{1..D}_{1..T}$, of some length T and some maximum depth D, given a sequence of observed tokens (e.g. generated target language words), $e_{1..T}$, using HHMM state transition model θ_A and observation symbol model θ_B (Rabiner, 1990):

$$\hat{s}^{1..D}_{1..T} \stackrel{\text{def}}{=} \underset{s^{1..D}_{1..T}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\theta_\text{A}}(s^{1..D}_t \mid s^{1..D}_{t-1}) \cdot \mathsf{P}_{\theta_\text{B}}(e_t \mid s^{1..D}_t) \tag{8}$$

The HHMM parser is equivalent to a probabilistic pushdown automaton with a bounded pushdown store. The model generates each successive store (using store model θ_S) only after considering whether each nested sequence of incomplete constituents has completed and reduced (using reduction model θ_R):

$$\begin{aligned}\mathsf{P}_{\theta_\text{A}}(s^{1..D}_t \mid s^{1..D}_{t-1}) \stackrel{\text{def}}{=} \\ \sum_{r^1_t..r^D_t} \prod_{d=1}^{D} & \mathsf{P}_{\theta_\text{R}}(r^d_t \mid r^{d+1}_t s^d_{t-1} s^{d-1}_{t-1}) \\ & \cdot \mathsf{P}_{\theta_\text{S}}(s^d_t \mid r^{d+1}_t r^d_t\, s^d_{t-1} s^{d-1}_t)\end{aligned} \tag{9}$$

Store elements are defined to contain only the active (c_η) and awaited ($c_{\eta\iota}$) constituent categories necessary to compute an incomplete constituent probability:

$$s^d_t \stackrel{\text{def}}{=} \langle c_\eta, c_{\eta\iota} \rangle \tag{10}$$

Reduction states are defined to contain only the complete constituent category $c_{r^d_t}$ necessary to compute an inside likelihood probability, as well as a flag $f_{r^d_t}$ indicating whether a reduction has taken place (to end a sequence of incomplete constituents):

$$r^d_t \stackrel{\text{def}}{=} \langle c_{r^d_t}, f_{r^d_t} \rangle \tag{11}$$

The model probabilities for these store elements and reduction states can then be defined (from Murphy and Paskin 2001) to expand a new incomplete constituent after a reduction has taken place ($f_{r^d_t} = 1$; using depth-specific store state expansion model $\theta_{\text{S-E},d}$), transition along a sequence of store elements

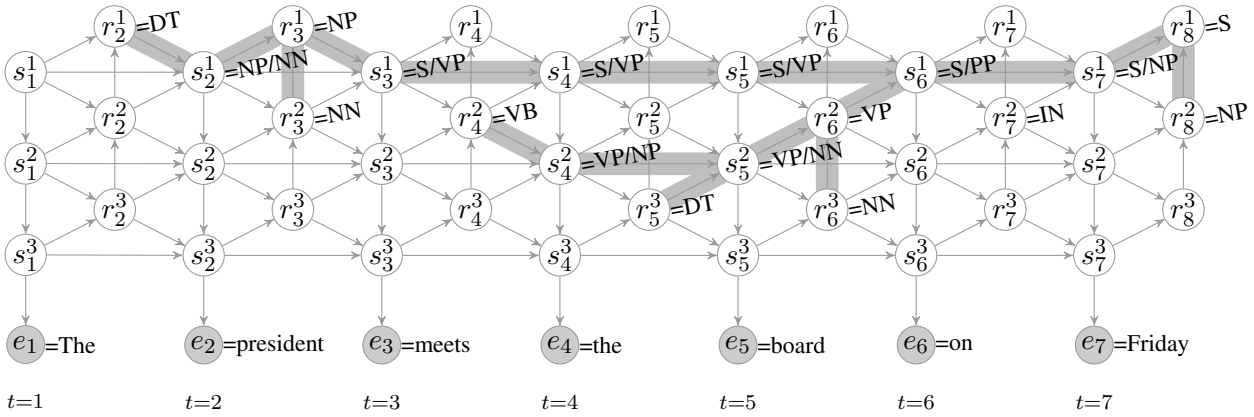

Figure 5: Graphical representation of the Hierarchic Hidden Markov Model after parsing input sentence *The president meets the board on Friday*. The shaded path through the parse lattice illustrates the recognized right-corner tree structure of Figure 3.

if no reduction has taken place ($f_{r_t^d}=0$; using depth-specific store state transition model $\theta_{\text{S-T},d}$): [2]

$$\mathsf{P}_{\theta_{\text{S}}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}}=1,\; f_{r_t^d}=1 : \mathsf{P}_{\theta_{\text{S-E},d}}(s_t^d \mid s_t^{d-1}) \\ \text{if } f_{r_t^{d+1}}=1,\; f_{r_t^d}=0 : \mathsf{P}_{\theta_{\text{S-T},d}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \\ \text{if } f_{r_t^{d+1}}=0,\; f_{r_t^d}=0 : [\![s_t^d = s_{t-1}^d]\!] \end{cases} \tag{12}$$

and possibly reduce a store element (terminate a sequence) if the store state below it has reduced ($f_{r_t^{d+1}}=1$; using depth-specific reduction model $\theta_{\text{R},d}$):

$$\mathsf{P}_{\theta_{\text{R}}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}}=0 : [\![r_t^d = \mathbf{r}_\perp]\!] \\ \text{if } f_{r_t^{d+1}}=1 : \mathsf{P}_{\theta_{\text{R},d}}(r_t^d \mid r_t^{d+1} \; s_{t-1}^d \; s_{t-1}^{d-1}) \end{cases} \tag{13}$$

where $\mathbf{r}_\perp$ is a null state resulting from the failure of an incomplete constituent to complete, and constants are defined for the edge conditions of s_t^0 and r_t^{D+1}. Figure 5 illustrates this model in action.

These pushdown automaton operations are then refined for right-corner parsing (Schuler, 2009), distinguishing *active* transitions (model $\theta_{\text{S-T-A},d}$, in which an incomplete constituent is completed, but not reduced, and then immediately expanded to a new incomplete constituent in the same store element) from *awaited* transitions (model $\theta_{\text{S-T-W},d}$, which involve no completion):

$$\mathsf{P}_{\theta_{\text{S-T},d}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } r_t^d \neq \mathbf{r}_\perp : \mathsf{P}_{\theta_{\text{S-T-A},d}}(s_t^d \mid s_t^{d-1} r_t^d) \\ \text{if } r_t^d = \mathbf{r}_\perp : \mathsf{P}_{\theta_{\text{S-T-W},d}}(s_t^d \mid s_{t-1}^d r_t^{d+1}) \end{cases} \tag{14}$$

$$\mathsf{P}_{\theta_{\text{R},d}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } c_{r_t^{d+1}} \neq x_t : [\![r_t^d = \mathbf{r}_\perp]\!] \\ \text{if } c_{r_t^{d+1}} = x_t : \mathsf{P}_{\theta_{\text{R-R},d}}(r_t^d \mid s_{t-1}^d s_{t-1}^{d-1}) \end{cases} \tag{15}$$

These HHMM right-corner parsing operations are then defined in terms of branch- and depth-specific PCFG probabilities $\theta_{\text{G-R},d}$ and $\theta_{\text{G-L},d}$: [3]

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{0} c_{\eta 0} \ldots) \stackrel{\text{def}}{=} \sum_{c_{\eta 1}} \mathsf{P}_{\theta_{\text{G-R},d}}(c_\eta \rightarrow c_{\eta 0} \; c_{\eta 1}) \tag{16}$$

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{k} c_{\eta 0^k 0} \ldots) \stackrel{\text{def}}{=} \sum_{c_{\eta 0^k}} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{k-1} c_{\eta 0^k} \ldots) \cdot \sum_{c_{\eta 0^k 1}} \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta 0^k} \rightarrow c_{\eta 0^k 0} \; c_{\eta 0^k 1}) \tag{17}$$

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota} \ldots) \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{k} c_{\eta\iota} \ldots) \tag{18}$$

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{+} c_{\eta\iota} \ldots) \stackrel{\text{def}}{=} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota} \ldots) - \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{0} c_{\eta\iota} \ldots) \tag{19}$$

[2] An indicator function $[\![\cdot]\!]$ is used to denote deterministic probabilities: $[\![\phi]\!] = 1$ if ϕ is true, 0 otherwise.

[3] Model probabilities are also defined in terms of left-progeny probability distribution $\mathsf{E}_{\theta_{\text{G-RL}^*,d}}$ which is itself defined in terms of PCFG probabilities:

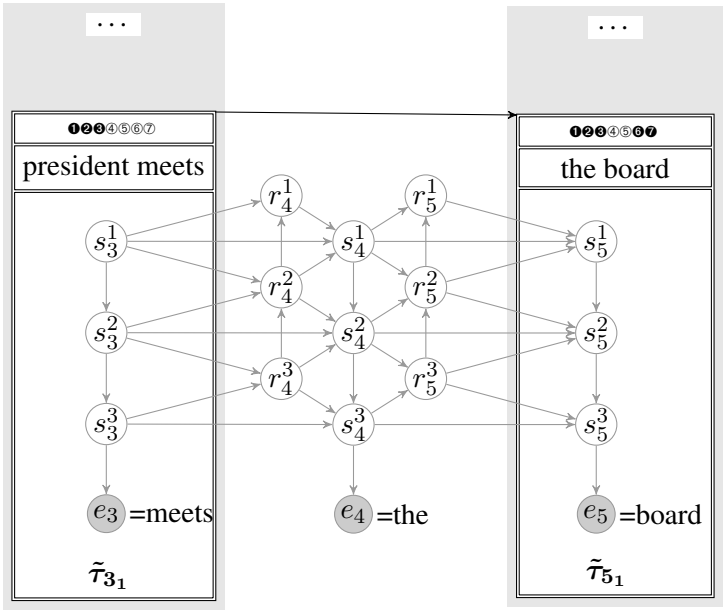

Figure 6: A hypothesis in the phrase-based decoding lattice from Figure 1 is expanded using translation option *the board* of source phrase *den Vorstand*. Syntactic language model state $\tilde{\boldsymbol{\tau}}_{\mathbf{3_1}}$ contains random variables $s_3^{1..3}$; likewise $\tilde{\boldsymbol{\tau}}_{\mathbf{5_1}}$ contains $s_5^{1..3}$. The intervening random variables $r_4^{1..3}$, $s_4^{1..3}$, and $r_5^{1..3}$ are calculated by transition function δ (Eq. 6, as defined by §4.1), but are not stored. Observed random variables ($e_3..e_5$) are shown for clarity, but are not explicitly stored in any syntactic language model state.

- for expansions:

$$\mathsf{P}_{\theta_{\text{S-E},d}}(\langle c_{\eta\iota}, c'_{\eta\iota}\rangle \mid \langle -, c_\eta\rangle) \stackrel{\text{def}}{=} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota}\ ...) \cdot [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \quad (20)$$

- for awaited transitions:

$$\mathsf{P}_{\theta_{\text{S-T-W},d}}(\langle c_\eta, c_{\eta\iota 1}\rangle \mid \langle c'_\eta, c_{\eta\iota}\rangle\ c_{\eta\iota 0}) \stackrel{\text{def}}{=} [\![c_\eta = c'_\eta]\!] \cdot \frac{\mathsf{P}_{\theta_{\text{G-R},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_{\eta\iota} \xrightarrow{0} c_{\eta\iota 0}\ ...)} \quad (21)$$

- for active transitions:

$$\mathsf{P}_{\theta_{\text{S-T-A},d}}(\langle c_{\eta\iota}, c_{\eta\iota 1}\rangle \mid \langle -, c_\eta\rangle\ c_{\eta\iota 0}) \stackrel{\text{def}}{=} \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota}\ ...) \cdot \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{+} c_{\eta\iota 0}\ ...)} \quad (22)$$

- for cross-element reductions:

$$\mathsf{P}_{\theta_{\text{R-R},d}}(c_{\eta\iota}, \mathbf{1} \mid \langle -, c_\eta\rangle\ \langle c'_{\eta\iota}, -\rangle) \stackrel{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{0} c_{\eta\iota}\ ...)}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota}\ ...)} \quad (23)$$

- for in-element reductions:

$$\mathsf{P}_{\theta_{\text{R-R},d}}(c_{\eta\iota}, \mathbf{0} \mid \langle -, c_\eta\rangle\ \langle c'_{\eta\iota}, -\rangle) \stackrel{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{+} c_{\eta\iota}\ ...)}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \xrightarrow{*} c_{\eta\iota}\ ...)} \quad (24)$$

We use the parser implementation of (Schuler, 2009; Schuler et al., 2010).

5 Phrase Based Translation with an Incremental Syntactic Language Model

The phrase-based decoder is augmented by adding additional state data to each hypothesis in the de-

coder's hypothesis stacks. Figure 1 illustrates an excerpt from a standard phrase-based translation lattice. Within each decoder stack t, each hypothesis h is augmented with a syntactic language model state $\tilde{\boldsymbol{\tau}}_{\boldsymbol{t_h}}$. Each syntactic language model state is a random variable store, containing a slice of random variables from the HHMM. Specifically, $\tilde{\boldsymbol{\tau}}_{\boldsymbol{t_h}}$ contains those random variables $s_t^{1..D}$ that maintain distributions over syntactic elements.

By maintaining these syntactic random variable stores, each hypothesis has access to the current language model probability for the partial translation ending at that hypothesis, as calculated by an incremental syntactic language model defined by the HHMM. Specifically, the random variable store at hypothesis h provides $\mathsf{P}(\tilde{\boldsymbol{\tau}}_{\boldsymbol{t_h}}) = \mathsf{P}(e_{1..t}^h, s_{1..t}^{1..D})$, where $e_{1..t}^h$ is the sequence of words in a partial hypothesis ending at h which contains t target words, and where there are D syntactic random variables in each random variable store (Eq. 5).

During stack decoding, the phrase-based decoder progressively constructs new hypotheses by extending existing hypotheses. New hypotheses are placed in appropriate hypothesis stacks. In the simplest case, a new hypothesis extends an existing hypothesis by exactly one target word. As the new hypothesis is constructed by extending an existing stack element, the store and reduction state random variables are processed, along with the newly hypothesized word. This results in a new store of syntactic random variables (Eq. 6) that are associated with the new stack element.

When a new hypothesis extends an existing hypothesis by more than one word, this process is first carried out for the first new word in the hypothesis. It is then repeated for the remaining words in the hypothesis extension. Once the final word in the hypothesis has been processed, the resulting random variable store is associated with that hypothesis. The random variable stores created for the non-final words in the extending hypothesis are discarded, and need not be explicitly retained.

Figure 6 illustrates this process, showing how a syntactic language model state $\tilde{\boldsymbol{\tau}}_{\mathbf{5_1}}$ in a phrase-based decoding lattice is obtained from a previous syntactic language model state $\tilde{\boldsymbol{\tau}}_{\mathbf{3_1}}$ (from Figure 1) by parsing the target language words from a phrase-based translation option.

LM	In-domain WSJ 23 ppl	Out-of-domain ur-en dev ppl
WSJ 1-gram	1973.57	3581.72
WSJ 2-gram	349.18	1312.61
WSJ 3-gram	262.04	1264.47
WSJ 4-gram	244.12	1261.37
WSJ 5-gram	**232.08**	1261.90
WSJ HHMM	384.66	**529.41**
Interpolated WSJ 5-gram + HHMM	***209.13***	225.48
Giga 5-gram	258.35	312.28
Interp. Giga 5-gr + WSJ HHMM	222.39	***123.10***
Interp. Giga 5-gr + WSJ 5-gram	174.88	321.05

Figure 7: Average per-word perplexity values. HHMM was run with beam size of 2000. **Bold** indicates best single-model results for LMs trained on WSJ sections 2-21. Best overall in ***italics***.

Our syntactic language model is integrated into the current version of Moses (Koehn et al., 2007).

6 Results

As an initial measure to compare language models, average per-word perplexity, ppl, reports how surprised a model is by test data. Equation 25 calculates ppl using log base b for a test set of T tokens.

$$ppl = b^{\frac{-log_b \mathsf{P}(e_1 \ldots e_T)}{T}} \quad (25)$$

We trained the syntactic language model from §4 (HHMM) and an interpolated n-gram language model with modified Kneser-Ney smoothing (Chen and Goodman, 1998); models were trained on sections 2-21 of the Wall Street Journal (WSJ) treebank (Marcus et al., 1993). The HHMM outperforms the n-gram model in terms of out-of-domain test set perplexity when trained on the same WSJ data; the best perplexity results for in-domain and out-of-domain test sets[4] are found by interpolating

[4]In-domain is WSJ Section 23. Out-of-domain are the English reference translations of the dev section , set aside in (Baker et al., 2009) for parameter tuning, of the NIST Open MT 2008 Urdu-English task.

Sentence length	Moses	+HHMM beam=50	+HHMM beam=2000
10	0.21	533	1143
20	0.53	1193	2562
30	0.85	1746	3749
40	1.13	2095	4588

Figure 8: Mean per-sentence decoding time (in seconds) for dev set using Moses with and without syntactic language model. HHMM parser beam sizes are indicated for the syntactic LM.

HHMM and n-gram LMs (Figure 7). To show the effects of training an LM on more data, we also report perplexity results on the 5-gram LM trained for the GALE Arabic-English task using the English Gigaword corpus. In all cases, including the HHMM significantly reduces perplexity.

We trained a phrase-based translation model on the full NIST Open MT08 Urdu-English translation model using the full training data. We trained the HHMM and n-gram LMs on the WSJ data in order to make them as similar as possible. During tuning, Moses was first configured to use just the n-gram LM, then configured to use both the n-gram LM and the syntactic HHMM LM. MERT consistently assigned positive weight to the syntactic LM feature, typically slightly less than the n-gram LM weight.

In our integration with Moses, incorporating a syntactic language model dramatically slows the decoding process. Figure 8 illustrates a slowdown around three orders of magnitude. Although speed remains roughly linear to the size of the source sentence (ruling out exponential behavior), it is with an extremely large constant time factor. Due to this slowdown, we tuned the parameters using a constrained dev set (only sentences with 1-20 words), and tested using a constrained devtest set (only sentences with 1-20 words). Figure 9 shows a statistically significant improvement to the BLEU score when using the HHMM and the n-gram LMs together on this reduced test set.

Moses LM(s)	BLEU
n-gram only	18.78
HHMM + n-gram	**19.78**

Figure 9: Results for Ur-En devtest (only sentences with 1-20 words) with HHMM beam size of 2000 and Moses settings of distortion limit 10, stack size 200, and ttable_limit 20.

7 Discussion

This paper argues that incremental syntactic languages models are a straightforward and appropriate algorithmic fit for incorporating syntax into phrase-based statistical machine translation, since both process sentences in an incremental left-to-right fashion. This means incremental syntactic LM scores can be calculated during the decoding process, rather than waiting until a complete sentence is posited, which is typically necessary in top-down or bottom-up parsing.

We provided a rigorous formal definition of incremental syntactic languages models, and detailed what steps are necessary to incorporate such LMs into phrase-based decoding. We integrated an incremental syntactic language model into Moses. The translation quality significantly improved on a constrained task, and the perplexity improvements suggest that interpolating between n-gram and syntactic LMs may hold promise on larger data sets.

The use of very large n-gram language models is typically a key ingredient in the best-performing machine translation systems (Brants et al., 2007). Our n-gram model trained only on WSJ is admittedly small. Our future work seeks to incorporate large-scale n-gram language models in conjunction with incremental syntactic language models.

The added decoding time cost of our syntactic language model is very high. By increasing the beam size and distortion limit of the baseline system, future work may examine whether a baseline system with comparable runtimes can achieve comparable translation quality.

A more efficient implementation of the HHMM parser would speed decoding and make more extensive and conclusive translation experiments possible. Various additional improvements could include caching the HHMM LM calculations, and exploiting properties of the right-corner transform that limit the number of decisions between successive time steps.

References

Anne Abeillé, Yves Schabes, and Aravind K. Joshi. 1990. Using lexicalized tree adjoining grammars for machine translation. In *Proceedings of the 13th International Conference on Computational Linguistics*.

Anthony E. Ades and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Kathy Baker, Steven Bethard, Michael Bloodgood, Ralf Brown, Chris Callison-Burch, Glen Coppersmith, Bonnie Dorr, Wes Filardo, Kendall Giles, Anni Irvine, Mike Kayser, Lori Levin, Justin Martineau, Jim Mayfield, Scott Miller, Aaron Phillips, Andrew Philpot, Christine Piatko, Lane Schwartz, and David Zajic. 2009. Semantically informed machine translation (SIMT). SCALE summer workshop final report, Human Language Technology Center Of Excellence.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Peter Brown, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Frederick Jelinek, John Lafferty, Robert Mercer, and Paul Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of the Ninth Machine Translation Summit of the International Association for Machine Translation*.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 225–231.

Ciprian Chelba and Frederick Jelinek. 2000. Structured language modeling. *Computer Speech and Language*, 14(4):283–332.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University.

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 72–80.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 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.

John Cocke and Jacob Schwartz. 1970. Programming languages and their compilers. Technical report, Courant Institute of Mathematical Sciences, New York University.

Michael Collins, Brian Roark, and Murat Saraclar. 2005. Discriminative syntactic language modeling for speech recognition. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 507–514.

Brooke Cowan, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 232–241.

Steve DeNeefe and Kevin Knight. 2009. Synchronous tree adjoining machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 727–736.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 755–763.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 541–548.

Jay Earley. 1968. *An efficient context-free parsing algorithm*. Ph.D. thesis, Department of Computer Science, Carnegie Mellon University.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208.

Michel Galley and Christopher D. Manning. 2009. Quadratic-time dependency parsing for machine translation. 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 773–781.

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, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 273–280.

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.

Niyu Ge. 2010. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 849–857.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 80–87.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 105–112.

Hany Hassan, Khalil Sima'an, and Andy Way. 2007. Supertagged phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 288–295.

James Henderson. 2004. Lookahead in deterministic left-corner parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 26–33.

Liang Huang and Haitao Mi. 2010. Efficient incremental decoding for tree-to-string translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 273–283.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086.

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*.

Kenji Imamura, Hideo Okuma, Taro Watanabe, and Eiichiro Sumita. 2004. Example-based machine translation based on syntactic transfer with statistical models. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 99–105.

Frederick Jelinek. 1969. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, pages 675–685.

T. Kasami. 1965. An efficient recognition and syntax analysis algorithm for context free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

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.

Yang Liu, Yun Huang, Qun Liu, and Shouxun Lin. 2007. Forest-to-string statistical translation rules. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 704–711.

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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 653–660.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 192–199.

Kevin P. Murphy and Mark A. Paskin. 2001. Linear time inference in hierarchical HMMs. In *Proceedings of Neural Information Processing Systems*, pages 833–840.

Rebecca Nesson, Stuart Shieber, and Alexander Rush. 2006. Induction of probabilistic synchronous tree-insertion grammars for machine translation. In *Proceedings of the 7th Biennial conference of the Association for Machine Translation in the Americas*, pages 128–137.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 161–168.

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 172–181.

Matt Post and Daniel Gildea. 2009. Language modeling with tree substitution grammars. In *NIPS workshop on Grammar Induction, Representation of Language, and Language Learning*.

Arjen Poutsma. 1998. Data-oriented translation. In *Ninth Conference of Computational Linguistics in the Netherlands*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 271–279.

Lawrence R. Rabiner. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1):1–30.

William Schuler. 2009. Positive results for parsing with a bounded stack using a model-based right-corner transform. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 344–352.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 577–585.

Stuart M. Shieber and Yves Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*.

Stuart M. Shieber. 2004. Synchronous grammars as tree transducers. In *Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms*.

Mark Steedman. 2000. *The syntactic process*. MIT Press/Bradford Books, Cambridge, MA.

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.

D.H. Younger. 1967. Recognition and parsing of context-free languages in time n cubed. *Information and Control*, 10(2):189–208.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Seng Li, and Chew Lim Tan. 2007. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of the 11th Machine Translation Summit of the International Association for Machine Translation*, pages 535–542.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141.