Learning Multiple-Nonterminal Synchronous Grammars for Statistical Machine Translation

Andreas Zollmann

CMU-LTI-11-007

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Stephan Vogel (Chair)
Alon Lavie
Noah A. Smith
Jay Ponte, Google Inc.

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

ABSTRACT

Recent work in machine translation has evolved from the traditional word and phrase-based models to include hierarchical phrase-based and syntax-based models. These advances are motivated by the desire to integrate richer knowledge within the translation process to explicitly address limitations of the purely lexical phrase-based model.

Generalized phrases as discussed in (Chiang, 2005) attempt to directly address the limitations of purely lexical phrases, and have shown significant improvements in translation quality by introducing constructs for sub-phrase representation. However, generalizations are represented by a single sub-phrase category (and a glue rule for serial combination), providing the ability (and risk) of inserting any available sub-phrase into a larger phrase.

The first contribution of this dissertation work is the grammar extraction method of syntax-augmented machine translation (SAMT), an extension to Chiang's model that provides multiple generalization types based on the phrase-structure parse trees of the training target sentences. We report improvements over strong phrase-based as well as hierarchical phrase-based baselines for French-to-English, Chinese-to-English, and Urdu-to-English.

We then propose several improvements to hierarchical and syntax-augmented MT. We add a source-span variance model that estimates rule probabilities based on the number of source words spanned by the rule and its substituted child rules, introduce methods of combining hierarchical and syntax-based PSCFG models, and experiment with syntax-augmented MT variants based on source-side syntax as well as joint source and target syntax.

Syntax-based models such as SAMT typically rely on word-alignments

and parse trees of the training sentence pairs, which are assumed to be correct. In reality, these alignments and parses are not human-generated, but instead result from the most probable configuration of a stochastic model. We provide a method to induce grammars over hidden alignments and parses, approximated from N-best lists. We present results showing improvements for hierarchical phrase-based MT as well as SAMT when using the widened pipeline.

The SAMT model presupposes the availability of phrase-structure parse trees for the target training sentences. However, syntactic parsers are only available for a limited set of languages. We propose methods to label probabilistic synchronous context-free grammar (PSCFG) rules using only word tags, generated by either part-of-speech analysis or unsupervised word class induction. The proposals range from simple tag-combination schemes to a phrase clustering model that can incorporate an arbitrary number of features. Our models improve translation quality over Chiang's hierarchical phrase-based MT model on the NIST large resource Chinese-to-English translation task. These improvements persist when using automatically learned word tags, suggesting broad applicability of our technique across diverse language pairs for which syntactic resources are not available.

TABLE OF CONTENTS

CHAPTER 1

Introduction

Recent work in machine translation (MT) has evolved from the traditional word (Brown, Pietra, Pietra, and Mercer, 1993) and phrase-based (Koehn, Och, and Marcu, 2004) models to include hierarchical phrase-based (Chiang, 2005) and syntax based models (Poutsma, 2000; Yamada and Knight, 2001; Galley, Hopkins, Knight, and Marcu, 2004). These advances are motivated by the desire to integrate more information such as context-dependent reordering behavior and hypothesis compatibility within the translation process to explicitly address limitations of the purely lexical phrase-based model. As Chiang (2005) and Koehn, Och, and Marcu (2003) note, phrase-based models suffer from sparse data effects when required to translate conceptual elements that span or skip across several words, and distortion based reordering techniques tend to limit their range of operation for reasons of efficiency and model strength (Och and Ney, 2004).

Generalized phrases as discussed in Chiang (2005) and noted in Block (2000), attempt to directly address the limitations of purely lexical phrases, and have shown significant improvements in translation quality by introducing constructs for sub-phrase representation. Block (2000) introduces a single generalization per phrase within the example-based MT framework, while Chiang (2005) can generate multiple generalizations within each phrase. In both these cases, however, generalizations are represented by a single sub-

phrase category (and a glue rule for serial combination), providing the ability (and risk) of inserting any available sub-phrase into a larger phrase.

The use of a single generalization category X (left hand side in CFG notation) in the model of Chiang (2005) comes at the cost of the lost opportunity to model the types of phrase pairs represented by a generalization operation, resulting in a less directed search process during decoding. Figure 1.0.1 gives a training corpus consisting of only training sentence pairs, and some of the rules that would be extracted from it when using Chiang's method. Even for that simple corpus, the model is unable to unambiguously reproduce the correct translation when presented with the second training source sentence. Instead it is ambivalent between two equally probable translation hypotheses, one of them correct and one false. Even worse, if the first training sentence pair were repeated twice, the incorrect hypothesis would become more probable than the correct one.

The first contribution of this dissertation work is syntax-augmented machine translation (SAMT), an extension to Chiang's model that provides multiple generalization types based on the phrase-structure parse trees of the training target sentences (Chapter 3). Figure 1.0.2 sketches how this model solves the problem of reproducing the example training corpus above unambiguously. This was the first syntax-based MT system to achieve an improvement over phrase-based MT (Zollmann and Venugopal, 2006). We show how to scale our model to large-data translation tasks using the multi-processor MapReduce paradigm and present experimental results across different language pairs, showing improvements over strong phrase-based as well as hierarchical phrase-based baselines for French-to-English, Chinese-to-English, and Urdu-to-English.

S1: die Frau , die ein UFO gesehen hat , ist nicht verrueckt . |
the woman who has seen a UFO is not crazy .

(1A) die Frau | the woman

(1B) Frau | woman

(1C) ein UFO gesehen | seen a UFO

(1D) X_1 , die X_2 hat X_3 . | X_1 who has X_2 X_3 .

S2: ich glaube , die Frau hat ein UFO gesehen . |
I think the woman has seen a UFO .

(2A) ich glaube | I think

(2B) X_1 , die X_2 hat X_3 . | X_1 the X_2 has X_3 .

Test sentence: ich glaube , die Frau hat ein UFO gesehen .

either: $\xrightarrow{2B+2A+1B+1C}$ I think the woman has seen a UFO .

or: $\xrightarrow{1D+2B+1B+1C}$ I think who has woman seen a UFO .

Figure 1.0.1: A training corpus which the hierarchical SMT model of Chiang (2005) fails to reproduce unambivalently.

- Use nonterminal labels to constrain the space of eligible derivations

S1: die Frau , die ein UFO gesehen hat , ist nicht verrueckt . |
the woman who has seen a UFO is not crazy .

(1A) ***NP*** $\rightarrow$ die Frau | the woman

(1B) ***NN*** $\rightarrow$ Frau | woman

(1C) ***VBN+NP*** $\rightarrow$ ein UFO gesehen | seen a UFO

(1D) ***S*** $\rightarrow$ ***NP*** , die ***VBN+NP*** hat ***VP*** . | ***NP*** who has ***VBN+NP VP*** .

S2: ich glaube , die Frau hat ein UFO gesehen . |
I think the woman has seen a UFO .

(2A) ***NP+V*** $\rightarrow$ ich glaube | I think

(2B) ***S*** $\rightarrow$ ***NP+V*** , die ***NN*** hat ***VBN+NP*** . | ***NP+V*** the ***NN*** has ***VBN+NP*** .

Test sentence: ich glaube , die Frau hat ein UFO gesehen .
$\xrightarrow{2B+2A+1B+1C}$ I think the woman has seen a UFO .

Figure 1.0.2: The syntax-augmented MT model applied to the training corpus from Figure 1.0.1.

In Chapter 4, we propose several improvements to the hierarchical phrase-based MT model of Chiang (2005) and its syntax-based extension by Zollmann and Venugopal (2006). We add a source-span variance model that, for each rule utilized in a probabilistic synchronous context-free grammar (PSCFG) derivation, gives a confidence estimate in the rule based on the number of source words spanned by the rule and its substituted child rules, with the distributions of these source span sizes estimated during training time. We further propose different methods of combining hierarchical and syntax-based PSCFG models, by merging the grammars as well as by interpolating the translation models. Finally, we compare syntax-augmented MT, which extracts rules based on target-side syntax, to a corresponding variant based on source-side syntax, and experiment with a model extension that jointly takes source and target syntax into account.

As is the case with phrase-based MT models, SAMT relies on word alignments of the training sentence pairs, which it assumes to be correct. In reality, these alignments are not human-generated, but instead result from the most probable configuration of a generative latent-variable model. Similarly (but unique to SAMT), the syntactic parses used for the grammar induction are 1-best results returned from a parser and thus prone to errors. In Chapter 5, we provide a method to induce grammars over hidden alignments and parses, approximated from N-best lists. We present results showing improvements for hierarchical phrase-based MT as well as SAMT when using the widened pipeline.

The SAMT model presupposes the availability of phrase-structure parse trees for the target training sentences. However, syntactic parsers are only available for a limited set of languages. In Chapter 6, we propose a label-

ing approach that is based merely on part-of-speech analysis of the source or target language (or even both). We achieve improvements in translation quality over Chiang's hierarchical phrase-based MT model on a large-scale NIST Chinese-to-English translation task. These improvements persist when using automatically learned word tags, suggesting broad applicability of our technique across diverse language pairs for which syntactic resources are not available. We further introduce a more flexible labeling approach based on K-means clustering, which allows the incorporation of an arbitrary number of word-class based features, including phrasal contexts, can make use of multiple tagging schemes, and also allows non-class features such as phrase sizes.

All code written for this dissertation work is made available as part of the open-source SAMT toolkit, co-written with Ashish Venugopal and available at:

`www.cs.cmu.edu/~zollmann/samt`

The remainder of this document is structured as follows: Chapter 2 establishes the necessary background by discussing phrase-based and hierarchical phrase-based machine translation. Chapters 3 to 6 present our contributions, as outlined above. Chapter 7 gives a summary of the contributions of this work.

CHAPTER 2

Background

2.1 Statistical machine translation

Given a source language sentence $\boldsymbol{f}$, most current statistical machine translation (SMT) approaches define the translation task as selecting the translation $\mathrm{tgt}(\hat{\boldsymbol{d}})$ represented by the most likely derivation[1] $\hat{\boldsymbol{d}}$ under a model $P(\boldsymbol{d}|\boldsymbol{f})$, i.e.:

$$\hat{\boldsymbol{d}} = \underset{\boldsymbol{d} \in \mathcal{D}.\, \mathrm{src}(\boldsymbol{d}) = \boldsymbol{f}}{\arg\max} \; P(\boldsymbol{d}|\boldsymbol{f}) \;, \tag{2.1.1}$$

which is accomplished by a search through a structured space $\mathcal{D}$ of translation hypotheses. Here, a derivation is a sequence of translation units: phrase pairs in the case of phrase-based SMT, grammar rules in the case of synchronous-grammar based SMT. Not all sequences of translation units are valid derivations; for example, in the case of probabilistic synchronous context-free grammars the rules in the sequence must correspond to successive substitutions of each rule into the source-left-most nonterminal pair of the partial derivation produced so far. This will become clear later in this chapter.

[1] Some approaches instead sum over derivations representing the same translation.

Commonly, a log-linear model of the form

$$P(\boldsymbol{d}|\boldsymbol{f}) = \frac{1}{Z(\boldsymbol{\lambda})} \prod_{i=1}^{m} \phi_i(\boldsymbol{d})^{\lambda_i} \tag{2.1.2}$$

is employed, where $\phi_i(\boldsymbol{d})$ are bilingual features of $\boldsymbol{d}$ and monolingual features of $\mathrm{tgt}(\boldsymbol{d})$, and weights λ_i are trained discriminatively to maximize translation quality (based on automatic metrics) on held-out data. Most loss functions used in MT are piecewise linear with respect to a single parameter λ_i, whence coordinate descent can be applied using a simple line intersection method known as minimum-error-rate training (MERT) (Och, 2003). Other optimizations proposed for MT are sampling the error surface (Venugopal, Zollmann, and Waibel, 2005), minimum-risk annealing (Smith and Eisner, 2006), or online large-margin training (Watanabe, Suzuki, Tsukada, and Isozaki, 2007; Chiang, Marton, and Resnik, 2008).

Most SMT approaches make independence assumptions to structure this search space and thus most features $\phi_i(\mathbf{d})$ are designed to be local to each phrase pair or rule. A notable exception is the n-gram language model (LM), which evaluates the likelihood of the sequential target words output. Phrase-based systems also typically allow source segments to be translated out of order, and include distortion models to evaluate such operations. These features suggest the efficient dynamic programming algorithms for phrase-based systems described in Koehn et al. (2004).

2.2 Phrase-based MT

Phrase-based methods (Och, Tillmann, and Ney, 1999) identify contiguous bilingual phrase pairs based on automatically generated word alignments. Phrase pairs are extracted up to a fixed maximum length, since very long

phrases rarely have a tangible impact during translation (Koehn et al., 2003). During decoding, extracted phrase pairs are reordered to generate fluent target output. Reordered translation output is evaluated under a distortion model and corroborated by one or more n-gram language models. These models do not have an explicit representation of how to reorder phrases. To avoid search space explosion, most systems place a limit on the distance that source segments can be moved within the source sentence. This limit, along with the phrase length limit (where local reorderings are implicit in the phrase), determine the scope of reordering represented in a phrase-based system.

Phrase-based systems typically include in their log-linear model a target language model, a distortion model capturing the reordering of translated phrases, and the following six features, each factored into a product of phrasal components over the phrase pairs p applied in the translation process:

- $\hat{p}(p|\,\mathrm{src}(p))$: probability of a phrase pair given its source side;
- $\hat{p}(p|\,\mathrm{tgt}(p))$: probability of a phrase pair given its target side;
- $\hat{p}_w(p|\,\mathrm{src}(p)), \hat{p}_w(p|\,\mathrm{tgt}(p))$: translation model probabilities estimated based on word based models (Brown et al., 1993)
- $\exp(|\,\mathrm{tgt}(p)|)$: a word count feature to trade off shorter vs. longer translations
- $\exp(1)$: a phrase count to prefer translations with fewer or more segments

In our notation, src returns the source side of a phrase pair, and tgt returns the target side.

2.3 Probabilistic synchronous context-free grammars

Probabilistic synchronous context-free grammars (PSCFGs) (Aho and Ullmann, 1969) are defined by a source terminal set (source vocabulary) $\mathcal{T}_S$, a target terminal set (target vocabulary) $\mathcal{T}_T$, a shared nonterminal set $\mathcal{N}$ and induce rules of the form

$$X \rightarrow \langle \gamma, \alpha, \sim, w \rangle$$

where

- $X \in \mathcal{N}$ is a nonterminal (called the *left-hand side* of the rule),
- $\gamma \in (\mathcal{N} \cup \mathcal{T}_S)^*$ is a sequence of nonterminals and source terminals (called the *source right-hand side* or simply *source side*),
- $\alpha \in (\mathcal{N} \cup \mathcal{T}_T)^*$ is a sequence of nonterminals and target terminals (called the *target right-hand side* or *target side*),
- the count $\#\mathrm{NT}(\gamma)$ of nonterminal tokens in γ is equal to the count $\#\mathrm{NT}(\alpha)$ of nonterminal tokens in α,
- $\sim$: $\{1, \ldots, \#\mathrm{NT}(\gamma)\} \rightarrow \{1, \ldots, \#\mathrm{NT}(\alpha)\}$ is a one-to-one mapping from nonterminal tokens in γ to nonterminal tokens in α, and
- $w \in [0, \infty)$ is a nonnegative real-valued weight assigned to the rule.

In our notation, we will assume $\sim$ to be implicitly defined by indexing the NT occurrences in γ from left to right starting with 1, and by indexing the NT occurrences in α by the indices of their corresponding counterparts in γ.

PSCFG derivations function analogously to context-free grammar (CFG) derivations, and can be used to express probabilistic hypotheses for the task of

machine translation analogously to monolingual parsing with a probabilistic context-free grammar (PCFG): We find the most likely derivation $\hat{d}$ that has the input source sentence f as its source yield, but where we are free to range over all possible target yields.

$$\hat{d} = \arg\max_{d:\mathrm{src}(d)=f} p(d) \tag{2.3.1}$$

where $\mathrm{src}(\cdot)$ maps a derivation to its source yield. Finally we read off the English translation from this derivation:

$$\hat{e} = \mathrm{tgt}\left(\hat{d}\right) \tag{2.3.2}$$

where $\mathrm{tgt}(\cdot)$ maps a derivation to its target yield.

The distribution p over derivations can be defined by a log-linear model. The probability of a derivation D is defined in terms of the rules r that are used in D:

$$p(D) = \frac{p_{\mathrm{LM}}(\mathrm{tgt}(D))^{\lambda_{\mathrm{LM}}} \times \prod_{r \in D} \prod_i \phi_i(r)^{\lambda_i}}{Z(\lambda)} \tag{2.3.3}$$

where ϕ_i is a feature function on rules, p_{LM} is an n-gram probability of the target yield $\mathrm{tgt}(D)$, and $Z(\lambda)$ is a normalization constant chosen such that the probabilities sum up to one.[2]

Performing translation with PSCFG grammars containing only rules with no more than two nonterminal pairs on their right hand sides amounts to straight-forward generalizations of the CYK (Kasami, 1965) chart parsing algorithm for PCFG grammars. In contrast to PCFGs, however, PSCFGs are not generally reducible to a 2-NT normal form. Nevertheless, decoding time

[2]Note that we never need to actually compute $Z(\lambda)$ since we are merely interested in the maximum-probability derivation.

cubic in sentence length can still be achieved, for example by using a synchronous version of the CYK+ (Chappelier and Rajman, 1998) algorithm, itself a variant of the Earley algorithm (Earley, 1970) that parses arbitrary PCFGs in cubic time. Adaptations to the algorithms in the presence of n-gram LMs are discussed in (Chiang, 2007; Venugopal, Zollmann, and Vogel, 2007; Huang and Chiang, 2007; Zollmann, Venugopal, Och, and Ponte, 2008a).

The use of PSCFGs for statistical machine translation was first proposed by Wu (1997), who induce an inversion transduction grammar (ITG) from a parallel training corpus, where the right-hand-sides of the rules could have either only terminals (at most one each for source and target portion) or only nonterminals (exactly two nonterminal pairs). The former rules thus represent the possible lexical translations, and the latter rules allow to either monotonously glue the two partial translations represented by the two right-hand-side target nonterminals or to swap them.

2.4 Hierarchical phrase-based MT

Building upon the success of phrase-based methods, Chiang (2005) presents a model of translation that uses the bilingual phrase pairs of phrase-based MT as starting point to learn phrases containing gaps—so-called *hierarchical phrases*. Formally, these phrases are rules of a PSCFG with a single generic nonterminal symbol X, and an auxiliary nonterminal symbol S that is used to realize the glue operations explained below.

In Chiang's rule extraction method, for each training sentence pair a set of PSCFG rules is generated from a set of extracted phrase pairs as follows: First, each phrase pair is assigned the X-nonterminal as left-hand-side, mak-

ing it an *initial rule*. We can now recursively generalize each already obtained rule (initial or including nonterminals)

$$N \rightarrow f_1 \dots f_m \mid e_1 \dots e_n$$

for which there is an *initial* rule

$$M \rightarrow f_i \dots f_u \mid e_j \dots e_v$$

where $1 \leq i < u \leq m$ and $1 \leq j < v \leq n$, to obtain a new rule

$$N \rightarrow f_1^{i-1} X_k f_{u+1}^m \mid e_1^{j-1} X_k e_{v+1}^n$$

where e.g. f_1^{i-1} is short-hand for $f_1 \dots f_{i-1}$, and where k is an index for the nonterminal X that indicates the one-to-one correspondence between the new X tokens on the two sides (it is not in the space of word indices like i, j, u, v, m, n).

During decoding, Chiang allows application of all rules of the grammar for chart items spanning up to a fixed number of source words. When that limit is reached, only a special glue rule allowing monotonic concatenation of hypotheses is allowed, thus making decoding time asymptotically linear instead of cubic in sentence length. Such a limit is also employed during training, with the effect of only generalizing phrases up to a certain maximum length.

Chiang (2005) uses the same features as in phrase-based translation (cf. Section 2.2), with the addition of a binary glue rule feature that fires only for glue rule applications.

In contrast to a phrase-based model, hierarchical MT allows for translation of discontiguous words (e.g. French "ne ... pas") as a unit, thereby resulting in a hypothesis space that is a strict superset of that of phrase-based

MT. In practice, however, nearly all 1-best outputs of a hierarchical system turn out to be also reachable by a corresponding phrase-based system trained on the same training data; cf. Zollmann et al. (2008a) for an analysis on Chinese-to-English, and Auli, Lopez, Hoang, and Koehn (2009) for a more detailed analysis on French-to-English, German-to-English, and English-to-German, which also investigates the impact of unaligned word handling. The main benefit of hierarchical over phrase-based MT therefore stems from its statistical model, not from the increased hypothesis space.

Chiang's work has been preceded and succeeded by a plethora of other PSCFG approaches to machine translation. These will be discussed in Section 3.6.

2.5 Evaluation

In principle, machine translation is best evaluated by human judgments, directly comparing different systems' outputs against each other and against one or several reference translations. However, this methodology is expensive in practice because of the human in the loop. Automatic evaluation metrics such as word error rate borrowed from the speech recognition community became popular in the 1990's, but did not correlate well with human judgments. A breakthrough metric with much improved correlation was the Bilingual Evaluation Understudy (BLEU) of Papineni, Roukos, Ward, and Zhu (2002), which is still the most popular translation evaluation metric today and is therefore the metric of choice throughout this work. BLEU computes test-corpus-level n-gram (where n ranges from 1 to 4) precision rates, combines these by taking the geometric mean, and then multiplies the result by a brevity penalty preventing MT systems from obtaining perfect scores by

providing empty translations. BLEU has been criticized of favoring phrase-based models due to the n-gram base approach. Another issue is its inability to give credits to output words that are semantically equivalent to the corresponding references but do not appear in any reference. The approach of Banerjee and Lavie (2005) overcomes that problem by additionally allowing for stemming and synonymy matching.

2.6 Discussion

While non-statistical approaches based on hand-crafted rules were the dominating approach in commercial MT systems until mid way though the 2000's decade, phrase-based SMT has now become the de-facto standard. Despite of the superior translation quality of Chiang's hierarchical approach for certain language pairs and its popularity in the scientific community, it has not been able to dethrone phrase-based MT from its preeminence in commercial systems. This is chiefly due to its far higher decoding memory and time requirements. As rules are extracted recursively by carving out phrase pairs from other phrase pairs, the number of extracted grammar rules is in the order of s^K, where s is the number of phrase pairs in a corresponding phrase-based system and K the maximum number of allowed abstractions per rule. While decoding time complexity in principle is exponential in sentence length for the phrase-based model, in practice it is kept linear by limiting reordering to occur only within a fixed window. In contrast, decoding with the hierarchical model, which amounts to parsing the PSCFG, is cubic in sentence length in principle, and also becomes linear when fixing a maximum window size for non-glue rule applications (which corresponds to limiting reordering). However, due to high grammar size, decoding is usually orders of magnitudes

slower in practice.

As we will see later, our proposed methods suffer from the same curse, and in an even higher degree because of greater grammar constants. However, as has been proved time and time again in the field of computer science, even solutions that at the time of proposal seemed utterly practically infeasible (which we will demonstrate is not the case with our methods by providing empirical results on large-scale translation tasks), became viable only a few years later, either because of newly discovered smart approximation techniques or because of exponential improvements in computational resources due to Moore's law. Therefore, our aim in this thesis is to concentrate on devising novel models that improve translation quality, while worrying less about computational efficiency.

CHAPTER 3

Syntax-Augmented Machine Translation

The use of a single grammar nonterminal symbol (not counting the glue rule nonterminal) in the hierarchical phrase-based MT model of Chiang (2005) makes the model agnostic to the types of phrase pairs that can be substituted in a grammar derivation. As in traditional phrase-based MT, the task of distinguishing grammatically coherent from incoherent translations is left solely to the language model.

In practice, Chiang also restricts the grammar by allowing only two generalizations within a single rule and discarding rules which contain adjacent generalizations. These restrictions amongst others described are designed to compensate for the use of a single generalization category. It is easy to see why they are necessary. Every phrase is marked with the same category X, allowing it to fill in any generalization of a phrase above it in the hierarchy. Without the knowledge of syntactic categories to restrict possible hierarchical combinations, these restrictions are required to make parsing efficient, at the expense of representational ability in the grammar.

In this chapter, we consider the scenario where we have access to a target language parser to annotate and guide the generalization of the derived synchronous grammar. By associating target language parse trees with their corresponding search lattices built by lexical phrases (trained using traditional phrase extraction techniques (Koehn et al., 2004)), we assign syntactic cate-

gories to phrases that align directly with the parse hierarchy. We also introduce syntax-derived categories that represent partially matched syntactic categories, thereby annotating every phrase in the initial phrase table. Our techniques produce grammars with several thousand unique nonterminals; therefore an efficient decoding algorithm and effective pruning strategies are crucial to the success of our translation system. Our work addresses specific issues with inducing a grammar directly from parallel text, but does not move towards the work of (Yamada and Knight, 2002), where linguistic structures and motivation drive even the operation of the parsing process.

Most of the work in this chapter has been published in (Zollmann and Venugopal, 2006; Zollmann, Venugopal, Vogel, and Waibel, 2006; Zollmann, Venugopal, and Vogel, 2007; Zollmann et al., 2008a; Zollmann, Venugopal, and Vogel, 2008b; Venugopal and Zollmann, 2009). All of this work is part of this thesis contribution.

3.1 Rule extraction

SAMT extends the purely hierarchical grammar proposed in (Chiang, 2005) to use nonterminal labels learned from target language parse trees. The inputs to the SAMT rule extraction procedure are tuples, $\langle f, e, \text{Phrases}(a, f, e), \pi \rangle$, where f is a source sentence, e is a target sentence, a is a word-to-word alignment associating words in f with words in e, $\text{Phrases}(a, e, f)$, are the set of phrase pairs (source and target phrases) consistent with the alignment a (Koehn et al., 2003; Och and Ney, 2004), and π is a phrase structure parse tree of e. SAMT rule extraction associates each phrase pair from $\text{Phrases}(a, e, f)$ with a left-hand-side label, and then applies the rule generalization procedure from (Chiang, 2005) to generate complex rules with *labeled* nonterminal

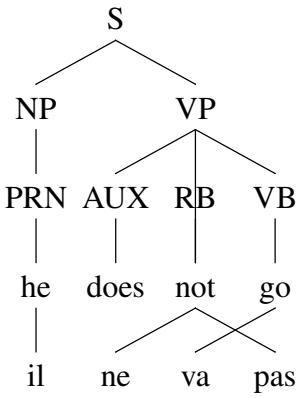

Figure 3.1.1: Alignment graph (word alignment and target parse tree) for a French-English sentence pair.

symbols on the right hand side.

Consider a French-to-English example sentence with alignment graph (a word alignment and target language parse tree as defined in (Galley et al., 2004)) given in Figure 3.1.1. The phrase extraction method from (Koehn et al., 2003) extracts all phrase pairs where no word inside the phrase pair is aligned to a word outside the phrase pair. The following phrase-pairs would be extracted for our example sentence:

il	|	he
va	|	go
ne va pas	|	does not go
ne va pas	|	not go
il ne va pas	|	he does not go

SAMT now assigns a left-hand-side label to every phrase pair extracted from the current sentence-pair, based on the corresponding target language

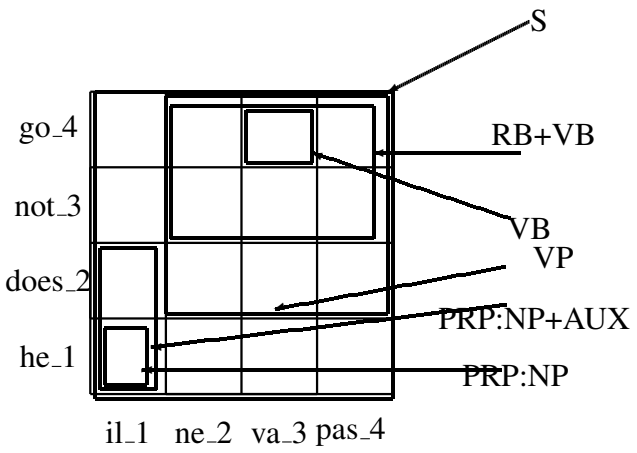

Figure 3.1.2: Spans of initial lexical phrases w.r.t. f, e. Each phrase is labeled with a category derived from the tree in Fig. 3.1.1.

parse tree π, forming *initial rules* (Figure 3.1.2. These labels are assigned based on the constituent spanning the target side word sequence in π. When the target side of the phrase-pair is spanned by a single constituent in π, the constituent label is assigned as the label of the phrase pair. If the target side of the phrase is not spanned by a single constituent in π, we use the labels of subsuming, subsumed, and neighboring constituents in π to assign an extended label of the form $C_1 + C_2$, C_1/C_2, or $C_2\backslash C_1$ (the latter two being inspired by the operations in combinatory categorial grammar (CCG) (Steedman, 2000)), indicating that the phrase pair's target side spans two adjacent syntactic categories (e.g., *she went: NP+VB*), a partial syntactic category C_1 missing a C_2 at the right (e.g., *the great: NP/NN*), or a partial C_1 missing a C_2 at the left (e.g., *great wall: DT\NP*), respectively. The label assignment is attempted in the order just described, i.e., assembling labels based on '+' concatenation of two subsumed constituents is preferred, as smaller constituents tend to be more accurately labeled. If no label is assignable by either of these

three methods, a default label '_FAIL' is assigned.

An ambiguity arises when unary rules $N_1 \rightarrow ... \rightarrow N_m$ in the target parse tree are encountered, such as the NP→PRP subtree in Figure 3.1.1. In this case, we use a combined label $N_m : \ldots : N_1$.

Based on the obtained labeled initial rules, we now perform the rule generalization procedure from (Chiang, 2005), resulting in the following complex rules extracted from our example sentence:

$$
\begin{aligned}
S &\rightarrow \text{PRP:NP}_1 \text{ ne va pas } \mid \text{PRP:NP}_1 \text{ does not go} \\
S &\rightarrow \text{il ne VB}_1 \text{ pas } \mid \text{he does not VB}_1 \\
S &\rightarrow \text{il VP}_1 \mid \text{he VP}_1 \\
S &\rightarrow \text{il RB+VB}_1 \mid \text{he does RB+VB}_1 \\
S &\rightarrow \text{PRP:NP}_1 \text{ VP}_2 \mid \text{PRP:NP}_1\text{VP}_2 \\
S &\rightarrow \text{PRP:NP}_1 \text{ RB+VB}_2 \mid \text{PRP:NP}_1 \text{ does RB+VB}_2 \\
\text{VP} &\rightarrow \text{ne VB}_1 \text{ pas } \mid \text{does not VB}_1 \\
\text{RB+VB} &\rightarrow \text{ne VB}_1 \text{ pas } \mid \text{not VB}_1 \\
\text{VP} &\rightarrow \text{RB+VB}_1 \mid \text{does RB+VB}_1
\end{aligned}
$$

Under the rectangle representation of phrase pairs from Fig. 3.1.2, generalization can be viewed as a process that selects a rectangle, and proceeds to subtract out one or more sub-rectangles to form a generalized rule.

We also add the following system rules to the grammar:

- Beginning-of-sentence rule: S→$< s >$ | $< s >$
- Glue rules similar to Chiang (2005) for each NT N in the grammar: S → $\text{S}_1 N_2$ | $\text{S}_1 N_2$

- End-of-sentence rule: S $\rightarrow$ S$_1$ $< /s >$ | S$_1$ $< /s >$
- 'Unknown'-rules (e.g. NNP $\rightarrow$ _UNKNOWN | _UNKNOWN) generating a limited set of labels for the word '_UNKNOWN', which the decoder substitutes for unknown source words

The design of the S-rules above anchors the glue operation to the beginning- and end-of-sentence and ensures that glue derivations are always left-branching, thereby avoiding spurious ambiguity. These glue operations allow the system to produce translations that violate the syntactic constraints encoded in the labels of the grammar—at a cost determined by the corresponding feature weight λ_{glue} (see Section 3.2).

The number of rules generated by this procedure is exponential in the number of initial phrase pairs, producing a grammar that is impractical for efficient translation. The following pruning parameters are used to restrict the number of rules extracted per sentence:

- *max_abstraction_count* (default: 2): maximum number of abstractions (nonterminal pairs) on a rule's right-hand-side.
- *max_source_symbol_count* (default: 5): maximum number of symbols (terminals and nonterminals) on the source side of the rule.
- *allow_consec_nts* (default: 1): if set to 0, discards rules that have consecutive nonterminals on the source side.
- *allow_src_abstract* (default: 1): if 0, discards rules that do not have any source terminal symbols, for example: S $\rightarrow$ NP $_1$VP$_2$ | NP $_2$VP$_1$. Setting this parameter to 0 drastically reduces decoding time.

- *nonlexminfreq, lexminfreq* (defaults: 0): minimum frequency (i.e., number of occurrences in the training data) thresholds for non-lexical and lexical rules respectively. Increasing these thresholds reduces the size of the grammar, but often at the cost of translation quality (Zollmann et al., 2008a).

- *min_freq_given_src_arg* (default: 0): minimum relative frequency of a rule given its labeled source.

- *allow_dangling_second_nt* (default: false): Whether to allow creation of rules with at least two right-hand-side nonterminal pairs in which one of the pairs is at the left or right phrase pair boundary (i.e., source NT is at beginning/end of source side of rule and corresponding target NT is at beginning/end of target side of rule, respectively). Setting this to false drastically speeds up decoding by prohibiting multi-NT-pair rules with dangling NTs. (Note, however, that all two-NT-pair rules doing reordering are kept, since in such a case the source and target parts of the NT pair could never be both at the beginning or both at the end of the phrase.)

3.2 SAMT Features

The labeling and extraction procedures defined above identify rules from the input word-aligned parallel corpora and associated parse trees. The occurrence counts from this extraction process are used in estimating the following features for each rule:

- $\hat{p}(r|\,\mathrm{lhs}(r))$: Relative frequency of a rule given its left-hand-side label

- $\hat{p}(r|\,\mathrm{src}(r))$: Relative frequency of a rule given its source side
- $\hat{p}(r|\,\mathrm{tgt}(r))$: Relative frequency of a rule given its target side
- $\hat{p}(r|\,\mathrm{ul}(\mathrm{src}(r)))$: Relative frequency of a rule given its un-labeled source side
- $\hat{p}(r|\,\mathrm{ul}(\mathrm{tgt}(r)))$: Relative frequency of a rule given its un-labeled target side
- $\hat{p}(\mathrm{ul}(\mathrm{tgt}(r))|\,\mathrm{ul}(\mathrm{src}(r))$: Relative frequency of the un-labeled target side of the rule given its un-labeled source side, i.e., marginalizing over left-hand-side and right-hand-side labels
- $\hat{p}(\mathrm{ul}(\mathrm{src}(r))|\,\mathrm{ul}(\mathrm{tgt}(r))$: Relative frequency of the un-labeled source side of the rule given its un-labeled target side

where lhs returns the left-hand-side of a rule, src returns the source side, tgt returns the target side, and ul removes all *labels* from nonterminal symbols. For example, $\mathrm{ul}(\text{NP+AUX}_1\text{does not go}) = \text{X}_1$ does not go.

To estimate the features above, we use relative frequency estimation based on counts of the rules extracted from the training data. For example, $p(r|\,\mathrm{lhs}(r))$ is estimated by computing $\#(r)/\,\#(\mathrm{lhs}(r))$, aggregating counts from all extracted rules.

We also add the following features to the model:

- $\hat{p}_w(\mathrm{src}(r)|\,\mathrm{tgt}(r)), \hat{p}_w(\mathrm{tgt}(r)|\,\mathrm{src}(r))$: lexical weights based on terminal symbols as for phrase-based and hierarchical phrase-based MT (cf. Chapter 2)

- $\phi_{glue}(r) = \exp(1)$ if the rule is a glue rule, 1 otherwise; as for hierarchical phrase-based MT

- $\phi_{ra}(r) = \exp(1)$ for all rules. A rule application count, allowing the model to favor derivations with more or less rules depending on the weight assigned to this feature, analogously to phrase-based and hierarchical phrase-based MT

- $\phi_{tgt}(r) = \exp($'# target terminals in r'. Allows the model to prefer longer or shorter translations, analogously to phrase-based and hierarchical phrase-based MT

- $\phi_{lex}(r) = \exp(1)$ if the rule's right-hand-side has no nonterminals, 1 otherwise.

- $\phi_{abs}(r) = \exp(1)$ if the right-hand-side has no terminals, 1 otherwise.

- $\phi_{srcadj}(r) = \exp(1)$ if the source side has adjacent nonterminals, 1 otherwise. Allows the model to indicate confidence in derivations that include multiple sequential nonterminals.

- $\phi_{tgtadj}(r) = \exp(1)$ if the target side has adjacent nonterminals, 1 otherwise.

- $\phi_{bal}(r) = \exp(|\log(R(r)/\hat{R})|)$ where $R(r)$ the ratio of source to target terminals in r and $\hat{R}$ is the same ratio measured over sentences in the corpus.

- $\phi_{mono}(r) = \exp(1)$ if the rule does not re-order its nonterminals, 1 otherwise.

- $\phi_{rare}(r) = \exp((1/\,\#(r)))$: uses the number of times a rule has been seen during training, $\#(r)$, to allow penalization of derivations that use rare rules.

3.3 Decoding

Apart from the issues to consider when parsing PSCFGs mentioned in Section 2.3, multi-nonterminal PSCFGs face the additional challenge of needing to keep hypotheses stemming from rules with different left-hand-side nonterminals in separate equivalence classes in the chart in order to guarantee obtaining the most probable derivation. In practice, some pruning across equivalence classes representing different nonterminals is necessary in order to keep memory and decoding time requirements at a reasonable level. Solutions to this problem are given by Venugopal (2008).

3.4 Large-scale training and decoding with MapReduce

Computing clusters with many parallel processors have become increasingly available to the research community. In 2008, for example, Yahoo! made a 4000-processor cluster called M45 available to universities, which runs MapReduce (Dean and Ghemawat, 2004) jobs under the Hadoop (Cutting and Baldeschwieler, 2007) architecture. For machine translation systems to benefit from large clusters like these, training and decoding have to be parallelized. In this section, we show how this can be achieved for SAMT under the MapReduce paradigm. To our knowledge, this contribution is the first to show how to do training and decoding with a PSCFG-based SMT system using MapReduce.

3.4.1 MapReduce

Given a cluster of machines, there are several solutions to exploit these resources for computational work. Systems like Condor (Thain, Tannenbaum, and Livny, 2005) and Sun's Grid Engine, provide coarse-grained job management (accepting, scheduling, dispatching, and managing the remote execution) to a cluster of machines. These systems are primarily responsible for managing the smooth execution of jobs submitted to the cluster, while placing minimal constraints on the nature of the running jobs.

Alternatively, the MapReduce (Dean and Ghemawat, 2004) architecture is a programming model where large computational tasks are split into two distinct phases, a Map phase and a Reduce phase. In the Map phase, unstructured input data is processed by parallel tasks generating intermediate output in the form of key-value pairs. In the Reduce phase, tasks running in parallel receive this intermediate data with the guarantee that each process will receive all intermediate key-value pairs that share the same key. Under this framework, large computational tasks and task pipelines (like identifying and estimating parameters for SAMT rules and running decoding and MERT) can be distributed to run on a cluster of commodity hardware.

3.4.2 The SAMT pipeline

For each MapReduce phase of the pipeline, we specify the MapInput (data received by the Map task), MapOptions (parameters to the Map task), MapOutput (key-value pairs output by the Map task), ReduceInput (input guaranteed to be contiguous to the Reduce task), ReduceOptions (parameters to the Reduce task), and ReduceOutput (unstructured output format from the Reduce

task). SAMT assumes input of the format $e, f, a(e, f), \pi(e)$, where e is a target language sentence from the training data, f is a source language sentence from the training data, $a(e, f)$ is a word-to-word alignment (Brown et al., 1993) on e, f and $\pi(e)$ is phrase structure parse tree on e. Confer Dyer, Cordova, Mont, and Lin (2008a) on how to parallelize word alignment with MapReduce. We also parallelize the parsing of the training sentences, which can be achieved with a simple shell script acting as a mapper that calls the parser on its assigned chunk of sentences.

The SAMT pipeline can be split into the following phases: Phrase Extraction, Rule Extraction, Rule Filtering, LM filtering (optional), Decoding, N-Best Merge and MERT. In each phase we try to limit the number of key-value pairs to reduce I/O overhead, outputting multiple values that share the same key from the same Map task on a single line. The Rule Filtering and LM Filtering phases build sentence specific models for each sentence in the development and test corpus allowing the Decoding phrase to load these models directly into memory. We now describe each phase.

Phrase Extraction Map:

- MapInput: Input lines of the form $f, e, a(e, f), \pi(e)$
- MapOptions: Maximum extractable phrase length
- MapOutput:
  key $= \mathrm{sno}$, value $= \langle f, e, \mathrm{Phrases}(e, f), \pi(e) \rangle$
  where sno is the respective sentence number

The Phrase Extraction phase identifies $\mathrm{Phrases}(e, f)$ based on the word-aligned data and adds it to the training data stream. There is no Reduce step in this Phase.

Rule Extraction Map:

- MapInput: Each line contains $f, e, \text{Phrases}(e, f), \pi(e)$
- MapOptions: Maximum number of nonterminals per rule, maximum length of γ, options to select lhs from π
- MapOutput:

  key $= \text{ul}(\gamma)$, value $= \langle \gamma, \alpha, \text{lhs}, 1 \rangle$ and

  key $= \text{ul}(\alpha)$, value $= 1$ and

  key $= \text{lhs}$, value $= 1$

Rule Extraction uses its input to generate PSCFG rules via the procedure in Section 3.1, taking several parameters that constrain the grammar. MapOutput ouputs the unlabeled source side of each rule $\text{ul}(\gamma)$ as key, with the rule itself as value. Since the subsequent Reduce input will see rules grouped by $\text{ul}\,(\gamma)$, efficient computation of features $\hat{p}(r|\,\text{src}(r))$, $\hat{p}(\text{ul}(\text{tgt}(r))|\,\text{ul}(\text{src}(r)))$ is possible in the Reduce step. MapOutput also outputs occurrence statistics for each lhs and for each unlabelled target side of the rule in order to compute additional features in ϕ in later phases.

Rule Extraction Reduce:

- ReduceInput: All rules that share the same $\text{ul}(\gamma)$
- ReduceOptions: Minimum occurrence counts for lexical and nonlexical rules
- ReduceOutput: Rules with subset of features in ϕ. Rules that share the same $\text{ul}(\gamma)$ are output on the same line.

Features $\hat{p}(r|\,\text{src}(r))$, $\hat{p}(\text{ul}(\text{tgt}(r))|\,\text{ul}(\text{src}(r)))$ are computed in the Reduce step since all rules that share the same unlabelled source are available con-

tiguously to the Reduce step. Key-value pairs indicating lhs and $\mathrm{ul}(\alpha)$ are simply accumulated.

In the Rule Filtering phase, we select those rules that can possibly be applied to each sentence in the development and test corpus in the Map step, and in the Reduce step we take these rules, add special SAMT rules to handle unknown words and glue-rule (Chiang, 2005), resulting in a sentence specific PSCFG.

Rule Filtering Map:

- MapInput: Rules from Rule Extraction stage (single source as key with multiple rules as values)
- MapOptions: Source corpus to filter rules against (whole source corpus is loaded into memory)
- MapOutput:

  key $= \mathrm{sno}$

  value $= \langle \gamma, \alpha, \phi \rangle$

  such that all words in γ are in sentence sno in the source corpus

Count information for lhs and $\mathrm{ul}(\alpha)$ is keyed for every sentence. In the filtering step this count information is used to generate the remaining relative frequency features. Note that this can only be done at this point, because at the previous phase these counts were not yet available in accumulated form.

Rule Filtering Reduce:

- ReduceInput: All rules and special counts for a single test sentence
- ReduceOptions: Additional models to generate the remaining features ϕ
- ReduceOutput: Rules with fully formed ϕ for a single sentence. Rules for a particular sentence are written to a canonically named file.

The Rule Filtering phase outputs canonical per-sentence grammar files as a side-effect file on the distributed file system, rather than on the standard output stream. On the standard output stream we output the potential target vocabulary for each sentence based on the sentence-specific grammar.

LM Filtering is an optional phase to run when the n-gram language models used for decoding are too large to fit in memory. By using the potential target language vocabulary for each sentence, we can build sentence-specific n-gram language models which are much smaller without losing any relevant parameters.

LM Filtering Map:

- MapInput: Each line is a line from an ARPA format LM
- MapOptions: Access to a $\mathrm{sno} \rightarrow vocabulary$ map from the filtering stage (loaded into memory)
- MapOutput:

  key $= \mathrm{sno}$, value $= t_1 \cdots t_n$

  if every t_i is in the target vocabulary of sno.

The Map step selects relevant n-gram lines for each sentence, while the Reduce step re-builds a valid ARPA LM. Just like the Rule Filtering phase, LM Filtering produces canonically named sentence-specific language model files as side-effects on HDFS.

LM Filtering Reduce:

- ReduceInput: All n-grams that are compliant with a single sentence's vocabulary
- ReduceOuput: Statistics over n-grams are computed and output as a header to form a complete ARPA LM

The Decoding phase runs translation accessing the sentence specific translation and language models and outputting an n-best list for each sentence.

Decoding Map:

- MapInput: A single sentence to translate per line with sno information
- MapOptions: Options typically passed to a decoder to run translation. We also specify a path to a HDFS directory containing per-sentence grammars and language models.
- MapOutput: key $= \mathrm{sno}$, value $= N$-best list

If we are running Minimum Error Rate training (MERT), i.e., multiple iterations of development-set decoding and MERT parameter optimization, we perform an additional Merge phase that takes n-best list output from all iterations performed so far and runs a trivial MapReduce to merge n-best lists across iterations and remove duplicates. Minimum Error Rate training is implemented as a MapReduce task as well. We do not parallelize the inner-working of the MERT process, rather we simply allow multiple initial parameter configurations to be evaluated in parallel. In order to pass *different* parameters to each MERT task, we define MERT MapReduce as follows:

MERT Map:

- MapInput: N-Best lists for MERT optimization
- MapOptions: Multiple parameter conditions for MERT. Each parameter condition includes initial parameters to start MERT
- MapOutput:
  key = one MERT parameter config.
  value = all optimization data

In the Reduce step, each Reduce task receives all the N-best list data *and* parameters to run the optimization with, allowing each Reducer to run optimization with different parameters.

The Decoding and MERT phases are run subsequently until the number of new translations as a percentage of the total merged n-best list is below a certain threshold (in our experiments, 1%). This usually leads to around 10 to 15 iterations.

Our MapReduce framework has the advantages of scaling MT to huge training sets, thus overcoming the problem of limited per-processor memory and CPU speed. However, a drawback is that the test data must be known during training time (at least from the rule filtering phase onwards); thus, the framework is only suitable for batch-processing, not for online translation.

3.5 Empirical results

3.5.1 Experiments for a French-to-English translation task

We present experiments on the Europarl French-English task as defined at the NAACL 2006 workshop: Exploiting Parallel Texts for Statistical Machine Translation (Koehn and Monz, 2006). We compare a state-of-the-art phrase-based system against several degrees of modeling refinement within our system. All systems use the same initial phrase table (maximum phrase length 7) generated by the scripts provided for the workshop, described in (Koehn et al., 2003). The language model is also provided in the 2006 shared task, and is built on 13 million English words using Knesser-Ney smoothing. We evaluated our results using the BLEU metric (Papineni et al., 2002), optimizing the parameters on the first 500 sentences of the provided 'Develop-

ment Set' (identical to the 2005 development set), and testing on the provided 'Development Test Set' (identical to the 2005 test set). The threshold for statistical significance is 0.78 BLEU points at the 95 percent confidence level as calculated by (Zhang and Vogel, 2005).

The baseline phrase-based translation system is Pharaoh (Koehn et al., 2004), using the default settings specified by the provided minimum-error-rate training scripts (phrase pruning b=100, chart pruning = 1e-5, distortion limit=4, K-Best=100). Minimum Error Rate training is run for 13 iterations till convergence, compensating for the relatively smaller K-Best size compared to our experiments.

- Baseline - Pharaoh as described above
- Lex - Phrase-decoder simulation: using only the initial lexical rules from the phrase table, all with LHS X, and the glue rule. An additional re-ordering rule is added for swap based re-ordering and a feature is added to reflect this operation. Thus, adjacent phrases can swap during translation, and the resulting combined double-phrase can again swap with neighbors, and so on recursively, but the resulting space of possible reorderings is still a subset of that of a true phrase-based system.
- Hier - Hierarchical phrase-based MT, i.e., using only a single X nonterminal (besides the glue nonterminal); identical filtering to (Chiang, 2005)
- Syn - Syntactic extraction using the Penn Treebank parse categories as nonterminals; rules containing up to 4 nonterminal abstraction sites.
- SynExt - Syntactic extraction using the extended-category scheme, but with rules only containing up to 3 nonterminal abstraction sites (the

System	#Nonterminals	DevSet BLEU	TestSet BLEU
Baseline - max. phrase length 7	0	31.11	30.61
Lex - max. phrase length 7	2	28.96	29.12
Hier - max. phrase length 7	2	30.89	31.01
Syn - max. phrase length 7	75	31.52	31.31
SynExt - max. phrase length 7	3900	31.73	31.41
Baseline - max. phr. length 12	0	31.16	30.90
Lex - max. phr. length 12	2	29.30	29.51
Hier - max. phr. length 12	2	30.79	30.59
SynExt - max. phr. length 12	3900	31.07	31.76

Table 3.5.1: Translation results (IBM BLEU) for each system on the Fr-En '06 Shared Task 'Development Set' (used for MER parameter tuning) and '06 'Development Test Set' (identical to previous year's Shared Task's test set).

restriction to 3 nonterminals was necessary due to memory requirements).

We also explored the impact of longer initial phrases by training another phrase table with phrases up to length 12. The results based on the length-7 phrase table as well as the length-12 phrase table are presented in Table 3.5.1.

Our results show a statistically significant improvement of the Syn and SynExt system over the traditional phrase-based decoding system. We also see a clear trend towards improving translation quality as we employ richer extraction techniques. However, our results do not show as great an improvement over the baseline as Chiang (2005) reported on the Chinese-English Tides data. We believe that this is due to the difference in language pairs, French offers fewer opportunities to benefit from stronger and better informed

re-ordering models.

Note also that our decoding performance with the basic Lex system (which is essentially phrase-based) is significantly below par compared to direct beam based decoding. This is likely due to the limited reordering model: only binary swaps of two adjacent hypotheses in the chart are possible, and the reordering feature merely counts the number of such swaps, rather than being based on the reordering distance in terms of words as in Pharaoh's distortion model.

3.5.2 Experiments for a Spanish-to-English translation task

We participated with our SAMT system in the MT'07 Spanish-to-English shared task of the ACL 2007 Workshop on Statistical Machine Translation. We trained the system on the Spanish-English in-domain training data provided for the workshop. NIST-BLEU scores are reported on the 2K sentence development 'dev06' and test 'test06' corpora as per the workshop guidelines (case sensitive, de-tokenized). We compare our scores against the CMU-UKA ISL phrase-based submission, a state-of-the art phrase-based SMT system with part-of-speech (POS) based word reordering (Paulik, Rottmann, Niehues, Hildebrand, and Vogel, 2007).

3.5.2.1 Translation results

The SAMT system achieves a BLEU score of 32.48% on the 'dev06' development corpus and 32.15% on the unseen 'test06' corpus. This is slightly better than the score of the CMU-UKA phrase-based system, which achieves 32.20% and 31.85% when trained and tuned under the same in-domain con-

ditions. [1]

To understand why the syntax augmented approach has limited additional impact on the Spanish-to-English task, we consider the impact of reordering within our phrase-based system. Table 3.5.2 shows the impact of increasing reordering window length (Koehn et al., 2003) on translation quality for the 'dev06' data.[2] Increasing the reordering window past 2 has minimal impact on translation quality, implying that most of the reordering effects across Spanish and English are well modeled at the local or phrase level. The benefit of syntax-based systems to capture long-distance reordering phenomena based on syntactic structure seems to be of limited value for the Spanish to English translation task.

ReOrder	1	2	3	4	POS	**SAMT**
BLEU	31.98	32.24	32.30	32.26	32.20	**32.48**

Table 3.5.2: Impact of phrase-based reordering model settings compared to SAMT on the Spanish-to-English Shared Task 'dev06' corpus measured by NIST-BLEU.

3.5.3 Experiments on three NIST machine translation tasks

In the following experiments, published in Zollmann et al. (2008a), we compare the phrase-based MT system of the statistical MT research group at

[1] The CMU-UKA phrase-based workshop submission was tuned on out-of-domain data as well.

[2] Variant of the CMU-UKA ISL phrase-based system without POS based reordering. With POS-based reordering turned on, additional window-based reordering even for window length 1 had no improvement in NIST-BLEU.

Ch.-En. System \ %BLEU	Dev (MT04)	MT02	MT03	MT05	MT06	MT08	TstAvg
FULL							
Phraseb. reo=4	37.5	38.0	38.9	36.5	32.2	26.2	**34.4**
Phraseb. reo=7	40.2	40.3	41.1	38.5	34.6	27.7	**36.5**
Phraseb. reo=12	41.3*	41.0	41.8	39.4	35.2	27.9	**37.0**
Hier.	41.6*	40.9	42.5	40.3	36.5	28.7	**37.8**
SAMT	41.9*	41.0	43.0	40.6	36.5	29.2	**38.1**
TARGET-LM							
Phraseb. reo=4	35.9*	36.0	36.0	33.5	30.2	24.6	**32.1**
Phraseb. reo=7	38.3*	38.3	38.6	35.8	31.8	25.8	**34.1**
Phraseb. reo=12	39.0*	38.7	38.9	36.4	33.1	25.9	**34.6**
Hier.	38.1*	37.8	38.3	36.0	33.5	26.5	**34.4**
SAMT	39.9*	39.8	40.1	36.6	34.0	26.9	**35.5**
TARGET-LM, 10%TM							
Phraseb. reo=12	36.4*	35.8	35.3	33.5	29.9	22.9	**31.5**
Hier.	36.4*	36.5	36.3	33.8	31.5	23.9	**32.4**
SAMT	36.5*	36.1	35.8	33.7	31.2	23.8	**32.1**

Table 3.5.3: Results (% case-sensitive IBM-BLEU) for Ch-En NIST-large. Dev. scores with * indicate that the parameters of the decoder were MER-tuned for this configuration and also used in the corresponding non-marked configurations.

Google to a hierarchical phrase-based as well as a syntax-augmented MT system that use the same pre-processing, word alignment, phrase extraction, parameter tuning, and post-processing modules as the phrase-based system. The phrase-based system is based on Och and Ney (2004) but additionally features the lexicalized distortion model of Zens and Ney (2006).

Ar.-En. System \ %BLEU	Dev (MT04)	MT02	MT03	MT05	MT06	MT08	TstAvg
FULL							
Phraseb. reo=4	51.7	64.3	54.5	57.8	45.9	44.2	**53.3**
Phraseb. reo=7	51.7*	64.5	54.3	58.2	45.9	44.0	**53.4**
Phraseb. reo=9	51.7	64.3	54.4	58.3	45.9	44.0	**53.4**
Hier.	52.0*	64.4	53.5	57.5	45.5	44.1	**53.0**
SAMT	52.5*	63.9	54.2	57.5	45.5	44.9	**53.2**
TARGET-LM							
Phraseb. reo=4	49.3	61.3	51.4	53.0	42.6	40.2	**49.7**
Phraseb. reo=7	49.6*	61.5	51.9	53.2	42.8	40.1	**49.9**
Phraseb. reo=9	49.6	61.5	52.0	53.4	42.8	40.1	**50.0**
Hier.	49.1*	60.5	51.0	53.5	42.0	40.0	**49.4**
SAMT	48.3*	59.5	50.0	51.9	41.0	39.1	**48.3**
TARGET-LM, 10%TM							
Phraseb. reo=7	47.7*	59.4	50.1	51.5	40.5	37.6	**47.8**
Hier.	46.7*	58.2	48.8	50.6	39.5	37.4	**46.9**
SAMT	45.9*	57.6	48.7	50.7	40.0	37.3	**46.9**

Table 3.5.4: Results (% case-sensitive IBM-BLEU) for Ar-En NIST-large. Dev. scores with * indicate that the parameters of the decoder were MER-tuned for this configuration and also used in the corresponding non-marked configurations.

3.5.3.1 Chinese-English and Arabic-English

We report experiments on three data configurations. The first configuration (Full) uses all the data (both bilingual and monolingual) data available for the NIST 2008 large track translation task. The parallel training data comprises of 9.1M sentence pairs (223M Arabic words, 236M English words) for Arabic-English and 15.4M sentence pairs (295M Chinese Words, 336M English words) for Chinese-English. This configuration (for both Chinese-English and Arabic-English) includes three 5-gram LMs trained on the target side of the parallel data (549M tokens, 448M 1..5-grams), the LDC Gigaword corpus (3.7B tokens, 2.9B 1..5-grams) and the Web 1T 5-Gram Corpus (1T tokens, 3.8B 1..5-grams). The second configuration (TargetLM) uses a single language model trained only on the target side of the parallel training text to compare approaches with a relatively weaker n-gram LM. The third configuration is a simulation of a low data scenario (10%TM), where only 10% of the bilingual training data is used, with the language model from the TargetLM configuration. Translation quality is automatically evaluated by the IBM-BLEU metric (Papineni et al., 2002) (case-sensitive, using length of the closest reference translation) on the following publicly available NIST test corpora: MT02, MT03, MT05, MT06, MT08. We used the NIST MT04 corpus as development set to train the model parameters λ. For the purposes of stable comparison across multiple test sets, we additionally report a TstAvg score which is the average of all test set scores.[3]

Tables 3.5.3 and 3.5.4 show results comparing phrase-based, hierarchical

[3]We prefer this over taking the average over the aggregate test data to avoid artificially generous BLEU scores due to length penalty effects resulting from e.g. being too brief in a hard test set but compensating this by over-generating in an easy test set.

and SAMT systems on the Chinese-English and Arabic-English large-track NIST 2008 tasks, respectively. Our primary goal here is to evaluate the relative impact of the PSCFG methods above the phrase-based approach, and to verify that these improvements persist with the use of of large n-gram LMs. We also show the impact of larger reordering capability under the phrase-based approach, providing a fair comparison to the PSCFG approaches.

Chinese-to-English configurations: We see consistent improvements moving from phrase-based models to PSCFG models. This trend holds in both LM configurations (Full and TargetLM) as well as the 10%TM case, with the exception of the hierarchical system for TargetLM, which performs slightly worse than the maximum-reordering phrase-based system.

We vary the reordering limit "reo" for the phrase-based Full and TargetLM configurations and see that Chinese-to-English translation requires significant reordering to generate fluent translations, as shown by the TstAvg difference between phrase-based reordering limited to 4 words (34.4) and 12 words (37.0). Increasing the reordering limit beyond 12 did not yield further improvement. Relative improvements over the most capable phrase-based model demonstrate that PSCFG models are able to model reordering effects more effectively than the phrase-based approach, even in the presence of strong n-gram LMs (to aid the distortion models) and comparable reordering constraints.

Our results with hierarchical rules are consistent with those reported in Chiang (2007), where the hierarchical system uses a reordering limit of 10 (implicit in the maximum length of the initial phrase pairs used for the construction of the rules, and the decoder's maximum source span length, above which only the glue rule is applied) and is compared to a phrase-based system

with a reordering limit of 7.

Arabic-to-English configurations: Neither the hierarchical nor the SAMT system show consistent improvements over the phrase-based baseline, outperforming the baseline on some test sets, but underperforming on others. We believe this is due to the lack of mid- and long-range reordering phenomena between the two languages, as evident by the minimal TstAvg improvement the phrase-based system can achieve when increasing the reordering limit from 4 words (53.3) to 9 words (53.4).

N-Gram LMs: The impact of using additional language models in configuration Full instead of only a target-side LM (configuration TargetLM) is clear; the phrase-based system improves the TstAvg score from 34.6 to 37.0 for Chinese-English and from 50.0 to 53.4 for Arabic-English. Interestingly, the hierarchical system and SAMT benefit from the additional LMs to the same extent, and retain their relative improvement compared to the phrase-based system for Chinese-English.

3.5.3.2 Urdu-English

Table 3.5.5 shows results comparing phrase-based, hierarchical and SAMT system on the Urdu-English large-track NIST 2008 task. Systems were trained on the bilingual data provided by the NIST competition (207K sentence pairs; 2.2M Urdu words / 2.1M English words) and used a n-gram LM estimated from the English side of the parallel data (4M 1..5-grams). We see clear improvements moving from phrase-based to hierarchy, and additional improvements from hierarchy to syntax. As with Chinese-to-English, longer-distance reordering plays an important role when translating from Urdu to English (the phrase-based system is able to improve the test score from 18.1

System \ %BLEU	Dev	MT08
Phr.b. reo=4	12.8	**18.1**
Phr.b. reo=7	14.2	**19.9**
Phr.b. reo=10	14.8*	**20.2**
Phr.b. reo=12	15.0	**20.1**
Hier.	16.0*	**22.1**
SAMT	16.1*	**22.6**

Table 3.5.5: Translation quality (% case-sensitive IBM-BLEU) for Urdu-English NIST-large. We mark dev. scores with * to indicate that the parameters of the corresponding decoder were MER-tuned for this configuration.

to 20.2), and PSCFGs seem to be able to take this reordering better into account than the phrasal distance-based and lexical-reordering models.

3.6 Related work

There have been several previous proposals of using syntax to aid statistical machine translation. The data-oriented translation (DOT) model of Poutsma (2000) is an extension of data-oriented parsing paradigm (Scha, 1990), in which a probabilistic tree-substitution grammar is inferred from a treebank, to the case of a parallel treebank of pairs of source and target language phrase structure trees with linked sub-trees, from which a probabilistic synchronous tree-substitution grammar (PSTSG) is inferred. The task of translation then amounts to computing the most probable source-target parse-tree pair with the test sentence as its source yield, which is tractable, or computing the most probable target sentence by marginalizing over all tree pairs with identical

yields, which is intractable but can be approximated by sampling.

Yamada and Knight (2001) present a noisy-channel model transforming a target language phrase-structure parse tree into a source language sentence by applying the following stochastic operations on each tree node: child reordering, word insertion, and leaf node translation. The model can be cast as a top-down (also called *root-to-frontier*) tree transducer (Knight and Graehl, 2005), and is equivalent to a probabilistic context-free grammar model. Therefore, decoding amounts to CYK parsing, which will find the most probable target tree for a given source sentence in time cubic in sentence length (Yamada and Knight, 2002).[4]

The model of Galley et al. (2004) uses the formalism of tree-to-string transducers to model the transformation of a (target) phrase-structure tree into a (source) sentence, resulting in an efficient (linear in the number of parse tree nodes) algorithm to learn syntactic translation rules from data. Translation can thus be modeled in the traditional noisy-channel SMT approach (Brown, Cocke, Pietra, Pietra, Jelinek, Lafferty, Mercer, and Roossin, 1990) of maximizing the probability of the target tree multiplied by the probability of the source sentence given the target tree. The case of summing over trees to compute the most probable translation was considered for a scenario without language model by May and Knight (2006).

In work developed in parallel to ours, Galley, Graehl, Knight, Marcu, DeNeefe, Wang, and Thayer (2006) extend the work of Galley et al. (2004) to allow context into the rules, resulting in grammar rules of a restricted class of PSTSGs for which each rule's right-hand-side is flat (as in context-free

[4]The problem of finding the most probable target sentence, i.e., marginalizing over target trees representing the same translation, was not considered, but cf. also the next paragraph.

grammars) in the source part and a tree (as in tree-substitution grammars) in the target part. The model results in significant improvement in translation quality over the one of Galley et al. (2004), and in follow-up work, Marcu, Wang, Echihabi, and Knight (2006) manage to beat a phrase-based baseline with an improved model.

Another syntax-based model developed in parallel was the one of Liu, Liu, and Lin (2006), which extracts tree-to-string (i.e., source-side trees and flat target sides) translation rules from a parallel corpus with parsed source sentences. For translation, the source sentence is parsed with the same (external) parser used to parse the training source sentences, and then the resulting tree fragments are pattern-matched against the translation rules. The decoding algorithm is thus linear in sentence length; however, the system is inherently prone to errors in the parsing of the training as well as the test sentences, as the translation of a training corpus phrase can only be applied if the source subtree it spans during training is identical to the one it spans during translation. Mi, Huang, and Liu (2008) and Mi and Huang (2008) solve this problem with the efficient use of packed forests to represent alternative test sentence parses and training source sentence parses, respectively, achieving outperformance of phrase-based as well as hierarchical MT.

The work of Lavie, Parlikar, and Ambati (2008) follows Poutsma (2000) in using phrase-structure parse trees for both source and target training sentences. They provide a novel algorithm to align the source and target tree nodes, and then extract syntactic constituents from the training sentence pairs, resulting in a syntax-annotated phrase-table and synchronous context-free grammar rules.

Carreras and Collins (2009) present an approach to syntax-based MT

based on a variant tree-adjoining grammars (TAG) allowing for flexible reordering operations. Their formalism also enables direct integration of lexicalized syntactic language models. Hypothesis selection proceeds according to a log-linear combination of the models used in phrase-based SMT, with the addition of a syntactic language model and a dependency-based translation model, which is a discriminatively trained global linear model.

Hybrids of syntax-based and non-syntactic models are gaining popularity as well. Originally employed via N-best list system combination in MT evaluations, hybrid decoders have been proposed in recent work. Liu, Mi, Feng, and Liu (2009) devise a decoder integrating hierarchical phrase-based and tree-to-string models. Hanneman and Lavie (2009) present a hybrid of phrase-based and syntactic MT. Gimpel and Smith (2009) give a feature-based model that uses dependency syntax and phrases.

3.7 Conclusions and contributions

We presented syntax-augmented machine translation (SAMT), a novel statistical MT model over synchronous context-free grammars with multiple nonterminals. This was the first syntax-based MT system to achieve an improvement over phrase-based MT (Zollmann and Venugopal, 2006). We showed how to parallelize the system under the MapReduce paradigm, and reported experimental results comparing SAMT to phrase-based and hierarchical phrase-based MT for multiple language pairs. We reported improvements over these baselines for French-to-English, Chinese-to-English, and Urdu-to-English, but failed to obtain improvements for Spanish-to-English and Arabic-to-English. We draw the conclusion that SAMT (as well as hierarchical phrase-based MT) fails to outperform phrase-based MT for language

pairs that have mainly short-range word reordering. Indeed, Birch, Blunsom, and Osborne (2009) thoroughly substantiate this hypothesis for Arabic-to-English and Chinese-to-English by grouping the test set sentences into classes with low-range, mid-range, and high-range reordering, and then comparing phrase-based and hierarchical phrase-based performance, with the conclusion that phrase-based systems perform relatively better for low-range sentences, hierarchical phrase-based perform relatively better for medium-range sentences, and neither of the systems deal adequately with longer-range reordering.

CHAPTER 4

SAMT Extensions and Variations

In this chapter, we propose several improvements to the hierarchical phrase-based MT model of Chiang (2005) and its syntax-based extension introduced in Chapter 3. We add a source span variance model that, for each rule utilized in a probabilistic synchronous context-free grammar (PSCFG) derivation, gives a confidence estimate in the rule based on the number of source words spanned by the rule and its substituted child rules, with the distributions of these source span sizes estimated during training (i.e., rule extraction) time.

We further propose different methods of combining hierarchical and syntax-based PSCFG models, by merging the grammars as well as by interpolating the translation models.

Finally, we compare syntax-augmented MT, which extracts rules based on target-side syntax, to a corresponding variant based on source-side syntax, and experiment with a model extension based on source *and* target syntax.

We evaluate the different models on the NIST large resource Chinese-to-English translation task. Most of this work was published in Zollmann and Vogel (2010).

4.1 Related work

Chiang et al. (2008) introduce *structural distortion features* into a hierarchical phrase-based model, aimed at modeling nonterminal reordering given source span length, by estimating for each possible source span length ℓ a Bernoulli distribution $p(R|\ell)$ where R takes value one if reordering takes place and zero otherwise. Maximum-likelihood estimation of the distribution amounts to simply counting the relative frequency of nonterminal reorderings over all extracted rule instances that incurred a substitution of span length ℓ. In a more fine-grained approach they add a separate binary feature $\langle R, \ell \rangle$ for each combination of reordering truth value R and span length ℓ (where all $\ell \geq 10$ are merged into a single value), and then tune the feature weights discriminatively on a development set. Our approach differs from Chiang et al. (2008) in that we estimate one source span length distribution for each substitution site of each grammar rule, resulting in unique distributions for each rule, estimated from all instances of the rule in the training data. This enables our model to condition reordering range on the individual rules used in a derivation, and even allows to distinguish between two rules r_1 and r_2 that both reorder arguments with identical mean span lengths ℓ, but where the span lengths encountered in extracted instances of r_1 are all close to ℓ, whereas span length instances for r_2 vary widely.

Chen and Eisele (2010) propose a hybrid approach between hierarchical phrase-based MT and a rule based MT system, reporting improvement over each individual model on an English-to-German translation task. Essentially, the rule based system is converted to a single-nonterminal PSCFG, and hence can be combined with the hierarchical model, another single-nonterminal PSCFG, by taking the union of the rule sets and concatenating the feature

vectors. For rules that only exist in one of the two grammars, we assign zero-values for all features corresponding to the missing grammar. We face the challenge of combining the single-nonterminal hierarchical grammar with a multi-nonterminal syntax-augmented grammar. Thus one hierarchical rule typically corresponds to many syntax-augmented rules.

Chiang (2010) augments a hierarchical phrase-based MT model with binary syntax features representing the source and target syntactic constituents of a given rule's instantiations during training, thus taking source and target syntax into account while avoiding the data-sparseness and decoding-complexity problems of multi-nonterminal PSCFG models. In our approach, the source- and target-side syntax directly determines the grammar, resulting in a nonterminal set derived from the labels underlying the source- and target-language treebanks.

4.2 Modeling Source Span Length of PSCFG Rule Substitution Sites

Extracting a rule with k right-hand-side nonterminal pairs, i.e., substitution sites, (from now on called *order-k rule*) involves $k + 1$ phrase pairs: one phrase pair used as initial rule and k phrase pairs that are sub phrase pairs of the first and replaced by nonterminal pairs. Conversely, during translation, applying this rule amounts to combining k hypotheses from k different chart cells, each represented by a source span and a nonterminal, to form a new hypothesis and file it into a chart cell. Intuitively, we want the source span lengths of these $k + 1$ chart cells to be close to the source side lengths of the $k + 1$ phrase pairs from the training corpus that were involved in extracting

the rule. Of course, each rule generally was extracted from multiple training corpus locations, with different involved phrase pairs of different lengths. We therefore model $k + 1$ source span length distributions for each order-k rule in the grammar.

Ignoring the discreteness of source span length for the sake of easier estimation, we assume the distribution to be log-normal. This is motivated by the fact that source span length is positive and that we expect its deviation between instances of the same rule to be greater for long phrase pairs than for short ones.

We can now add $\hat{k} + 1$ features to the translation framework, where $\hat{k}$ is the maximum number of PSCFG rule nonterminal pairs, in our case two. Each feature is computed during translation time. Ideally, it should represent the probability of the hypothesized rule given the respective chart cell span length. However, as each competing rule underlies a different distribution, this would require a Bayesian setting, in which priors over distributions are specified. In this work we take a simpler approach: Based on the rule's span distribution, we compute the probability that a span length no likelier than the one encountered was generated from the distribution. This probability thus yields a confidence estimate for the rule. More formally, let μ be the mean and σ the standard deviation of the logarithm of the span length random variable X concerned, and let x be the span length encountered during decoding. Then the computed confidence estimate is given by

$$P\left(|\ln(X) - \mu| \geq |\ln(x) - \mu|\right) \quad = \quad 2 \ * \ Z\left(-(|\ln(x) - \mu|)/\sigma\right)$$

where Z is the cumulative density function of the normal distribution with mean zero and variance one.

The confidence estimate is one if the encountered span length is equal to

the mean of the distribution, and decreases as the encountered span length deviates further from the mean. The severity of that decline is determined by the distribution variance: the higher the variance, the less a deviation from the mean is penalized. Note that this also ameliorates the problem of sparsity: Rare events result in high variance distributions, leading to low penalties and thus small discriminative power of this feature.

Mean and variance of log source span length are sufficient statistics of the log-normal distribution. As we extract rules in a distributed fashion, we use a straightforward parallelization of the online algorithm of Welford (1962) and its improvement by West (1979) to compute the sample variance over all instances of a rule. In the case of single-occurrence events, the sample variance would be infinite. We avoid this issue by using add-0.01 smoothing.

4.3 Merging a Hierarchical and a Syntax-Based Model

While syntax-based grammars allow for more refined statistical models and guide the search by constraining substitution possibilities in a grammar derivation, grammar sizes tend to be much greater than for hierarchical grammars. Therefore the average occurrence count of a syntax rule is much lower than that of a hierarchical rule, and thus estimated probabilities are less reliable.

We propose to augment the syntax-based "rule given source side" and "rule given target side" distributions by hierarchical counterparts obtained by marginalizing over the left-hand-side and right-hand-side rule nonterminals. For example, the hierarchical equivalent of the "rule given source side" probability is obtained by summing occurrence counts over all rules that have the

same source and target terminals and substitution positions but possibly differ in the left- and/or right-hand side nonterminal labels, divided by the sum of occurrence counts of all rules that have the same source side terminals and source side substitution positions. Similarly, an alternative rareness penalty based on the combined frequency of all rules with the same terminals and substitution positions is obtained.

Using these syntax and hierarchical features side by side amounts to interpolation of the respective probability models in log-space, with minimum-error-rate training (MERT) determining the optimal interpolation coefficient. We also include respective models interpolated with coefficient .5 in probability-space as additional features to the system.

We further experiment with adding hierarchical rules separately to the syntax-augmented grammar, as proposed in Zollmann et al. (2008a), with the respective syntax-specific features set to zero. A 'hierarchical-indicator' feature is added to all rules, which is one for hierarchical rules and zero for syntax rules, allowing the joint model to trade off hierarchical against syntactic rules. During translation, the hierarchical and syntax worlds are bridged by glue rules, which allow monotonic concatenation of hierarchical and syntactic partial sentence hypotheses. We separate the glue feature used in hierarchical and syntax-augmented translation into a glue feature that only fires when a hierarchical rule is glued, and a distinct glue feature firing when gluing a syntax-augmented rule.

4.4 Extension of SAMT to a bilingually parsed corpus

Syntax-based MT models have been proposed both based on target-side syntactic annotations (Galley et al., 2004; Zollmann and Venugopal, 2006) as well source-side annotations (Liu et al., 2006). Syntactic annotations for both source *and* target language are available for popular language pairs such as Chinese-English. In this case, our grammar extraction procedure can be easily extended to impose both source and target constraints on the eligible substitutions simultaneously.

Let N_f be the nonterminal label that would be assigned to a given initial rule when utilizing the source-side parse tree, and N_e the assigned label according to the target-side parse. Then our bilingual model assigns '$N_f + N_e$' to the initial rule. The extraction of complex rules proceeds as before. The number of nonterminals in this model, based on a source-model label set of size s and a target label set of size t, is thus given by st.

4.5 Experiments

We evaluate our approaches by comparing translation quality according to the IBM-BLEU (Papineni et al., 2002) metric on the NIST Chinese-to-English translation task using MT04 as development set to train the model parameters λ, and MT05, MT06 and MT08 as test sets.

We perform PSCFG rule extraction and decoding using our own implementations for the hierarchical and syntax-augmented grammars. For all systems, we use a decoding-time reordering limit of 15 source words, and correspondingly extract rules from initial phrase pairs of maximum source length 15. All rules have at most two nonterminal symbols, which must be non-

consecutive on the source side, and rules must contain at least one source-side terminal symbol.

The parallel training data comprises of 9.6M sentence pairs (206M Chinese Words, 228M English words). The source and target language parses for the syntax-augmented grammar were generated by the Stanford parser (Klein and Manning, 2003). From manual inspection we found the quality of the parses, especially regarding constituents of small ($leq5$) and medium ($leq15$) length spans, which are most crucial for our syntax-augmented grammars, very accurate.

	Dev (MT04)	MT05	MT06	MT08	**TestAvg**	Time
Hierarchical	38.63	36.51	33.26	25.77	**31.85**	14.3
Hier+span	39.03	36.44	33.29	26.26	**32.00**	16.7
Syntax	39.17	37.17	33.87	26.81	**32.62**	59
Syntax+hiermodels	39.61	37.74	34.30	27.30	**33.11**	68.4
Syntax+hiermodels+hierrules	39.69	37.56	34.66	26.93	**33.05**	34.6
Syntax+span+hiermodels+hierrules	39.81	38.02	34.50	27.41	**33.31**	39.6
Syntax/src+span+hiermodels+hierrules	39.62	37.25	33.99	26.44	**32.56**	20.1
Syntax/src&tgt+span+hiermodels+hierrules	39.15	36.92	33.70	26.24	**32.29**	17.5

Table 4.5.1: Translation quality in % case-insensitive IBM-BLEU (i.e., brevity penalty based on closest reference length) for different systems on Chinese-English NIST-large translation tasks. 'TestAvg' shows the average score over the three test sets. 'Time' is the average decoding time per sentence in seconds on one CPU.

The results are given in Table 4.5.1. The source span models (indicated by +span) achieve small test set improvements of 0.15 BLEU points on average for the hierarchical and 0.26 BLEU points for the syntax-augmented system, but these are not statistically significant.

Augmenting a syntax-augmented grammar with hierarchical features ("Syntax+hiermodels") results in average test set improvements of 0.5 BLEU points. These improvements are not statistically significant either, but persist across all three test sets. This demonstrates the benefit of more reliable feature estimation. Further augmenting the hierarchical rules to the grammar ("Syntax+hiermodels+hierrules") does not yield additional improvements.

The use of bilingual syntactic parses ('Syntax/src&tgt') turns out detrimental to translation quality. We assume this is due to the huge number of nonterminals in these grammars and the great amount of badly-estimated low-occurrence-count rules, as well as an increasing number of blocked syntactic derivations due to nonterminal mismatches. Perhaps merging this grammar with a regular syntax-augmented grammar could yield better results. The problem of nonterminal mismatches is also investigated in the thesis work proposed by Hanneman (2011).

We also experimented with a source-parse based model ('Syntax/src'). While not being able to match translation quality of its target-based counterpart, the model still outperforms the hierarchical system on all test sets.

	TestAvg BLEU	LM cost	#Glues	Total #rules	#Target words
Hier+span	**32.00**	52.5	9.03	22.2	27.5
Syntax+span+hiermodels+hierrules	**33.31**	55.4	7.19	21.4	28.5
Syntax/src+span+hiermodels+hierrules	**32.56**	53.2	7.13	22.6	27.6
Syntax/src&tgt+span+hiermodels+hierrules	**32.29**	54.3	7.46	21.2	27.7

Table 4.5.2: Mean (taken over all MT08 test sentences) negative log base 10 language model probabilities, number of glue rule applications, number of total rule applications, and number of produced target-language words for different systems on Chinese-English NIST-large translation tasks.

4.5.1 Analysis of grammar rule, glue rule, and language model reliance

Table 4.5.2 shows the average language model costs, number of glue rule applications, number of total rule applications, and number of produced target-language words for the model-best hierarchical, target-syntax-augmented, source-syntax-augmented and source-and-target-syntax-augmented outputs, computed over all MT08 test sentences.[1] Hierarchical MT, despite having the lowest translation score, has the lowest language model costs (52.5 on average) assigned to its translations. Thus, a sentence from the hierarchical model tends to be considered about 800 times as probable ($10^{55.4-52.5} \approx 794.3$) as a (target-side-)syntax-model sentence by the language model. The other two syntax models are somewhere in between. The many-nonterminal PSCFG underlying the syntax models is thus much stronger than the hierarchical grammar in pushing good derivations despite the "disapproval" of the myopic 5-gram language model (note that all the grammars are weakly equivalent, i.e., can produce the same derivations).

With refined grammars, the need to resort to glue operations is diminished compared to the hierarchical grammar. This can be seen more clearly when considering the total number of rules (glue and regular grammar rules) per derivation: 9.03 for the hierarchical model vs. 7.19 to 7.46 for the syntax-augmented models. The syntax rules are specific enough to allow more accurate application to longer-range sentence structures than the purely hierarchical rules, thus leading to a smaller total number of rule applications and less

[1] We refrain from analysis of the corresponding feature weights, as these are not directly comparable across systems even when normalized by L_1 or L_2 norm, since the relative weight assigned to a given feature can be arbitrarily diminished by increasing the weights of two other negatively correlated features without diminishing the actual contribution of the former feature to the model.

gluing.

The average number of generated target words is highest for the target-syntax system, most likely due to it being the best performing system, decreasing the risk of incurring penalties from generating wrong translations.[2]

4.6 Conclusions and Contributions

We proposed several improvements to hierarchical phrase-based MT and syntax-augmented MT. We added a source span length model that, for each rule utilized in a probabilistic synchronous context-free grammar (PSCFG) derivation, gives a confidence estimate in the rule based on the number of source words spanned by the rule and its substituted child rules, resulting in small improvements for hierarchical phrase-based as well as syntax-augmented MT.

We further demonstrated the utility of combining hierarchical and syntax-based PSCFG models and grammars.

Finally, we compared the original SAMT, which extracts rules based on target-side syntax, to a corresponding variant based on source-side syntax, showing that target syntax is more beneficial, and unsuccessfully experimented with a model extension that jointly takes source and target syntax into account. We believe that using target syntax is superior to using soure syntax because translation is inherently asymmetric task: target side constraints directly enforce grammaticality of the translation output, while source

[2]The worse a statistical machine translation system tuned towards an automatic evaluation metric such as BLEU performs, the briefer its output translations tend to become, as its false guesses become more costly than the penalty incurred for being too brief.

side constraints yield synchronous grammars that discriminate based on input sentence structure, which is a less explicit way of achieving good translations. The challenge of moving from monolingual syntactic structures to bilingual ones is sparsity: the number of nonterminals is now the product of the number of source labels and the number of target labels, and thus the grammar size explodes. We will revisit this problem and propose a solution to it in Chapter 6.

CHAPTER 5

Widening the Pipeline: Grammar Learning from N-best Distributions of Parses and Alignments

So far we have been concerned with learning a grammar from sentence pairs annotated with word alignments and target parses. Even though these alignments and parses are usually not human-generated, but instead come from a separate module in the pipeline that is prone to errors, we have so far assumed them to be the truth. We will now lift that assumption and treat alignments and parses as hidden variables instead. Instead of providing a single integrated model, our aim is to retain the pipeline of self-contained modules responsible for alignment, parsing and extraction. Most of the work in this chapter has been published in (Venugopal, Zollmann, Smith, and Vogel, 2008; Zollmann et al., 2008b). All of the work is an original thesis contribution.

5.1 Motivation

Current phrase-based and hierarchically structured systems rely on the output of a sequential "pipeline" of maximum *a posteriori* inference steps to identify hidden translation structure and estimate the parameters of their translation models. The first step in this pipeline typically involves learning word-alignments (Brown et al., 1993) over parallel sentence aligned training data.

The outputs of this step are the model's most probable word-to-word correspondences within each parallel sentence pair. These alignments are used as the input to a phrase extraction step, where multi-word phrase pairs are identified and scored (with multiple features) based on statistics computed across the training data. The most successful methods extract phrases that adhere to heuristic constraints (Koehn et al., 2003; Och and Ney, 2004). Thus, errors made within the single-best alignment are propagated (1) to the identification of phrases, since errors in the alignment affect which phrases are extracted, and (2) to the estimation of phrase weights, since each extracted phrase is counted as evidence for relative frequency estimates. Methods like those described in Wu (1997), Marcu and Wong (2002), and DeNero, Gillick, Zhang, and Klein (2006) address this problem by jointly modeling alignment and phrase identification, yet have not achieved the same empirical results as surface heuristic based methods, or require substantially more computational effort to train.

In this work we describe an approach that "widens" the pipeline, rather than performing two steps jointly. We present N-best alignments and parses to the downstream phrase extraction algorithm and define a probability distribution over these alternatives to generate expected, possibly fractional counts for the extracted translation rules, under that distribution. These fractional counts are then used when assigning weights to rules.

This technique is directly applicable to both flat and hierarchically-structured translation models. In syntax-based translation, single-best target language parse trees (given by a statistical parser) are used to assign syntactic categories within each rule, and to constrain the combination of those rules. Decisions made during the parsing step of the pipeline affect the choice of

nonterminals used for each rule in the PSCFG. Presenting N-best parse alternatives to the rule extraction process allows the identification of more diverse structures for use during translation and, perhaps, better generalization ability.

The remainder of this chapter is structured as follows: In Section 5.2, we present a method of integrating PSCFG rules extracted from N-best alignments and parses and allow the posterior fractional counts to influence the rule weights. In Section 5.3, we show how the widened pipeline improves translation performance on the limited-domain domain speech translation task introduced in Chapter 3, the IWSLT Chinese-English data track (Paul, 2006). Section 5.4 summarizes our contributions.

5.2 N-best evidence

The SAMT rule extraction procedure (cf. Section 3.1) relies on high quality word alignments and parses. The quality of the alignments affects the set of phrases that can be identified by the heuristics in (Koehn et al., 2003). Improving or diversifying the set of initial phrases also affects the rules with nonterminals that are identified via the procedure described in Chapter 3. Since PSCFG systems rely on rules with nonterminal symbols to represent reordering operations, the set of these initial phrases has the potential to have a profound impact on translation quality. The quality of the parses affects the syntactic categories assigned to the left-hand-side and nonterminal symbols of each rule. These categories play an important role in constraining the decoding process to grammatically feasible target parse trees.

Several recent studies explore the relationship between the quality of

the initial models in the "pipeline" and final translation quality. Quirk and Corston-Oliver (2006) show improvements in translation quality when the quality of parsing is improved by adding additional training data within the "treelet" paradigm introduced by Quirk, Menezes, and Cherry (2005). Koehn et al. (2003) show that translation quality in a phrase-based system does not vary significantly when increasing the complexity of the model used for alignment (ranging from IBM model 1 through 4), but that increasing the amount of parallel training data does improve alignment quality. Ganchev, Graca, and Taskar (2008) demonstrate significant improvements in both alignment quality (as measured by alignment error rate (Och and Ney, 2003)) and translation quality when using a posterior decoding method to select alignments (as opposed to the single-best Viterbi alignment). Xue, Li, Zhao, Yang, and Li (2006) apply n-best alignments to improve phrase-based translation, while Dyer, Muresan, and Resnik (2008b) and Mi et al. (2008) widen the pipeline by considering word-lattice and forest-based translation, respectively, rather than translating the single-best hypothesis from a previous stage in the pipeline.

Our approach considers alignment and parse quality for a fixed training data size and model complexity. The alignment model and the parser are capable of generating N-best alternative candidates along with corresponding probabilities for each candidate. Informal examination of the highest probability alignment and target parse tree reveals two important arguments in favor of integrating N-best hypotheses into the rule extraction process. Firstly, there are often multiple reasonable alignments and parses that can model the bilingual sentence pair and the target sentence. We can expect that rules extracted from more diverse, correct evidence can improve translation quality on new sentences, since more (good) rules will be extracted. Secondly, where there is a high degree of agreement across each alternative in the N-best lists,

the remaining differences between alternatives are often the source of error or ambiguity.

Attempts to reduce the use (in decoding) of rules extracted from sections of the alignment and parse that are not consistent with other alternatives could reduce errors made during translation. Put another way, the more complete hypotheses a word-link or constituent appears in, and the more probable those hypotheses, the more we should trust rules that use these links.

Our approach toward the integration of N-best evidence into the grammar construction process allows us to take advantage of the diversity found in the N best alternatives, while reducing the negative impact of errors made in these alternatives.

5.2.1 Counting from N-best lists

In this work we propose extraction of complex rules over N-best alignments and N'-best parses, making use of probability distributions over these alternatives to assign fractional posterior counts to each extracted rule.

Taking the alignment N-best list to define a posterior distribution over alignments and the parse N'-best list to define a posterior over parse trees, we can estimate the posterior probability of each rule that might be extracted for each (alignment, tree) pair. Assuming that the alignment module gives alignments $a_1, ..., a_N$, with posterior probabilities $p(a_1 \mid e, f), ..., p(a_N \mid e, f)$, we approximate the posterior by renormalizing:

$$\hat{p}(a_i \mid e, f) = p(a_i \mid e, f) \Big/ \sum_{j=1}^{N} p(a_j \mid e, f) \tag{5.2.1}$$

The same is applied to the parser's N'-best parses, $\pi_1, ..., \pi_{N'}$.

Given a single alignment-parse pair, we can extract rules as described

in Section 3.1. Our approach is to extract rules from the cross-product $\{a_1, ..., a_N\} \times \{\pi_1, ..., \pi_{N'}\}$, incrementing the partial count of each rule extracted by $\hat{p}(a_i) \cdot \hat{p}(\pi_j)$. A rule instance r's total count for the sentence pair $\langle f, e \rangle$ is:

$$\sum_{i=1}^{N} \sum_{j=1}^{N'} \hat{p}(a_i \mid e, f) \cdot \hat{p}(\pi_j \mid e) \cdot \begin{cases} 1 & \text{if } r \text{ can be extracted from} \\ & e, f, a_i, \pi_j \\ 0 & \text{otherwise} \end{cases} \tag{5.2.2}$$

If r is extracted at multiple places in the sentence pair, all these instances' counts are added up. In practice, Formula 5.2.2 can be computed more efficiently through structure-sharing. Note that if $N = N' = 1$, this counting method is equivalent to the original counting method.

Note that GIZA++ (Och and Ney, 2003) can infer the N-best word alignments under IBM Model 4 and the Charniak parser (Charniak, 2000) outputs its N'-best parses, with their associated probabilities.

Instead of using the simple counts for rules given the derivation inferred using the maximum *a posteriori* estimated alignment and parse (a_1, π_1), we now use the expected counts under the approximate posterior. These posteriors encode (in a principled way) a measurement of confidence in substructures used to generate each rule. Possible rule instances supported by more and more likely alignments and parses should, intuitively, receive higher counts (approaching 1 as certainty increases, supported by more and higher-probability alternatives), while rule instances that rely on low probability or fewer alignments and parses will get lower counts (approaching 0 as certainty increases). We will give examples of such rules from real data in Section 5.3.3.

5.2.2 Refined alignments

Work by Och and Ney (2004) and Koehn et al. (2003) demonstrates the value of generating word alignments in both source-to-target and target-to-source directions in order to facilitate the extraction of phrases with many-to-many word relationships. We follow Koehn et al. (2003) in generating a refined bidirectional alignment using the heuristic algorithm "grow-diag-final-and" described in that work. Since we require N-best alignments, we first extract N-best alignments in each direction, and then perform the refinement technique to all N^2 bidirectional alignment pairs. The resulting alignments are assigned the probability $(p_f.p_r)^\alpha$ where p_f is the candidate probability for the forward alignment and p_r is the candidate probability to the reverse alignment.

We then remove any duplicate refined alignments (the refined alignment with the highest probability is retained) that came about due to the refinement process. Finally, we select the top N alignments from this set of refined alignments.

The parameter α controls the entropy of the resulting (normalized) distribution over candidate alignments (note, however, that the order of the ranked sequence of alignments is not affected). Higher values of α make the distribution more peaked (affecting the estimation of features on rules from these alignments), while smaller values make the distribution more uniform. A more peaked distribution favors rules from the top alignments, while a more uniform one gives rules from lower performing alignments more of a chance to participate in translation. As a special case, $\alpha = 1/2$ effectively uses the geometric mean of forward and reverse alignment weights. We can also use this same technique to control the distribution over parses.

5.3 Translation results

5.3.1 Experimental setup

We present results on the IWSLT 2007 and 2008 Chinese-to-English translation task, based on the full BTEC corpus of travel expressions with 120K parallel sentences (906K source words and 1.2M target words) as well as the evaluation corpora from the evaluation years preceding 2007. The development data consists of 489 sentences (average length of 10.6 words) from the 2006 evaluation, the 2007 test set contains 489 sentence (average length of 6.47 words) sentences and the 2008 test set contains 507 sentences (average length of 5.59 words). Word alignment was trained using the GIZA++ toolkit, and N-best parses generated by the Charniak (2000) parser, without additional re-ranking.[1] N-best alignments were generated from source to target and target to source, refined as described above.

Initial phrases of up to length 10 were identified using the heuristics proposed by Koehn et al. (2003). Rules with up to 2 nonterminals are extracted using our SAMT toolkit (cf. Chapter 3), modified to handle N-best alignments and parses and posterior counting. Note that lexical weights (Koehn et al., 2003) as described above are assigned to ϕ based on "single-best" word alignments. Rules that receive zero probability value for their lexical weights are immediately discarded, since they would then have a prohibitively high cost when used during translation. Rules extracted from single-best evidence as well as N best evidence can be discarded in this way.

The n-gram language model is trained on the target side of the parallel

[1] Reranking might be used to change estimates of $\hat{p}(\tau_i)$, but would not change the set of rules extracted—only the fractional counts.

training corpus[2] and translation experiments use the decoder and MER trainer available in the SAMT toolkit. We use the cube-pruning option (Chiang, 2007) in these experiments.

5.3.2 Cumulative (N, N')-best

We measure translation quality using the mixed-cased IBM-BLEU (Papineni et al., 2002) metric as we vary the size of N and N' for alignments and parses respectively. Each value of N implies that the first N alternatives have been considered when building the grammar. For each grammar we also track the number of rules relevant for the first sentence in the IWSLT 2007 test set (grammars are subsampled on a per-sentence basis to keep memory requirements low during decoding). We also note the number of seconds required to translate each test set. Due to time and resource constraints we limit our evaluation to varying the number of alignments and parses separately, and we limit N' to 10 (due to the significant increase in decoding time that results from adding more nonterminal labels to the grammar).

As noted above, many rules extracted based on N-best alignments cannot participate in the decoding process because lexical weight features can have costs of infinity if the underlying word based models $\hat{p}(s|t)$ and $\hat{p}(t|s)$, estimated based on "single-best" alignments, yield zero probabilities. Smoothing these models alleviates the problem, but does not fix it at its root. In the spirit of softening our pipelined decisions, we create lexical weight features based on the IBM Model 4 tables output by GIZA++ at the end of its training, instead of single-best alignment relative frequencies. Using these IBM Model

[2] As BTEC is a very domain-specific corpus, training the language model on large available monolingual corpora (e.g., from the news-domain) is of limited utility.

4 weights allows a larger number of rules to be added to the grammar since more rules have non-zero lexical weights.

We also investigate the impact of the shape N-best probability distribution used to estimate features ϕ by varying α.

N-best alignments. Table 5.3.1 shows translation results on the IWSLT translation task for the Development (IWSLT 2006) and two test corpora (IWSLT 2007 and 2008) using the Syntax Augmented grammar. In this table we vary the number of alternative alignments, consider first-best (1), 5, 10 and 50 best alternatives. We also experiment with lexical weights from the first-best alignment ($lex = 1st$) and directly from IBM Model 4 ($lex = m4$), while α controls the entropy of the normalized distribution over alternative alignments.

For the Syntax-Augmented grammar, using $lex = m4$ slightly increases the number of rules in the grammar, but only adds benefit for the 2007 test set. We continue to use $lex = m4$ for the remaining experiments since we do not want to discard rules based on the lexical weights. Increasing $N = 1$ to $N = 5$ brings significant improvements in translation quality on all 3 evaluation corpora, while increasing N further to $N = 10$ and $N = 50$ retain the improvements but at the cost of a significantly larger grammar and decoding times. Varying α to modify the entropy of the alignment distribution does not seem to have a consistent impact on translation quality; some test sets show improvements while others suffer.

N-best alignments (hierarchical grammar). Similar results with the purely hierarchical grammar are shown in Table 5.3.2. We see clear improve-

System	# Rules (1 sent.)	Dev	2007	2008	2007 Time (s)	2008 Time (s)
$N = 1$ (lex=1st)	400K	0.309	0.355	0.453	8108	8367
$N = 1$ ($\alpha = 1$ lex=m4)	420K	0.301	0.361	0.440	8024	8250
$N = 5$ ($\alpha = 1$ lex=m4)	680K	0.322	0.374	0.470	15376	15577
$N = 10$ ($\alpha = 1$ lex=m4)	900K	0.313	0.382	0.467	19298	19469
$N = 50$ ($\alpha = 1$ lex=m4)	1500K	0.316	0.370	0.478	29500	30894
$N = 10$ ($\alpha = 0.5$ lex=m4)	900K	0.315	0.395	0.477	20398	20118
$N = 50$ ($\alpha = 0.5$ lex=m4)	1500K	0.317	0.373	0.477	33682	34760
$N = 10$ ($\alpha = 2$ lex=m4)	900K	0.313	0.375	0.464	15117	15070
$N = 50$ ($\alpha = 2$ lex=m4)	1500K	0.315	0.373	0.488	26590	27126

Table 5.3.1: Grammar statistics and translation quality (IBM-BLEU) on development (IWSLT 2006) and test set (IWSLT 2007, 2008) when integrating N-best alignments for alternative Syntax Augmented grammar configurations. # Rules reflect rules that are applicable to the first sentence in IWSLT 2007. Decoding times in seconds are cumulative over all sentences in respective test set.

ments when moving to $N = 5$, and even further small improvement up to $N = 10$, but a slight degradation going further to $N = 50$. Again, we do not see a clear benefit from varying α. Surprisingly, while Dev. scores are significantly lower with the purely hierarchical grammar compared to the Syntax Augmented grammar, unseen test set scores are very similar, and achieved at significantly lower decoding times. Since the number of features in ϕ are very similar for both models, it is unlikely that this discrepancy is solely due to overfitting during MER training. It is more likely that this discrepancy is related to the relative lengths of each evaluation corpus. The development corpus contains longer sentences on average than the evaluation corpora. The number of rules used in purely hierarchical grammar is significantly lower than in the Syntax Augmented grammar, and increasing N does not exhibit the same growth in the number of rules either. The Syntax Augmented grammar grows much faster since rule identified from alternative alignment candidates have syntactic nonterminal symbols and are less likely to be duplicates of already identified rules.

N'-best parses. Table 5.3.3 summarizes results when varying the number of alternative parses. These experiments use $\alpha = 1$, $lex = m4$ and 1-best alignments only. We also additionally track the number of nonterminal labels represented in the grammar. Using additional evidence from N'-best parses seems to have an overall slightly negative impact on translation quality while taking significantly longer to perform decoding. It is possible that $N' = 10$ is still too small to provide enough variation in the N'-best list. However, as can bee seen in the table, the growth in the number of nonterminal labels when going from $N' = 1$ to $N' = 10$ already leads to nearly three times as many rules. Furthermore, this increased number of rules has a dramatic impact

System	# Rules (1 sent.)	Dev	2007	2008	2007 Time (s)	2008 Time (s)
Hier $N = 1$	10K	0.277	0.367	0.460	895	1451
Hier $N = 5$ ($\alpha = 1$)	12K	0.286	0.374	0.472	906	1476
Hier $N = 10$ ($\alpha = 1$)	13K	0.291	0.382	0.477	944	1516
Hier $N = 50$ ($\alpha = 1$)	14K	0.282	0.384	0.463	979	1596
Hier $N = 10$ ($\alpha = 0.5$)	13K	0.285	0.399	0.476	963	1547
Hier $N = 50$ ($\alpha = 0.5$)	14K	0.283	0.376	0.470	982	1599
Hier $N = 10$ ($\alpha = 2$)	13K	0.284	0.372	0.467	965	1570
Hier $N = 50$ ($\alpha = 2$)	14K	0.290	0.374	0.459	921	1483

Table 5.3.2: Grammar statistics and translation quality (IBM-BLEU) on development (IWSLT 2006) and test sets (IWSLT 2007, 2008) when integrating N-best alignments for purely hierarchical grammar configurations. # Rules reflect rules that are applicable to the first sentence in IWSLT 2007. Decoding times in seconds are cumulative over all sentences in respective test set.

System	# Rules (1 sent.)	# Labels	Dev	2007	2008	2007 Time (s)	2008 Time (s)
$N' = 1$	420K	10K	0.301	0.361	0.440	8024	8250
$N' = 5$	800K	15K	0.300	0.358	0.447	16930	15102
$N' = 10$	1079K	18K	0.299	0.361	0.460	26944	23662

Table 5.3.3: Grammar statistics and translation quality (IBM-BLEU) on development (IWSLT 2006) and test sets (IWSLT 2007, 2008) and when integrating N-best parses with the Syntax Augmented grammar. # Rules reflect rules that are applicable to the first sentence in IWSLT 2007. Decoding times in seconds are cumulative over all sentences in respective test set. All experiments in this table use $lex = m4$, $\alpha = 1$ and 1-best alignments.

on decoding time and likely contributes to additional search errors. The one corpus where alternative parses ($N' = 10$) produces results comparable to using N best alignments is IWSLT 2008, which is also the corpus with the shortest sentences on average, thus reducing the potential impact of search error.

5.3.3 Grammar rules

Figure 5.3.1 shows the most frequently occurring rules that exist only in the best performing $N = 10, N' = 1$ grammar, and not in the baseline (Model-4 lexicon) grammar. We show the estimated counts on these rules as well as their source, target and left-hand-side nonterminal symbol. These rules are particularly interesting when considering the domain of this translation task. The source side of the training data contains no punctuation (since it is transcribed speech), while the target side does (since they were manually generated translations). The system therefore attempts to generate punctuation during translation. Consider the first example, where the Chinese word

count	source	target	LHS NT
247.93	请	please .	@UH+.
210.69	请	please .	@VB+.
162.06	想	'd	@MD
153.42	我	, I	@,+PRP
146.32	我	I have	@PRP+AUX
141.96	我	.	@.
141.75	的	in	@IN
133.52	我 想	I 'd	@PRP+MD
130.99	~	did you	@AUX+PRP
125.18	的	is	@AUX

Figure 5.3.1: Top rules extracted by our method, but not the baseline.

for "please" (often found at the beginning of a sentence) is aligned to the English "please ." (at the end of the sentence as indicated by the punctuation). This rule is extracted from a lower-probability alignment with high levels of distortion. This pattern was not seen in any single-best alignments.

5.4 Conclusion and contributions

In this chapter we have demonstrated the feasibility and benefits of widening the MT pipeline to include additional evidence from N-best alignments and parses. We integrate this diverse knowledge under a principled model that uses a probability distribution over these alternatives. We achieve significant improvements in translation quality over grammars built on "single-best" evidence alone when considering N-best alignments, while N'-best parses seem to have no impact on translation quality. Using a relatively small number of additional alternative alignments results in significant improvements in quality, with minimal impact on the number of rules in the grammar and the translation runtime for a hierarchical system, but at significantly increased

grammar size and runtime for a syntax-augmented system. We made our 'wider-pipeline' model freely available to the MT community by integrating it into the PSCFG grammar construction process of our SAMT system.

CHAPTER 6

Word-Class Based Rule Labeling

As we have seen in Chapters 2 and 3, the PSCFG formalism suggests an intuitive approach to model the long-distance and lexically sensitive reordering phenomena that often occur across language pairs considered for statistical machine translation. As in monolingual parsing, nonterminal symbols in translation *rules* are used to generalize beyond purely lexical operations. In the SAMT model, *labels* on these nonterminal symbols are used to enforce syntactic constraints in the generation of bilingual sentences and imply conditional independence assumptions in the statistical translation model. While this method results in improvements in translation quality over the single X label approach (cf. Section 3.5), high quality syntactic trees are not readily available for all languages. Parse trees used for SAMT training rely on stochastic parsers that have been trained on manually created syntactic treebanks. These treebanks are difficult and expensive to create and exist for a limited set of languages only, while simpler part-of-speech (POS) taggers are available on a wide range of languages such as, e.g., Slovene, Galician, Greek, Russian and Finnish. Furthermore, when the genre and domain of the parallel data differs from that of the treebank, parses deteriorate, a problem that is less severe for part-of-speech taggers.

In this chapter, we propose a labeling approach that is based merely on part-of-speech analysis source or target language (or even both). When using

English POS tags in our labeling approach we achieve improvements in translation quality over the single label approach from Chiang (2005), and come close to the SAMT model.

Towards the ultimate goal of building end-to-end machine translation systems without *any* human annotations, we also experiment with automatically inferred word classes using distributional clustering (Kneser and Ney, 1993). Since the number of classes is a parameter of our clustering method and the resulting nonterminal size of our grammar is a function of the number of word classes, the PSCFG grammar complexity can be adjusted to the specific translation task at hand. Varying the number of classes in our clustering model allows us to find conditions where clustered tags almost match the performance of the POS-based approach.

Finally, we introduce a more flexible labeling approach based on K-means clustering, which allows the incorporation of an arbitrary number of word-class based features, including phrasal contexts, can make use of multiple tagging schemes, and also allows non-class features such as phrase sizes.

The work in this chapter has been published in Zollmann and Vogel (2011).

6.1 Hard rule labeling from word classes

We now describe the extraction of PSCFG rules based on a parallel corpus with word-tagged target side sentences. The same procedure can straightforwardly be applied to a corpus with tagged source side sentences. We use the simple term 'tag' to stand for any kind of word-level analysis—a syntactic, statistical, or other means of grouping words into classes, possibly based on

their position and context in the sentence, part-of-speech tagging being the most obvious example.

As in Chiang's hierarchical model and the SAMT model, we rely on an external phrase-extraction procedure, such as the one of Koehn et al. (2003). Let $f = f_1 \cdots f_m$ be the current source sentence, $e = e_1 \cdots e_n$ the current target sentence, and $t = t_1 \cdots t_n$ its corresponding target tag sequence. We convert each extracted phrase pair p, represented by its source span start and end points $\langle \mathrm{srcbeg}(p), \mathrm{srcend}(p) \rangle$ and target span start and end points $\langle \mathrm{trgbeg}(p), \mathrm{trgend}(p) \rangle$, into an initial rule

$$t_{\mathrm{trgbeg}(p)}\text{-}t_{\mathrm{trgend}(p)} \rightarrow f_{\mathrm{srcbeg}(p)} \cdots f_{\mathrm{srcend}(p)} \mid e_{\mathrm{trgbeg}(p)} \cdots e_{\mathrm{trgend}(p)}$$

by assigning it a nonterminal "$t_{\mathrm{trgbeg}(p)}$-$t_{\mathrm{trgend}(p)}$" constructed by combining the tag of the target phrase's left-most word with the tag of its right-most word.

The creation of complex rules based on all initial rules obtained from the current sentence now proceeds just as in Chiang's model.

Consider the target-tagged example sentence pair

"Ich habe ihn gesehen | I/PRP saw/VBD him/PRP"

with extracted span-annotated phrase pairs:

1: Ich (1-1) | I (1-1)

2: ihn (3-3) | him (3-3)

3: gesehen (4-4) | saw (2-2)

4: habe ihn gesehen (1-3) | saw him (2-3)

5: Ich habe ihn gesehen (1-4) | I saw him (1-3)

Then our method extracts the rules shown in Figure 6.1.1, where the subscript indices of the right-hand-side nonterminals indicate the one-to-one correspondence between the source and target substitution sites. For example, the rule

$$\text{“VBD-PRP} \rightarrow \text{habe PRP-PRP}_1 \text{ gesehen } \mid \text{saw PRP-PRP}_1\text{”}$$

would be extracted by abstracting-out phrase pair 2 from phrase pair 4.

Intuitively, the labeling of initial rules with tags marking the boundary of their target sides results in complex rules whose nonterminal occurrences impose weak syntactic constraints on the rules eligible for substitution in a PSCFG derivation: The left and right boundary word tags of the inserted rule's target side have to match the respective boundary word tags of the phrase pair that was replaced by a nonterminal when the complex rule was created from a training sentence pair. Since consecutive words within a rule stem from consecutive words in the training corpus and thus are already consistent, the boundary word tags are more informative than tags of words between the boundaries for the task of combining different rules in a derivation, and are therefore a more appropriate choice for the creation of grammar labels than tags of inside words.

Initial rules

PRP-PRP $\rightarrow$ Ich , I

PRP-PRP $\rightarrow$ ihn , him

VBD-VBD $\rightarrow$ gesehen , saw

VBD-PRP $\rightarrow$ habe ihn gesehen , saw him

PRP-PRP $\rightarrow$ Ich habe ihn gesehen , I saw him

Complex rules

VBD-PRP $\rightarrow$ habe PRP-PRP$_1$ gesehen , saw PRP-PRP$_1$

VBD-PRP $\rightarrow$ habe ihn VBD-VBD$_1$, VBD-VBD$_1$ him

VBD-PRP $\rightarrow$ habe PRP-PRP$_1$ VBD-VBD$_2$, VBD-VBD$_2$ PRP-PRP$_1$

PRP-PRP $\rightarrow$ PRP-PRP$_1$ habe ihn gesehen , PRP-PRP$_1$ saw him

PRP-PRP $\rightarrow$ Ich VBD-PRP$_1$, I VBD-PRP$_1$

PRP-PRP $\rightarrow$ PRP-PRP$_1$ VBD-PRP$_2$, PRP-PRP$_1$ VBD-PRP$_2$

PRP-PRP $\rightarrow$ Ich habe ihn VBD-VBD$_1$, I VBD-VBD$_1$ him

PRP-PRP $\rightarrow$ Ich habe PRP-PRP$_1$ VBD-VBD$_2$, I VBD-VBD$_2$ PRP-PRP$_1$

PRP-PRP $\rightarrow$ PRP-PRP$_1$ habe ihn VBD-VBD$_2$, PRP-PRP$_1$ VBD-VBD$_2$ him

PRP-PRP $\rightarrow$ PRP-PRP$_1$ habe PRP-PRP$_2$ VBD-VBD$_3$, PRP-PRP$_1$ VBD-VBD$_3$ PRP-PRP$_2$

Figure 6.1.1: Rules extracted from our example by the basic target-tag based model.

6.1.1 Glue rules

As in the SAMT model, we use glue rules of the form

$$\text{S} \rightarrow < s > \ | \ < s >$$

$$\text{S} \rightarrow \text{S}_1 \ N_2 \ | \ \text{S}_1 \ N_2 \quad \text{(for all nonterminals } N\text{)}$$

$$\text{S} \rightarrow \text{S}_1 \ < /s > \ | \ \text{S}_1 \ < /s >$$

(where $< s >$ is the beginning-of-sentence and $< /s >$ the end-of-sentence marker).

6.1.2 Accounting for phrase size

In syntax-based PSCFGs, initial rules corresponding to large phrase pairs tend to be assigned different nonterminal symbols than ones corresponding to small phrase pairs, as they represent constituents at different depths in the corresponding parse tree. A potential pitfall of the tag-based model suggested above is that a single-word initial rule such as

$$\text{PRP-PRP} \rightarrow \text{Ich} \ | \ \text{I}$$

can have the same left-hand-side nonterminal as a long rule with identical left and right boundary tags, such as (when using target-side tags):

$$\text{PRP-PRP} \rightarrow \text{Ich habe ihn gesehen} \ | \ \text{I saw him}$$

In an extension of our model, we therefore introduce a means of distinguishing between one-word, two-word, and multiple-word phrases as follows: Each one-word phrase with tag T simply receives the label T, instead of T-T. Two-word phrases with tag sequence T_1T_2 are labeled T_1-T_2 as before.

Phrases of length greater two with tag sequence $T_1 \cdots T_n$ are labeled $T_1..T_n$ to denote that tags were omitted from the phrase's tag sequence. The resulting number of grammar nonterminals based on a tag vocabulary of size t is thus given by $2t^2 + t$. We do not mark phrase sizes greater than two explicitly by length as this would create many sparse, low-frequency rules, and one of the strengths of PSCFG-based translation is the ability to substitute flexible-length spans into nonterminals of a derivation.

As explained in more detail in Section 4.1, an alternative way of accounting for phrase size is presented by Chiang et al. (2008), who introduce binary *structural distortion features* into a hierarchical phrase-based model. Our approach instead uses distinct grammar rules and labels to discriminate phrase size, with the advantage of enabling all translation models to estimate distinct weights for distinct size classes and avoiding the need of additional models in the log-linear framework; however, the increase in the number of labels and thus grammar rules decreases the reliability of estimated models for rare events due to increased data sparseness, and results in more blocked rules due to increased label mismatch.

6.1.3 Extension to a bilingually tagged corpus

While the availability of syntactic annotations for both source *and* target language is unlikely in most translation scenarios, some form of word tags, be it part-of-speech tags or learned word clusters (cf. Section 6.1.4) might be available on both sides. In this case, our grammar extraction procedure can be easily extended to impose both source and target constraints on the eligible substitutions simultaneously.

Let N_f be the nonterminal label that would be assigned to a given initial

rule when utilizing the source-side tag sequence, and N_e the assigned label according to the target-side tag sequence. Then our bilingual tag-based model assigns '$N_f + N_e$' to the initial rule. The extraction of complex rules proceeds as before. The number of nonterminals in this model, based on a source tag vocabulary of size s and a target tag vocabulary of size t, is thus given by s^2t^2 for the regular labeling method and $(2s^2 + s)(2t^2 + t)$ when accounting for phrase size.

Consider again our example sentence pair (now also annotated with source-side part-of-speech tags):

Ich/PRP habe/AUX ihn/PRP gesehen/VBN
I/PRP saw/VBD him/PRP

Given the same phrase extraction method as before, the resulting initial rules for our bilingual model, when also accounting for phrase size, are as follows:

1: PRP+PRP $\rightarrow$ Ich | I
2: PRP+PRP $\rightarrow$ ihn | him
3: VBN+VBD $\rightarrow$ gesehen | saw
4: AUX..VBN+VBD-PRP $\rightarrow$ habe ihn gesehen | saw him
5: PRP..VBN+PRP..PRP $\rightarrow$ Ich habe ihn gesehen | I saw him

Abstracting-out rule 2 from rule 4, for instance, leads to the complex rule:

AUX..VBN+VBD-PRP $\rightarrow$ habe PRP+PRP$_1$ gesehen | saw PRP+PRP$_1$

A full list of extracted rules is shown in Figure 6.1.2.

Initial rules

PRP+PRP → Ich , I

PRP+PRP → ihn , him

VBN+VBD → gesehen , saw

AUX..VBN+VBD-PRP → habe ihn gesehen , saw him

PRP..VBN+PRP..PRP → Ich habe ihn gesehen , I saw him

Complex rules

AUX..VBN+VBD-PRP → habe PRP+PRP$_1$ gesehen , saw PRP+PRP$_1$

AUX..VBN+VBD-PRP → habe ihn VBN+VBD$_1$, VBN+VBD$_1$ him

AUX..VBN+VBD-PRP → habe PRP+PRP$_1$ VBN+VBD$_2$, VBN+VBD$_2$ PRP+PRP$_1$

PRP..VBN+PRP..PRP → PRP+PRP$_1$ habe ihn gesehen , PRP+PRP$_1$ saw him

PRP..VBN+PRP..PRP → Ich AUX..VBN+VBD-PRP$_1$, I AUX..VBN+VBD-PRP$_1$

PRP..VBN+PRP..PRP → PRP+PRP$_1$ AUX..VBN+VBD-PRP$_2$, PRP+PRP$_1$ AUX..VBN+VBD-PRP$_2$

PRP..VBN+PRP..PRP → Ich habe ihn VBN+VBD$_1$, I VBN+VBD$_1$ him

PRP..VBN+PRP..PRP → Ich habe PRP+PRP$_1$ VBN+VBD$_2$, I VBN+VBD$_2$ PRP+PRP$_1$

PRP..VBN+PRP..PRP → PRP+PRP$_1$ habe ihn VBN+VBD$_2$, PRP+PRP$_1$ VBN+VBD$_2$ him

PRP..VBN+PRP..PRP → PRP+PRP$_1$ habe PRP+PRP$_2$ VBN+VBD$_3$, PRP+PRP$_1$ VBN+VBD$_3$ PRP+PRP$_2$

Figure 6.1.2: Rules extracted from our example by the bilingual tag based model with accounting for phrase size.

6.1.4 Unsupervised word class assignment by clustering

The unsupervised clustering methods we employed to obtain the class assignments are based on the exchange algorithm (Kneser and Ney, 1993). Its objective function is maximizing the likelihood

$$\prod_{i=1}^{n} P(w_i|w_1, \ldots, w_{i-1})$$

of the training data $w = w_1, \ldots, w_n$ given a partially class-based bigram model of the form

$$P(w_i|w_1, \ldots, w_{i-1}) \approx p(c(w_i)|w_{i-1}) \cdot p(w_i|c(w_i))$$

where $c : \mathcal{V} \rightarrow \{1, \ldots, N\}$ maps a word w to its class $c(w)$, $\mathcal{V}$ is the vocabulary, and N the fixed number of classes, which has to be chosen *a priori*. We use the publicly available implementation MKCLS (Och, 1999) to train this model. As training data we use the respective side of the parallel training data for the translation system.

We also experiment with the extension of this model by Clark (2003), who incorporated morphological information by imposing a Bayesian prior on the class mapping c, based on N individual distributions over strings, one for each word class. Each such distribution is a character-based hidden Markov model, thus encouraging the grouping of morphologically similar words into the same class.

6.2 Clustering phrase pairs directly using the K-means algorithm

Even though we have only made use of the first and last words' classes in the labeling methods described so far, the number of resulting grammar

nonterminals quickly explodes. Using a scheme based on source and target phrases with accounting for phrase size, with 36 word classes (the size of the Penn English POS tag set) for both languages, yields a grammar with $(36 + 2 * 36^2)^2 = 6.9$ million nonterminal labels.

Quite plausibly, phrase labeling should be informed by more than just the classes of the first and last words of the phrase. Taking phrase context into account, for example, can aid the learning of syntactic properties: a phrase beginning with a determiner and ending with a noun, with a verb as right context, is more likely to be a noun phrase than the same phrase with another noun as right context. In the current scheme, there is no way of distinguishing between these two cases. Similarly, it is conceivable that using non-boundary words inside the phrase might aid the labeling process.

When relying on unsupervised learning of the word classes, we are forced to chose a fixed number of classes. A smaller number of word clusters will result in smaller number of grammar nonterminals, and thus more reliable feature estimation and more opportunities for rules to combine during decoding, while a larger number has the potential to discover more subtle syntactic properties. Using multiple word clusterings simultaneously, each based on a different number of classes, could turn this global, hard trade-off into a local, soft one, informed by the number of phrase pair instances available for a given granularity.

Lastly, our method of accounting for phrase size is somewhat displeasing: While there is a hard partitioning of one-word and two-word phrases, no distinction is made between phrases of length greater than two. Marking phrase sizes greater than two explicitly by length, however, would create many sparse, low-frequency rules, and one of the strengths of PSCFG-based

translation is the ability to substitute flexible-length spans into nonterminals of a derivation. A partitioning where phrase size is instead merely a feature informing the labeling process seems more desirable.

We thus propose to represent each phrase pair instance (including its bilingual one-word contexts) as feature vectors, i.e., points of a vector space. We then use these data points to partition the space into clusters, and subsequently assign each phrase pair instance the cluster of its corresponding feature vector as label.

The feature mapping Consider the phrase pair instance

$$(f_0)f_1 \cdots f_m(f_{m+1}) \mid (e_0)e_1 \cdots e_n(e_{n+1})$$

(where $f_0, f_{m+1}, e_0, e_{n+1}$ are the left and right, source and target side contexts, respectively). We begin with the case of only a single, target-side word class scheme (either a tagger or an unsupervised word clustering/POS induction method). Let $C = \{c_1, \ldots, c_N\}$ be its set of word classes. Further, let c_0 be a short-hand for the result of looking up the class of a word that is out of bounds (e.g., the left context of the first word of a sentence, or the second word of a one-word phrase). We now map our phrase pair instance to the real-valued vector (where $\mathbb{1}_{[P]}$ is the indicator function defined as 1 if property P is true,

and 0 otherwise):

$$\Big\langle \mathbb{1}_{[e_1=c_0]}, \ldots, \mathbb{1}_{[e_1=c_N]}, \mathbb{1}_{[e_n=c_0]}, \ldots, \mathbb{1}_{[e_n=c_N]},$$
$$\alpha_{\text{sec}}\mathbb{1}_{[e_2=c_0]}, \ldots, \alpha_{\text{sec}}\mathbb{1}_{[e_2=c_N]},$$
$$\alpha_{\text{sec}}\mathbb{1}_{[e_{n-1}=c_0]}, \ldots, \alpha_{\text{sec}}\mathbb{1}_{[e_{n-1}=c_N]},$$
$$\frac{\alpha_{\text{ins}}\sum_{i=1}^{n}\mathbb{1}_{[e_i=c_0]}}{n}, \ldots, \frac{\alpha_{\text{ins}}\sum_{i=1}^{n}\mathbb{1}_{[e_i=c_N]}}{n},$$
$$\alpha_{\text{cntxt}}\mathbb{1}_{[e_0=c_0]}, \ldots, \alpha_{\text{cntxt}}\mathbb{1}_{[e_0=c_N]},$$
$$\alpha_{\text{cntxt}}\mathbb{1}_{[e_{n+1}=c_0]}, \ldots, \alpha_{\text{cntxt}}\mathbb{1}_{[e_{n+1}=c_N]},$$
$$\alpha_{\text{phrsize}}\sqrt{N+1}\log_{10}(n)\Big\rangle$$

The α parameters determine the influence of the different types of information. The elements in the first line represent the phrase boundary word classes, the next two lines the classes of the second and penultimate word, followed by a line representing the accumulated contents of the whole phrase, followed by two lines pertaining to the context word classes. The final element of the vector is proportional to the logarithm of the phrase length.[1] We chose the logarithm assuming that length deviation of syntactic phrasal units is not constant, but proportional to the average length. Thus, all other features being equal, the distance between a two-word and a four-word phrase is the same as the distance between a four-word and an eight-word phrase.

We will mainly use the Euclidean (L_2) distance to compare points for clustering purposes. Our feature space is thus the Euclidean vector space $\mathbb{R}^{7N+8}$.

To additionally make use of source-side word classes, we append ele-

[1]The $\sqrt{N+1}$ factor serves to make the feature's influence independent of the number of word classes by yielding the same distance (under L_2) as $N+1$ identical copies of the feature.

ments analogous to the ones above to the vector, all further multiplied by a parameter α_{src} that allows trading off the relevance of source-side and target-side information. In the same fashion, we can incorporate multiple tagging schemes (e.g., word clusterings of different granularities) into the same feature vector. As finer-grained schemes have more elements in the feature vector than coarser-grained ones, and thus exert more influence, we set the α parameter for each scheme to $1/N$ (where N is the number of word classes of the scheme).

The K-means algorithm To create the clusters, we chose the K-means algorithm (Steinhaus, 1956; MacQueen, 1967) for both its computational efficiency and ease of implementation and parallelization. Given an initial mapping from the data points to K clusters, the procedure alternates between (i) computing the centroid of each cluster and (ii) re-allocating each data point to the closest cluster centroid, until convergence.

We implemented two commonly used initialization methods: Forgy and Random Partition. The Forgy method randomly chooses K observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds straight to step (ii). Forgy tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. As the resulting clusters looked similar, and Random Partition sometimes led to a high rate of empty clusters, we settled for Forgy.

6.3 Experiments

We evaluate our approach by comparing translation quality, as evaluated by the IBM-BLEU (Papineni et al., 2002) metric on the NIST Chinese-to-English translation task using MT04 as development set to train the model parameters λ, and MT05, MT06 and MT08 as test sets. Even though a key advantage of our method is its applicability to resource-poor languages, we used a language pair for which linguistic resources are available in order to determine how close translation performance can get to a fully syntax-based system. Accordingly, we use Chiang's hierarchical phrase-based translation model (Chiang, 2007) as a baseline, and the syntax-augmented MT model from Chapter 3 as a 'targetline', a model that would not be applicable for language pairs without linguistic resources.

We perform PSCFG rule extraction and decoding with the "SAMT" system (Chapter 3), using the provided implementations for the hierarchical and syntax-augmented grammars. To mitigate badly estimated PSCFG derivations based on low-frequency rules of the much sparser syntax model, the syntax grammar also contains the hierarchical grammar as a backbone (cf. Chapter 4 for details and empirical analysis).

We implemented our rule labeling approach within the SAMT rule extraction pipeline, resulting in comparable features across all systems. For all systems, we use the bottom-up chart parsing decoder implemented in the SAMT toolkit with a reordering limit of 15 source words, and correspondingly extract rules from initial phrase pairs of maximum source length 15. All rules have at most two nonterminal symbols, which must be non-consecutive on the source side, and rules must contain at least one source-side terminal symbol. The beam settings for the hierarchical system are 600 items per 'X'

(generic rule) cell, and 600 per 'S' (glue) cell.[2] Due to memory limitations, the multi-nonterminal grammars have to be pruned more harshly: We allow 100 'S' items, and a total of 500 non-'S' items, but maximally 40 items per nonterminal. For all systems, we further discard non-initial rules occurring only once.[3] For the multi-nonterminal systems, we generally further discard all non-generic non-initial rules occurring less than 6 times, but we additionally give results for a 'slow' version of the Syntax targetline system and our best word class based systems, where only single-occurrences were removed.

For parameter tuning, we use the L_0-regularized minimum-error-rate training tool provided by the SAMT toolkit. Each system is trained separately to adapt the parameters to its specific properties (size of nonterminal set, grammar complexity, features sparseness, reliance on the language model, etc.).

The parallel training data comprises of 9.6M sentence pairs (206M Chinese and 228M English words). The source and target language parses for the syntax-augmented grammar, as well as the POS tags for our POS-based grammars were generated by the Stanford parser (Klein and Manning, 2003).

The results are given in Table 6.3.1. Results for the Syntax system are consistent with previous results (Zollmann et al., 2008a), indicating improvements over the hierarchical system. Our approach, using target POS tags ('POS-tgt (no phr. s.)'), outperforms the hierarchical system on all three tests sets, and gains further improvements when accounting for phrase size ('POS-

[2]For comparison, Chiang (2007) uses 30 and 15, respectively, and further prunes items that deviate too much in score from the best item. He extracts initial phrases of maximum length 10.

[3]As shown in Zollmann et al. (2008a), the impact of these rules on translation quality is negligible.

	Dev (MT04)	MT05	MT06	MT08	**TestAvg**	Time
Hierarchical	38.63	36.51	33.26	25.77	**31.85**	14.3
Syntax	39.39	37.09	34.01	26.53	**32.54**	18.1
Syntax-slow	39.69	37.56	34.66	26.93	**33.05**	34.6
POS-tgt (no phr. s.)	39.31	37.29	33.79	26.13	**32.40**	27.7
POS-tgt	39.14	37.29	33.97	26.77	**32.68**	19.2
POS-src	38.74	36.75	33.85	26.76	**32.45**	12.2
POS-src&tgt	38.78	36.71	33.65	26.52	**32.29**	18.8
POS-tgt-slow	39.86	37.78	34.37	27.14	**33.10**	44.6
Clust-7-tgt	39.24	36.74	34.00	26.93	**32.56**	24.3
Clust-7-morph-tgt	39.08	36.57	33.81	26.40	**32.26**	23.6
Clust-7-src	38.68	36.17	33.23	26.55	**31.98**	11.1
Clust-7-src&tgt	38.71	36.49	33.65	26.33	**32.16**	15.8
Clust-7-tgt-slow	39.48	37.70	34.31	27.24	**33.08**	45.2
kmeans-POS-src&tgt	39.11	37.23	33.92	26.80	**32.65**	18.5
kmeans-POS-src&tgt-L_1	39.33	36.92	33.81	26.59	**32.44**	17.6
kmeans-POS-src&tgt-cosine	39.15	37.07	33.98	26.68	**32.58**	17.7
kmeans-POS-src&tgt ($\alpha_{\text{ins}} = .5$)	39.07	36.88	33.71	26.26	**32.28**	16.5
kmeans-Clust-7-src&tgt	39.19	36.96	34.26	26.97	**32.73**	19.3
kmeans-Clust-7..36-src&tgt	39.09	36.93	34.24	26.92	**32.70**	17.3
kmeans-POS-src&tgt-slow	39.28	37.16	34.38	27.11	**32.88**	36.3
kmeans-Clust-7..36-s&t-slow	39.18	37.12	34.13	27.35	**32.87**	34.3

Table 6.3.1: Translation quality in % case-insensitive IBM-BLEU (i.e., brevity penalty based on closest reference length) for Chinese-English NIST-large translation tasks, comparing baseline Hierarchical and Syntax systems with POS and clustering based approaches proposed in this work. 'TestAvg' shows the average score over the three test sets. 'Time' is the average decoding time per sentence in seconds on one CPU.

tgt'). The latter approach is roughly on par with the corresponding Syntax system, slightly outperforming it on average, but not consistently across all test sets. The same is true for the 'slow' version ('POS-tgt-slow').

The model based on bilingually tagged training instances ('POS-src&tgt') does not gain further improvements over the merely target-based one, but actually performs worse. We assume this is due to the huge number of nonterminals of 'POS-src&tgt' ($(2 * 33^2 + 33)(2 * 36^2 + 36) = 5.8$M in principle) compared to 'POS-tgt' ($2 * 36^2 + 36 = 2628$), increasing the sparseness of the grammar and thus leading to less reliable statistical estimates.

We also experimented with a source-tag based model ('POS-src'). In line with previous findings for syntax-augmented grammars (Zollmann and Vogel, 2010), the source-side-based grammar does not reach the translation quality of its target-based counterpart; however, the model still outperforms the hierarchical system on all test sets. Further, decoding is much faster than for 'POS-ext-tgt' and even slightly faster than 'Hierarchical'. This is due to the fact that for the source-tag based approach, a given chart cell in the CYK decoder, represented by a start and end position in the source sentence, almost uniquely determines the nonterminal any hypothesis in this cell can have: Disregarding part-of-speech tag ambiguity and phrase size accounting, that nonterminal will be the composition of the tags of the start and end source words spanned by that cell. At the same time, this demonstrates that there is hence less of a role for the nonterminal labels to resolve translational ambiguity in the source based model than in the target based model.

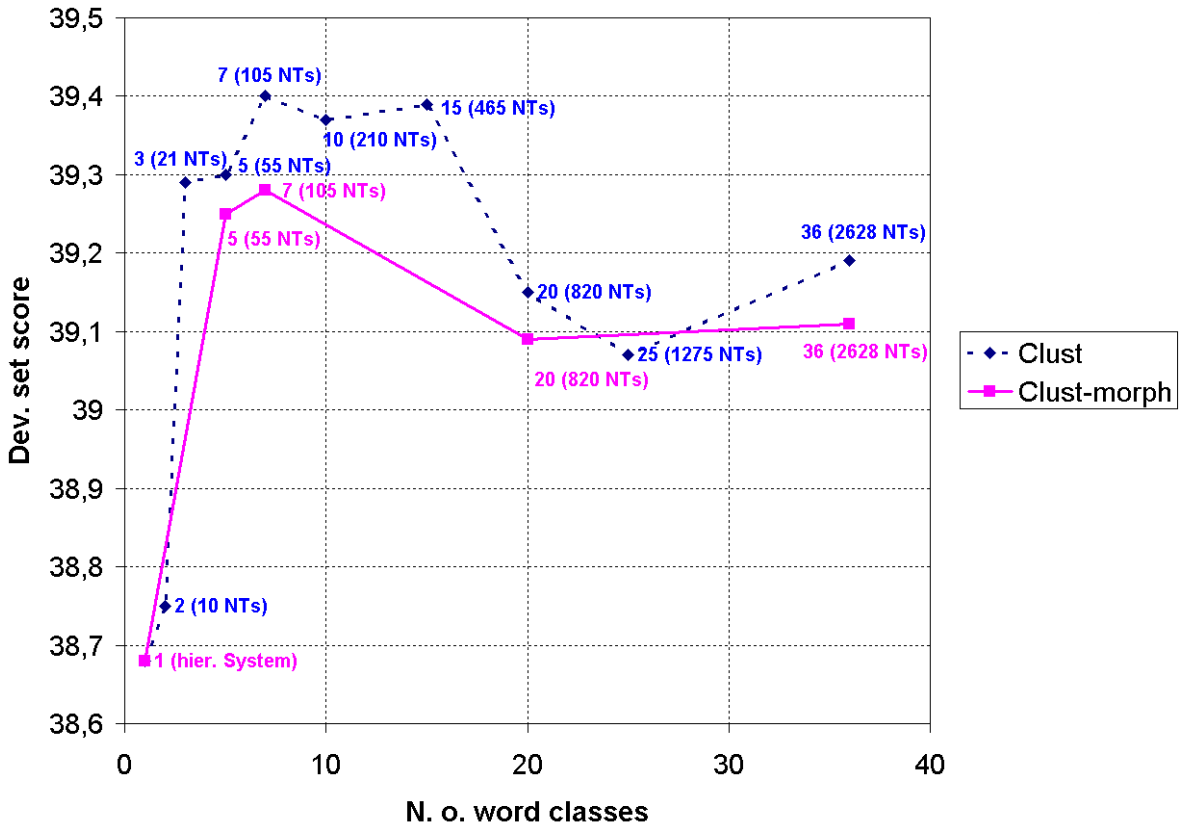

Figure 6.3.1: Performance of the distributional clustering model 'Clust' and its morphology-sensitive extension 'Clust-morph' according to L_0-penalized development set BLEU score for varying numbers N of word classes. For each data point N, its corresponding # nonterminals of the induced grammar is stated in parentheses.

6.3.1 Performance of the word-clustering based models

To empirically validate the unsupervised clustering approaches, we first need to decide how to determine the number of word classes, N. A straightforward approach is to run experiments and report test set results for many different N. While this would allow us to reliably conclude the optimal number N, a comparison of that best-performing clustering method to the hierarchical, syntax, and POS systems would be tainted by the fact that N was effectively tuned on the test sets. We therefore choose N merely based on development set performance. Unfortunately, variance in development set BLEU scores tends to be higher than test set scores, despite of SAMT MERT's inbuilt algorithms to overcome local optima, such as random restarts and zeroing-out. We have noticed that using an L_0-penalized BLEU score[4] as MERT's objective on the merged n-best lists over all iterations is more stable and will therefore use this score to determine N.

Figure 6.3.1 shows the performance of the distributional clustering model ('Clust') and its morphology-sensitive extension ('Clust-morph') (cf. Section 6.1.4) according to this score for varying values of $N = 1, \ldots, 36$ (the number Penn treebank POS tags, used for the 'POS' models, is 36).[5] For 'Clust', we see a comfortably wide plateau of nearly-identical scores from $N = 7, \ldots, 15$. Scores for 'Clust-morph' are lower throughout, and peak at $N = 7$.

Looking back at Table 6.3.1, we now compare the clustering models chosen by the procedure above—resulting in $N = 7$ for the morphology-unaware

[4]Given by: BLEU $-\beta \times |\{i \in \{1, \ldots, K\} | \lambda_i \neq 0\}|$, where $\lambda_1, \ldots, \lambda_K$ are the feature weights and the constant β (which we set to 0.00001) is the regularization penalty.

[5]All these models account for phrase size.

model (‘Clust-7-tgt’) as well as the morphology-aware model (‘Clust-7-morph-tgt’)—to the other systems. ‘Clust-7-tgt’ improves over the hierarchical baseline on all three test sets and is on par with the corresponding Syntax and POS targetlines. The same holds for the ‘Clust-7-tgt-slow’ version. We also experimented with a model variant based on seven source and seven target language clusters (‘Clust-7-src&tgt’) and a source-only labeled model (‘Clust-7-src’)—both performing worse.

Surprisingly, the morphology-sensitive clustering model (‘Clust-7-morph-tgt’), while still improving over the hierarchical system, performs worse than the morphology-unaware model. An inspection of the trained word clusters showed that the model, while superior to the morphology-unaware model in e.g. mapping all numbers to the same class, is overzealous in discovering morphological regularities (such as the ‘-ed’ suffix) to partition functionally only slightly dissimilar words (such present-tense and past-tense verbs) into different classes. While these subtle distinctions make for good partitionings when the number of clusters is large, they appear to lead to inferior results for our task that relies on coarse-grained partitionings of the vocabulary. Note that there are no ‘src’ or ‘src&tgt’ systems for ‘Clust-morph’, as Chinese, being a syllabic writing system, does not lend itself to morphology-sensitive clustering.

6.3.2 K-means clustering based models

To establish suitable values for the α parameters and investigate the impact of the number of clusters, we looked at the development performance over various parameter combinations for a K-means model based on source and/or tar-

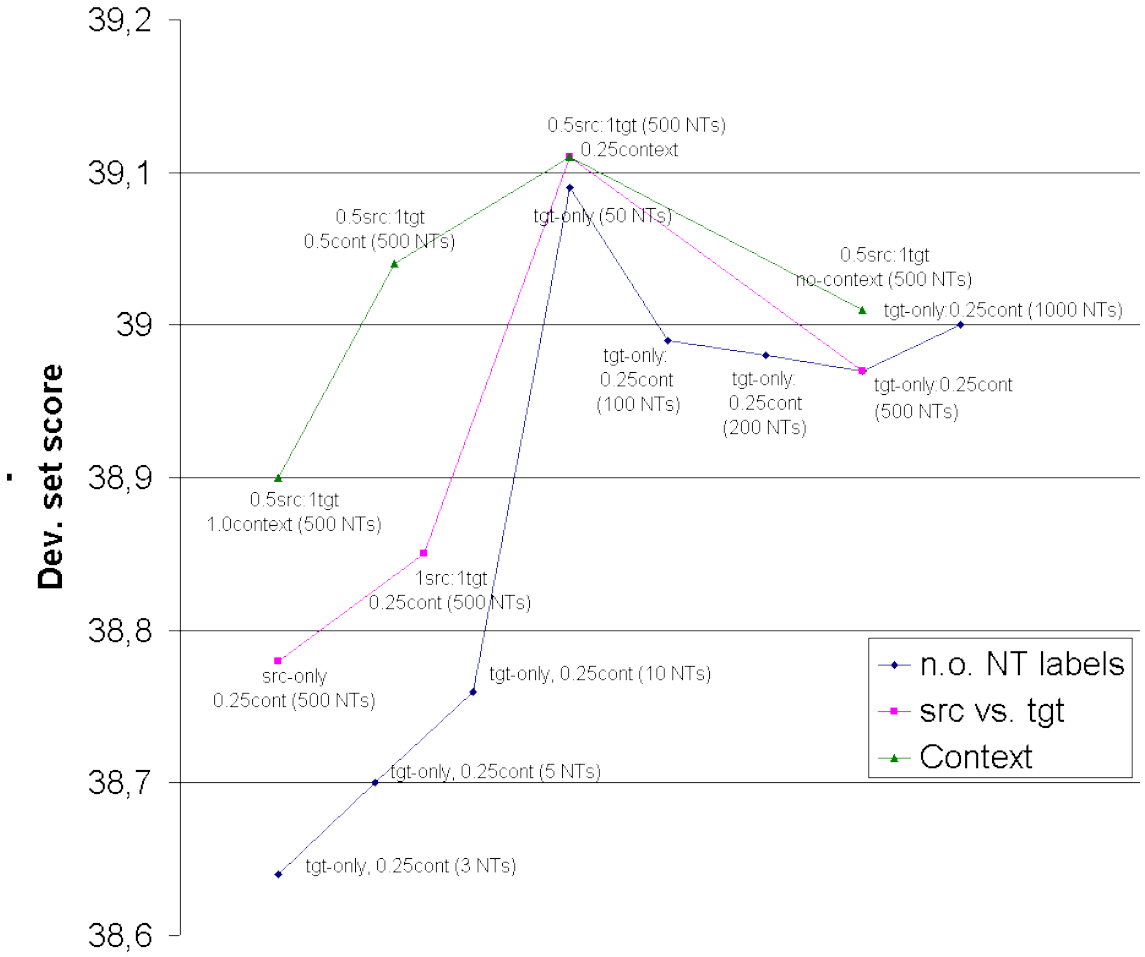

Figure 6.3.2: Dev. set performance of K-means for various #labels and values of α_{src} and α_{cntxt}. Note that there is no x-axis, as we only compare the development scores of the different parameterizations and thus horizontal stretching is for presentational purposes only.

get part-of-speech tags (Figure 6.3.2).[6] As explained in Section 6.3.1, we use an L_0-penalty adjusted MERT objective to decrease MERT fluctuation, but the differences in score are too small here to exclude the possibility of difference in performance due to chance. As can be seen quite clearly though, our method reaches its peak performance at around 50 clusters and then levels off slightly. Encouragingly, in contrast to the hard labeling procedure, K-means actually improves when adding source-side information. The optimal ratio of weighting source and target classes is 0.5:1, corresponding to $\alpha_{\text{src}} = .5$. Incorporating context information also helps, and does best for $\alpha_{\text{cntxt}} = 0.25$, i.e. when giving contexts 1/4 the influence of the phrase boundary words.

Entry 'kmeans-POS-src&tgt' in Table 6.3.1 shows the test set results for the development-set best K-means configuration (i.e., $\alpha_{\text{src}} = .5$, $\alpha_{\text{cntxt}} = 0.25$, and using 500 clusters). While beating the hierarchical baseline, it is only minimally better than the much simpler target-based hard labeling method 'POS-tgt'. We also tried K-means variants in which the Euclidean distance metric is replaced by the city block distance L_1 ('kmeans-POS-src&tgt-L_1') and the cosine dissimilarity ('kmeans-POS-src&tgt-cosine'), respectively, with slightly worse outcomes. Configuration 'kmeans-POS-src&tgt ($\alpha_{\text{ins}} = .5$)' investigates the incorporation of non-boundary word tags inside the phrase. Unfortunately, these features appear to worsen performance, presumably because given a fixed number of clusters, accounting for contents inside the phrase comes at the cost of neglect of boundary words, which are more relevant to producing correctly reordered translations.

The two completely unsupervised systems 'kmeans-Clust-7-src&tgt' (based on 7-class MKCLS distributional word clustering) and 'kmeans-Clust-

[6] We set $\alpha_{\text{sec}} = .25$, $\alpha_{\text{ins}} = 0$, and $\alpha_{\text{phrsize}} = .5$ throughout.

7..36-src&tgt' (using six different word clustering models simultaneously: all the MKCLS models from Figure 6.3.1 except for the two-, three- and five-class models) have the best results, outperforming the other K-means models as well as 'Syntax' and 'POS-tgt' on average, but not on all test sets.

Lastly, we give results for 'slow' K-means configurations ('kmeans-POS-src&tgt-slow' and 'kmeans-Clust-7..36-s&t-slow'). Unfortunately (or fortunately, from a pragmatic viewpoint), the models are outperformed by the much simpler 'POS-tgt-slow' and 'Clust-7-tgt-slow' models. Figure 6.3.3 shows automatically selected random phrase pairs from ten random clusters (out of 500) for system 'kmeans-Clust-7..36-s&t-slow'. While smaller phrases are being clustered quite well (e.g. the 'verb phrase followed by punctuation mark' cluster in the lower right-hand corner or the named-entities cluster in the lower left), the clusterings for large phrases are rather disappointing.

6.4 Related work

Hassan, Sima'an, and Way (2007) improve the statistical phrase-based MT model by injecting *supertags*, lexical information such as the POS tag of the word and its subcategorization information, into the phrase table, resulting in generalized phrases with placeholders in them. The supertags are also injected into the language model. Our approach also generates phrase labels and placeholders based on word tags (albeit in a different manner and without the use of subcategorization information), but produces PSCFG rules for use in a parsing-based decoding system.

Liang, Petrov, Jordan, and Klein (2007) propose a nonparametric

政府 在 设计 所 需 的 乡郊 改善 计划 工程 时 ， 是 经过 咨询	in fact , the various rural improvement programmes were mapped out after consultation with
） 任何 人 没有 遵从 或 拒绝 遵从	a person who fails or refuses to comply with
虽然 鼓励 升级 机构 考虑 到 调动	although promotion bodies are encouraged to take mobility into
拟议 大会 工作 时间表 和 会议 安排 （	proposed timetable of work for the assembly and the organization of meetings (
， 加强 基础 设施 建设 ， 集中 力量	, the construction of infrastructure facilities should be intensified and energies should be concentrated on
总统府 负责 与 人民 代表院 联系 和	in the presidency for relations with the house of representatives of the people and for
方案 将 侧重于 实施 和 初步 保持	programme will focus on the implementation and initial maintenance of
昂松戈 - 梅纳卡 动物 季节 保护区	ansongo - menaka , reserve temporaire de faune d '
大会 仅仅 举 几 个 例子 毫 无 例外 地 指出 ，	conference to name just a few bodies without exception , pointed the finger at
这些 弹药 和 其他 武器 方面 ， 联合 王国 政府 完全 遵守	of these rounds and other weaponry , her majesty ' s government complies strictly with
管理 交通 ， 政府 将 会 实行 一 个 全面 的 智能 交通 运输	traffic management , the administration will be implementing a comprehensive intelligent transport
江泽民 主席 ， 对 邓小平 逝世 表示 沉痛 哀悼	chinese president jiang zemin to express condolences over deng xiaoping 's death
中国 人民 从来 没有 畏惧 过 ， 一定 会 坚决 采取 行动 捍卫 国家 的	chinese people will never show fear and will definitely take resolute actions to safeguard national
志愿 人员 方案 规模 相当 小 而且 仍旧 限于 英语 培训	unv programme is rather small and still too confined to english language training
核武器 威胁 着 整 个 区域 ， 从 西地 中 海	nuclear weapons threatened an entire region , ranging from the western mediterranean
个 实质性 的 司 ， 负责 执行 该 部 的 标准	substantive divisions responsible for carrying out the department ' s normative
性别 ， 语言 ， 宗教 ， 政治 或 其他 见解 ， 财产	sex , language , religion , political or other opinions , property
决议 草案 A / 54 / L.83 / Rev 。 1	draft resolution a / 54 / l. 83 / rev. 1
其他 主礼 嘉宾 为 署理 房屋署 署长	other officiating guests at the opening ceremony include the acting director of housing
中华 人民 共和国 政府 是 中国 的 唯一 合法 政府 ， 台湾	prc government is the sole legitimate government of china , and that taiwan

这 毫 不 奇怪 ， 吕秀莲	, this is not surprising at all . annette lu
在 日内瓦 开幕 . 美国 总统 克林顿	opens in geneva u s president bill clinton
， 德国 ， 摩纳哥 ， 摩洛哥 ， 秘鲁	, germany , monaco , morocco , peru
， 在 被 强迫 返回 并 移交	, among those forcibly returned and handed over
， 委员会 将 收到	, the committee will have before it
发生 ， 刑事 情报 工作 将 更形	, criminal intelligence work will become more
及 皇后 军营 连接路 " 的	and queen 's lines link "
， 台湾 ， 新加坡 ， 马来西亚	, taiwan , singapore , malaysia
， 行政 和 组织 问题 以及 在	, administrative and organizational questions and in
兼 外长 钱其琛 和 哈萨克斯坦	and foreign minister qian qichen and kazakhstan
它们	their
现时	this
在	the
在	the
后	the
到	the
使	the
，	the
不	no
在	the

农村 的 非政府 组织 也 正在 推行	of rural non - governmental organizations were also carrying out
主管 维持 和平 行动 事务 副秘书长 发	under - secretary - general for peacekeeping operations made
除 规管 架构 外 ， 我们 正	outside the regulatory framework , we are making
， 对 他们 抱有 偏见 的 多数 人 对 他们	by the majority of the population , who are prejudiced against them
增加 伊拉克 平民 的 苦难 和 剥夺	at increasing the suffering of iraqi citizens and depriving
鉴于 中东 局势 恶化 ， 文莱	in light of the worsening situation in the middle east , brunei
今年 施政 报告 在 教毓 部分	in the part on education in this year ' s policy address
（ f ） 应 鼓励 民间 社会 的 参与	participation of civil society should be encouraged
来自 玻利维亚 ， 智利 ， 古巴 和 墨西哥	from bolivia , chile , cuba and mexico
（ i ） 或 （ c ） （ i	（ i
要求	call for
感谢	expressed appreciation
解释	explained in
呼吁	calls upon
设置	provided with
为了	aimed at
带来	brought about
自然	only natural
源于	originated in
基于	based on

g ） 职业	occupation
修订 ） 条例 草案	bill
消除 贫困 措施	poverty eradication measures
这 种 做法	such a course
摩根斯坦利 资本 国际 公司 世界 指数	msci world index
18 亿 美元	1.8 billion dollars
Castlereagh 拘留 中心	castlereagh detention centre
缅甸 ， 柬埔寨	myanmar , cambodia
不公平 贸易 做法 也 同样	unfair trade practices
财产 调查 委员会	property survey board
就 共同 关心	common concern
建设 开支 所	capital expenditure
海洋 生态 系统	marine ecosystems
现 有 合同	current contracts
石油 气 车辆	lpg vehicles
海湾 地区 问题	gulf issues
国际 社会 进行	international society
各 区域 组织	regional organizations
法律 制度 各	legal systems
国际 恐怖 分子	international terrorists

李光耀 与 陈焕友	lee and chen
普拉卡什 沙阿 大使	ambassador prakash shah
Jorge Carpio	mr. jorge carpio
让 - 巴蒂斯特· 巴	jean - baptiste
莫里斯 顿河	moriston , r.
Driss Houssein Khatari El	driss houssein khatari el
Fernndez Gmez 和 Alvarado	fernndez gmez and alvarado
（ 40%	（ 40 per cent
54 . 名	54 . ming
伊奎贝 先生 （	mr. ikouebe （
能 改变 的 。	to be changed .
应该 慎重 为好 。	should proceed with caution .
不 能 做 。	cannot be done .
没有 任何 旅行 计划 。	had no travel plans .
要 小心 研究 的 。	to be carefully studied .
受到 饥荒 的 威胁 。	are threatened with famine .
龚鹏程 说 ，	says kung peng - cheng .
考虑 其他 建议 。	to consider the other proposals .
也 在 制订 中 。	is also being developed .
趁早 做 准备 了 。	should make their preparations early .

Figure 6.3.3: Randomly sampled phrase pairs from ten random clusters (out of 500) for the K-means based unsupervised phrase pair clustering model ‘kmeans-Clust-7..36-s&t-slow’.

Bayesian model that can learn probabilistic context-free grammars. The technique is also used to improve on a syntactic parsing task, by learning to split each label in the training set into K subsymbols. Petrov, Haghighi, and Klein (2008) present a coarse-to-fine PSCFG decoder for syntax-based MT, in which word clustering of the target language is performed to compress the language model, enabling them to decode in multiple passes with increasingly more fine grained language models, thus trading off speed vs. search error. We use shallow syntactic analysis and word clustering not as a means to grammar refinement or approximative decoding, but rather as an alternative to fully syntax-based approaches.

Unsupervised *synchronous* grammar induction, apart from the contribution of Chiang (2005) discussed earlier, has been proposed by Wu (1997) for inversion transduction grammars, but like Chiang's model only uses a single generic nonterminal label (and no glue nonterminal). Blunsom, Cohn, Dyer, and Osborne (2009) present a nonparametric PSCFG translation model that directly induces a grammar from parallel sentences without constraints from a word-alignment model, and Cohn and Blunsom (2009) achieve the same for tree-to-string grammars, with encouraging results on small data. Our approach treats the training sentences' word alignments and phrase pairs, obtained from external modules, as ground truth and employs a straight-forward generalization of Chiang's popular rule extraction approach to labeled phrase pairs, resulting in a PSCFG with multiple nonterminal labels.

Our phrase pair clustering approach is similar in spirit to the work of Lin and Wu (2009), who use K-means to cluster (monolingual) phrases and use the resulting clusters as features in discriminative classifiers for a named-entity-recognition and a query classification task. Phrases are represented

in terms of their contexts, which can be more than one word long; words within the phrase are not considered. Further, each context contributes one dimension per vocabulary word (not per word class as in our approach) to the feature space, allowing for the discovery of subtle semantic similarities in the phrases, but at much greater computational expense. Another distinction is that Lin and Wu (2009) work with phrase types instead of phrase instances, obtaining a phrase type's contexts by averaging the contexts of all its phrase instances.

Nagata, Saito, Yamamoto, and Ohashi (2006) present a reordering model for machine translation, and make use of clustered phrase pairs to cope with data sparseness in the model. They achieve the clustering by reducing phrases to their head words and then applying the MKCLS tool to these pseudo-words.

Kuhn, Chen, Foster, and Stratford (2010) cluster the phrase pairs of an SMT phrase table based on their co-occurrence counts and edit distances in order to arrive at semantically similar phrases for the purpose of phrase table smoothing. The clustering proceeds in a bottom-up fashion, gradually merging similar phrases while alternating back and forth between the two languages.

The idea of using (tag-level) contexts for unsupervised learning of phrase classes underlies both the distributional model of Clark (2001) and the generative Constituent-Context Model of Klein and Manning (2002). Both models aim only to learn syntactic constituents, i.e., phrase spans that are yielded by a single node in the (inferred) phrase-structure tree, whereas we need to be able to provide *any* phrase span with a label. As demonstrated in Figure 3 of Klein and Manning (2002), which shows constituents and distituents,

as context vectors, projected onto their first two principal components, with constituents well-separable and distituents badly separable, the task of distinguishing between constituents and distituents based on context vectors is very hard; therefore, and application of these models to our problem does not look very promising. A variant of the Constituent Context Model in which the beginning-of-phrase and end-of-phrase tags are generated explicitly in addition to the contexts (instead of being generated as part of the constituent/distituent yield) might be more successful.

6.5 Conclusion

In this chapter, we proposed methods of labeling phrase pairs to create automatically learned PSCFG rules for machine translation. Crucially, our methods only rely on "shallow" lexical tags, either generated by POS taggers or by automatic clustering of words into classes. Evaluated on a Chinese-to-English translation task, our approach improves translation quality over a popular PSCFG baseline—the hierarchical model of Chiang (2005) —and performs on par with the syntax-augmented model of Chapter 3, using heuristically generated labels from parse trees. Using automatically obtained word clusters instead of POS tags yields essentially the same results, thus making our methods applicable to all languages pairs with parallel corpora, whether syntactic resources are available for them or not.

We also propose a more flexible way of obtaining the phrase labels from word classes using K-means clustering. While currently the simple hard-labeling methods perform just as well, we hope that the ease of incorporating new features into the K-means labeling method will spur interesting future research.

CHAPTER 7

Conclusions and Future Work

In this thesis, we developed several new methods for machine translation based on PSCFGs with multiple nonterminals. While we originally designed these methods to be informed by source and/or target phrase-structure parse trees, we showed how to adapt the models to make use only of part-of-speech analysis or even completely unsupervised word clusters, without degradation in translation performance. We further proposed and evaluated a simple way of widening the in- and output "pipelines" between individual modules of our MT system by moving from hard *single-best* decisions to distributions over inputs and outputs approximated by N-best lists.

We started out with syntax-augmented machine translation (SAMT), which extracts a PSCFG from a bilingual corpus in which target sentences are annotated with phrase-structure parse trees (Chapter 3). This was the first syntax-based MT system to achieve an improvement over phrase-based MT (Zollmann and Venugopal, 2006). We showed how to parallelize the system under the MapReduce paradigm, and reported experimental results comparing SAMT to phrase-based and hierarchical phrase-based MT for multiple language pairs. We reported improvements over these baselines for French-to-English, Chinese-to-English, and Urdu-to-English, but failed to obtain improvements for Spanish-to-English and Arabic-to-English. We drew the conclusion that SAMT (as well as hierarchical phrase-based MT) fails to outper-

form phrase-based MT for language pairs that have mainly short-range word reordering.

We then proposed several improvements to hierarchical phrase-based MT and syntax-augmented MT (Chapter 4). We added a source span length model that, for each rule utilized in a probabilistic synchronous context-free grammar (PSCFG) derivation, gives a confidence estimate in the rule based on the number of source words spanned by the rule and its substituted child rules, resulting in small improvements for hierarchical phrase-based as well as syntax-augmented MT. We further demonstrated the utility of combining hierarchical and syntax-based PSCFG models and grammars.

We also compared the original SAMT, which extracts rules based on target-side syntax, to a corresponding variant based on source-side syntax, showing that target syntax is more beneficial, and unsuccessfully experimented with a model extension that jointly takes source and target syntax into account. A more promising future avenue for source and target syntax combination is the simultaneously developed approach of Chiang (2010), who supplies his hierarchical MT model with separate binary features, each indicating whether a given source or target syntactic constituent is present. This approach could be applied to a multi-nonterminal grammar such as our SAMT grammar by adding such features based on the source parses. Unfortunately, due to the large number of features of such a model, minimum-error-rate training then becomes infeasible and alternative tuning approaches such as that of Chiang et al. (2008) have to be used, implementations of which are not publicly available and hard to reproduce due to the amount of technical detail in the algorithm.

In Chapter 5, we demonstrated the feasibility and benefits of widening

the MT pipeline to include additional evidence from N-best alignments and parses. We integrate this diverse knowledge under a principled model that uses a probability distribution over these alternatives. We achieved significant improvements in translation quality over grammars built on "single-best" evidence alone when considering N-best alignments, while N'-best parses seem to have no impact on translation quality. Using a relatively small number of additional alternative alignments results in significant improvements in quality, with minimal impact on the number of rules in the grammar and the translation runtime for a hierarchical system, but at significantly increased grammar size and runtime for a syntax-augmented system.

Finally, we proposed methods of creating multi-nonterminal PSCFG rules just from "shallow" lexical tags (Chapter 6). It turned out that, at least for Chinese-to-English NIST, translation quality is as good as that of SAMT. Using automatically obtained word clusters instead of POS tags yields essentially the same results, thus making our methods applicable to all languages pairs with parallel corpora, whether syntactic resources are available for them or not. Future work should confirm our findings for other than the Chinese-to-English NIST translation task, and include alternative parameterizations of the clustering used to generate word-tags, with a focus on evaluating performance degradation when moving to limited resource conditions.

We also proposed a more flexible way of obtaining the phrase labels from word classes using K-means clustering. While currently the simple hard-labeling methods perform just as well, we hope that the ease of incorporating new features into the K-means labeling method will spur interesting future research. For example, the notorious Chinese 'DE' word, which, depending on its role in the sentence, can trigger long-range reorderings of noun phrases,

could easily be incorporated into the clustering process by means of features indicating its presence before, at the beginning, at the end, or after the phrase.

The clustering model could also show a way towards mastering the challenge of moving from syntax to semantics in machine translation: semantic features such as role labels can directly be employed.

When considering the constraints and independence relationships implied by each labeling approach, we can distinguish between approaches that label rules differently within the context of the sentence that they were extracted from, and those that do not. The syntax-augmented model is at one end of this spectrum. A given target span might be labeled differently depending on the syntactic analysis of the sentence that it is a part of. On the other extreme, the clustering based approach labels phrases based on the contained words alone.[1] The POS grammar represents an intermediate point on this spectrum, since POS tags can change based on surrounding words in the sentence; and the position of the K-means model depends on the influence of the phrase contexts on the clustering process. Context *insensitive* labeling has the advantage that there are less alternative left-hand-side labels for initial rules, producing grammars with less rules, whose weights can be more accurately estimated. This could explain the strong performance of the word-clustering based labeling approach. Our results also suggest a hybrid approach, where instead of following the heuristics in Chapter 3 to label non-constituent spans, word-tag based labels are used instead.

Hierarchical phrase-based MT suffers from spurious ambiguity: A single translation for a given source sentence can usually be accomplished by

[1]Note, however, that the creation of clusters itself did take the context of the clustered words into account.

many different PSCFG derivations. This problem is exacerbated by syntax-augmented MT with its thousands of nonterminals, and made even worse by its joint source-and-target extension. Future research should apply the work of Blunsom, Cohn, and Osborne (2008) and Blunsom and Osborne (2008), who marginalize over derivations to find the most probable translation rather than the most probable derivation, to these multi-nonterminal grammars.

All algorithms developed in this work were incorporated into the open-source SAMT toolkit, co-written with Ashish Venugopal and available at: `www.cs.cmu.edu/~zollmann/samt`

LIST OF FIGURES

LIST OF TABLES

REFERENCES

Alfred V. Aho and Jeffrey D. Ullmann. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 1969. 18

Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. A systematic analysis of translation model search spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 224–232, 2009. 22

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2005. 23

Alexandra Birch, Phil Blunsom, and Miles Osborne. A quantitative analysis of reordering phenomena. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 197–205, Athens, Greece, March 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W09-0434`. 55

Hans Ulrich Block. Example based incremental synchronous interpretation. In *Vermobil: Foundations of Speech-to-Speech Translation*, 2000. 9

Phil Blunsom and Miles Osborne. Probabilistic inference for machine translation. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 215–223, Morristown, NJ, USA, 2008. Association for Computational Linguistics. 117

Phil Blunsom, Trevor Cohn, and Miles Osborne. A discriminative latent variable model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008. 117

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of ACL*, Singapore, August 2009. URL `http://www.aclweb.org/anthology/P/P09/P09-1088`. 110

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2), 1990. 52

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2), 1993. 9, 17, 36, 69

Xavier Carreras and Michael Collins. Non-projective parsing for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 200–209, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D/D09/D09-1021`. 53

J.C. Chappelier and M. Rajman. A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of Tabulation in Parsing and Deduction (TAPD)*, pages 133–137, Paris, 1998. URL `citeseer.ist.psu.edu/chappelier98generalized.html`. 20

Eugene Charniak. A maximum entropy-inspired parser. In *Proceedings*

of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL), 2000. 74, 76

Yu Chen and Andreas Eisele. Hierarchical hybrid translation between english and german. In Viggo Hansen and Francois Yvon, editors, *Proceedings of the 14th Annual Conference of the European Association for Machine Translation*, pages 90–97. EAMT, EAMT, 5 2010. 58

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005. 3, 9, 10, 11, 13, 20, 21, 25, 26, 29, 38, 42, 43, 57, 86, 110, 112, 118

David Chiang. Hierarchical phrase based translation. *Computational Linguistics*, 33(2), 2007. 20, 49, 77, 99, 100

David Chiang. 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 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P10-1146`. 59, 114

David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, October 2008. URL `http://www.aclweb.org/anthology/D08-1024`. 16, 58, 91, 114

Alexander Clark. Unsupervised induction of stochastic context free grammars with distributional clustering. In *Proc. of Conference on Computa-*

tional Natural Language Learning, pages 105–112, Toulouse, France, July 2001. 111

Alexander Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL)*, pages 59–66, 2003. 94

Trevor Cohn and Phil Blunsom. A Bayesian model of syntax-directed tree to string grammar induction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2009. URL `http://portal.acm.org/citation.cfm?id=1699510.1699557`. 110

Doug Cutting and Eric Baldeschwieler. Meet Hadoop. In *O'Reilly Open Software Convention*, Portland, OR, 2007. 34

Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data process on large cluster. In *Proceedings of Symposium on Operating System Design and Implementation*, 2004. 34, 35

John DeNero, Dan Gillick, James Zhang, and Dan Klein. Why generative phrase models underperform surface heuristics. In *Proc. of the NAACL Workshop on Statistical Machine Translation*, 2006. 70

Christopher Dyer, Aaron Cordova, Alex Mont, and Jimmy Lin. Fast, easy, and cheap: Construction of statistical machine translation models with mapreduce. In *Proceedings of the Workshop on Statistical Machine Translation, ACL*, 2008a. 36

Christopher Dyer, Smaranda Muresan, and Philip Resnik. Generalizing word

lattice translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008b. 72

J. Earley. An efficient context-free parsing algorithm. *Communications of the Assocation for Computing Machinery*, 13(2):94–102, 1970. 20

Michael Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*, 2004. 9, 27, 52, 53, 63

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inferences and training of context-rich syntax translation models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2006. 52

Kuzman Ganchev, Joao V. Graca, and Ben Taskar. Better alignments = better translations? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008. 72

Kevin Gimpel and Noah A. Smith. Feature-rich translation by quasi-synchronous lattice parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 219–228. Association for Computational Linguistics, 2009. ISBN 978-1-932432-59-6. URL `http://dl.acm.org/citation.cfm?id=1699510.1699539`. 54

Greg Hanneman. Automatically improved category labels for syntax-based statistical machine translation: Thesis proposal. Technical report, Carnegie Mellon University, January 2011. 65

Greg Hanneman and Alon Lavie. Decoding with syntactic and non-syntactic phrases in a syntax-based machine translation system. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL/HLT 2009*, pages 1–9, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W09-2301`. 54

Hany Hassan, Khalil Sima'an, and Andy Way. Supertagged phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June 2007. 108

Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007. 20

T. Kasami. An efficient recognition and syntax-analysis algorithm for context-free languages. Technical report, Air Force Cambridge Research Lab, 1965. 19

Dan Klein and Christoper Manning. Accurate unlexicalized parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003. 64, 100

Dan Klein and Christopher D. Manning. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 128–135, 2002. 111

Reinhard Kneser and Hermann Ney. Improved clustering techniques for

class-based statistical language modelling. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 973–976, Berlin, Germany, 1993. 86, 94

Kevin Knight and Jonathan Graehl. An overview of probabilistic tree transducers for natural language processing. Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), 2005. 52

Philipp Koehn and Christof Monz, editors. *Proceedings of the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, New York City, June 2006. 41

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*, 2003. 9, 17, 26, 27, 41, 45, 70, 71, 72, 75, 76, 87

Philipp Koehn, Franz J. Och, and Daniel Marcu. Pharaoh: A beam search decoder for phrase-base statistical machine translation models. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, 2004. 9, 16, 25, 42

Roland Kuhn, Boxing Chen, George Foster, and Evan Stratford. Phrase clustering for smoothing TM probabilities - or, how to extract paraphrases from phrase tables. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 608–616, Beijing, China, August 2010. URL `http://www.aclweb.org/anthology/C10-1069`. 111

Alon Lavie, Alok Parlikar, and Vamshi Ambati. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the ACL/HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 87–95, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0411`. 53

P. Liang, S. Petrov, M. I. Jordan, and D. Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, 2007. 108

Dekang Lin and Xiaoyun Wu. Phrase clustering for discriminative learning. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2009. URL `http://portal.acm.org/citation.cfm?id=1690219.1690290`. 110, 111

Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006. 53, 63

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. Joint decoding with multiple translation models. 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 576–584, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P09/P09-1065`. 54

J. B. MacQueen. Some methods for classification and analysis of multivariate

observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. 98

Daniel Marcu and Daniel Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118711. URL `http://www.aclweb.org/anthology/W02-1018`. 70

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia, 2006. 53

Jonathan May and Kevin Knight. A better N-best list: Practical determinization of weighted finite tree automata. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*, 2006. 52

Haitao Mi and Liang Huang. Forest-based translation rule extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008. 53

Haitao Mi, Liang Huang, and Qun Liu. Forest-based translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008. 53, 72

Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. A clustered global phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 713–720, 2006. doi: http://dx.doi.org/10.3115/1220175.1220265. URL `http://dx.doi.org/10.3115/1220175.1220265`. 111

Franz J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003. 16

Franz J. Och and Hermann Ney. A systematic comparison of various alignment models. *Computational Linguistics*, 29(1), 2003. 72, 74

Franz J. Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4), 2004. 9, 26, 46, 70, 75

Franz J. Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, 1999. 16

Franz Josef Och. An efficient method for determining bilingual word classes. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL)*, pages 71–76, 1999. doi: http://dx.doi.org/10.3115/977035.977046. 94

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings*

of the Annual Meeting of the Association for Computational Linguistics (ACL), 2002. 22, 41, 48, 63, 77, 99

Michael Paul. Overview of the IWSLT 2006 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2006. 71

Matthias Paulik, Kay Rottmann, Jan Niehues, Silja Hildebrand, and Stephan Vogel. The ISL phrase-based MT system for the 2007 ACL workshop on statistical MT. In *Proc. of the Association of Computational Linguistics Workshop on Statistical Machine Translation*, 2007. 44

Slav Petrov, Aria Haghighi, and Dan Klein. Coarse-to-fine syntactic machine translation using language projections. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008. 110

Arjen Poutsma. Data-oriented translation. In *Proceedings of the 18th conference on Computational linguistics*, pages 635–641, Morristown, NJ, USA, 2000. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/992730.992738. 9, 51, 53

Chris Quirk and Simon Corston-Oliver. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006. 72

Chris Quirk, Arul Menezes, and Collin Cherry. Dependency tree translation: Syntactically informed mt. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005. 72

R. Scha. Taaltheorie en taaltechnologie; competence en performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek, LVVN-jaarboek*, pages 7–22, Almere, The Netherlands, 1990. English translation as: Language Theory and Language Technology; Competence and Performance; http://iaaa.nl/rs/LeerdamE.html. 51

David A. Smith and Jason Eisner. Minimum-risk annealing for training log-linear models. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL)*, pages 787–794, Sydney, July 2006. 16

Mark Steedman. *The Syntactic Process*. MIT Press, 2000. 28

Hugo Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci. Cl. III. 4*, pages 801–804, 1956. 98

Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the Condor experience. *Concurrency - Practice and Experience*, 2005. 35

Ashish Venugopal. *Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation*. PhD thesis, Carnegie Mellon University, 2008. 34

Ashish Venugopal and Andreas Zollmann. Grammar based statistical MT on Hadoop: An end-to-end toolkit for large scale PSCFG based MT. *The Prague Bulletin of Mathematical Linguistics*, 91:67–78, 2009. 26

Ashish Venugopal, Andreas Zollmann, and Alex Waibel. Training and evaluating error minimization decision rules for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*,

pages 208–215, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W05/W05-0836. 16

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*, 2007. 20

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. Wider pipelines: N-best alignments and parses in MT training. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, 2008. 69

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2007. 16

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962. ISSN 00401706. doi: 10.2307/1266577. URL http://dx.doi.org/10.2307/1266577. 61

D. H. D. West. Updating mean and variance estimates: an improved method. *Commun. ACM*, 22(9):532–535, 1979. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/359146.359153. 61

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3), 1997. 20, 70, 110

Yong-Zeng Xue, Sheng Li, Tie-Jun Zhao, Mu-Yun Yang, and Jun Li. Bilingual phrase extraction from n-best alignments. In *Proceedings of the International Conference on Innovative Computing, Information and Control (ICICIC)*, 2006. 72

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA, 2001. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073012.1073079. 9, 52

Kenji Yamada and Kevin Knight. A decoder for syntax-based statistical mt. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. 26, 52

Richard Zens and Hermann Ney. Discriminative reordering models for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation, HLT/NAACL*, 2006. 46

Ying Zhang and Stephan Vogel. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the Conference of the European Association for Machine Translation (EAMT)*, Budapest, Hungary, May 2005. The European Association for Machine Translation. 42

Andreas Zollmann and Ashish Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation, HLT/NAACL*, 2006. 10, 13, 26, 54, 63, 113

Andreas Zollmann and Stephan Vogel. New parameterizations and features for PSCFG-based machine translation. In *Proceedings of the 4th Workshop*

on Syntax and Structure in Statistical Translation (SSST), Beijing, China, 2010. URL http://www.aclweb.org/anthology/W10-3814. 57, 102

Andreas Zollmann and Stephan Vogel. A word-class approach to labeling PSCFG rules for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1001. 86

Andreas Zollmann, Ashish Venugopal, Stephan Vogel, and Alex Waibel. The CMU-UKA Syntax Augmented Machine Translation System for IWSLT-06. In *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006. 26

Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. The syntax augmented mt (samt) system for the shared task in the 2007 acl workshop on statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation, ACL*, 2007. 26

Andreas Zollmann, Ashish Venugopal, Franz J. Och, and Jay Ponte. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the Conference on Computational Linguistics (COLING)*, 2008a. 20, 22, 26, 31, 45, 62, 100

Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. The CMU Syntax-Augmented Machine Translation System: SAMT on Hadoop with N-best Alignments. In *Proc. of the International Workshop on Spoken Language Translation*, pages 18–25, Hawaii, USA, 2008b. 26, 69