Automatically Improved Category Labels for Syntax-Based Statistical Machine Translation

Greg Hanneman
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Ph.D. Thesis Proposal
Draft: January 10, 2011

Thesis Committee:
Alon Lavie (chair), Carnegie Mellon University
Stephan Vogel, Carnegie Mellon University
Noah Smith, Carnegie Mellon University
David Chiang, University of Southern California

Abstract

A common modeling choice in syntax-based statistical machine translation is the use of synchronous context-free grammars, or SCFGs. When training a translation model in a supervised setting, an SCFG is extracted from parallel text that has been statistically word-aligned and parsed by monolingual statistical parsers. However, the set of syntactic category labels used in a monolingual statistical parser is decided upon quite independently of the machine translation task, and there is no guarantee that it is optimal for a bilingual SCFG or for machine translation at all.

In this thesis, we first demonstrate that the set of category labels used in a machine translation system's grammar strongly affects three inter-related characteristics of the system: spurious ambiguity, rule sparsity, and reordering precision. We propose using these characteristics as the basis for evaluating the properties of an SCFG both outside of and within an actual translation task. Finally, as our main work, we propose three automatic relabeling methods that will create a better set of category labels for a given language pair and choice of automatic parsers. These methods involve clustering and collapsing unnecessary labels, splitting existing labels into multiple subtypes, and swapping specific instances of existing labels to correct for local errors. Improved properties of the grammar and improved translation results will be demonstrated for at least two language pairs.

Contents

Chapter 1

Introduction and Setup

1.1 MT with Synchronous Context-Free Grammar

In the last decade, researchers working on syntax-based statistical machine translation have been increasingly successful at developing hierarchical models of translation that explicitly incorporate the structure of the sentences being translated. Syntactic models have proved particularly useful for language pairs, such as German–English or Chinese–English, where long-distance reordering of words or phrases must be carried out. Models of hierarchical structure provide a way of generalizing a very large number of possible word strings to a more manageable number of abstract patterns. Word grouping and reordering, then, can be carried out in a principled way according to the linguistic divergences between languages that are captured in the patterns.

A common modeling choice in syntax-based statistical MT is the use of synchronous context-free grammars, or SCFGs (Aho and Ullman, 1969). Under an SCFG, a source-language string and a target-language string are produced simulaneously by applying a series of re-write rules. For example, in French we may have a grammar rule that specifies how to create a noun phrase (NP) such as *la voiture bleue* out of a determiner (DET), a noun (N), and an adjective (ADJ). This CFG rule is shown in Rule 1.1. A corresponding NP rule in English for *the blue car*, using the Penn Treebank part-of-speech labels for a determiner (DT), base adjective (JJ), and a singular common noun (NN), is shown in Rule 1.2.

$$\text{NP} \rightarrow \text{DET N ADJ} \tag{1.1}$$

$$\text{NP} \rightarrow \text{DT JJ NN} \tag{1.2}$$

We can produce both the French NP and the English NP simultaneously in the single SCFG rule shown in Rule 1.3.

$$\text{NP} :: \text{NP} \rightarrow [\text{DET}^1\ \text{N}^2\ \text{ADJ}^3] :: [\text{DT}^1\ \text{JJ}^3\ \text{NN}^2] \tag{1.3}$$

In an SCFG rule, we use the string "::" to separate the source and target languages, which may use different labels, on each side of the rule. On the right-hand side, we indicate correspondences between nonterminals with coindexes on the labels. For instance, the superscript "2" in Rule 1.3 means that the translation of the N in French (e.g. *voiture*) corresponds to the NN in English (e.g. *car*).

Whereas in the past SCFG rules like these would have been written by human experts with knowledge of both English and French, in modern syntax-based MT they are extracted automatically, according to various schemes, from parallel text that has been statistically parsed and word-

aligned.[1] The focus of this thesis will be the particular nonterminal category labels that appear in a given SCFG. As we have introduced them so far, these labels $\ell_s :: \ell_t$ are made up of a nonterminal ℓ_s from the source-language parse tree and a nonterminal ℓ_t from the target-language tree. Distinguishing ℓ_s from ℓ_t — and copying them both directly from parse trees — is only one point in the space of possible labelings that could be given to an SCFG. In practice, grammar extraction schemes have varied on the level of labeling detail used in their implementations of SCFG-based translation.

David Chiang's Hiero system (Chiang, 2005), one of the most well-known SCFG-based decoders, represents the simplest extreme of the label spectrum, using the single label X for all rules and phrase pairs. Grammar extraction in Hiero is actually not based on linguistic syntax at all: the parallel corpus is not parsed, and SCFG rules are extracted by finding subphrases with standard phrase-based statistical MT heuristics (Koehn, Och, and Marcu, 2003).

Moving into linguistic syntax, GHKM rule extraction (Galley et al., 2004) operates on a parallel corpus that has been parsed on the target side only. Labels from the target-side parse trees are used in the rules, but source-side labels remain as generic Xs. Liu, Liu, and Lin (2006) present a symmetric approach, with syntactic labels on the source side and generic X labels on the target side. The numerous grammar extraction techniques that use parse trees on both sides (Lavie, Parlikar, and Ambati, 2008; Zhechev and Way, 2008; Liu, Lü, and Liu, 2009) result in SCFG rules labeled with the more complex joint labels we have already introduced. Again, the exact inventory of source- and target-side labels used comes directly from the underyling parse trees.

Other systems have extended the label space beyond that defined by the parsers. The so-called Syntax-Augmented MT (SAMT) system (Zollmann and Venugopal, 2006), though it is still based on Hiero-style non-linguistic phrase extraction heuristics, replaces the generic X category label with labels derived from a target-side parse tree. If the target side of a phrase pair corresponds to a syntactic constituent in the sentence from which it was extracted, it is labeled with that constituent. For the majority of phrase pairs, where this is not the case, a more complex label is created: a label of the form A+B if the phrase spans adjacent syntactic nodes of type A and B in the parse tree, a label of the form A/B if the phrase would span a node of type A if it had a node of type B immediately to the right, and a label of the form A\B if the phrase would span a node of type A if it had a node of type B immediately to the left. A new SCFG-based MT system by Chiang (2010) applies the SAMT labeling conventions to the source side as well, again producing joint labels from two parse trees.

In summary, the exact labels in use within an SCFG-based MT system fall into three broad types: (1) completely generic X labels, (2) labels taken directly from the underlying parser or parsers, and (3) expanded labels based on creative combinations of parser labels. In the latter two cases, there is an additional dimension of expressive power: whether the labels are based on syntax from one or both sides of the language pair being translated. Put together, these schemes define a large space of possible labelings, with many intermediate points left unexplored. Most SCFG-based MT systems that extract rules from a parallel parsed corpus take a middle ground: the node labels assigned by the parser or parsers are directly used in the SCFG rules of the system.

However, there is a fundamental problem with this approach. The set of part-of-speech and nonterminal labels used in a statistical parser to annotate each word and constituent is decided upon quite independently of the MT task. The set is typically defined by the human linguists who

[1] The scope of this thesis is limited to syntax-based MT systems that learn their grammars in a supervised way from parse trees; we are not considering systems that perform unsupervised grammar induction from unparsed texts.

constructed the treebank that the statistical parser was trained from, and it can vary greatly by language and treebank. There is no *a priori* guarantee that the set of parts of speech and non-terminal labels separately decided upon for treebank development is optimal even for monolingual statistical parsing, let alone MT done in conjunction with a set of labels equally separately decided upon in the other language. We will discuss this problem and its fallout in more detail in Chapter 2; here, we will simply summarize the issue and state some conclusions.

At model-training time, the estimation of the translation model is tightly bound up in the choice of SCFG category labels, as many models include scoring features that take into account the label of each phrase pair or grammar rule. The optimal set of category labels is one that avoids modeling spurious ambiguity on the one hand while not throwing out important distinctions between nonterminals on the other. A bilingual label set that is too coarse risks generalizing over systematic divergences in languages, while an overly fine label set fragments the observed data into spurious categories. Thus, a key aspect of getting the "right" category labels is balancing spurious ambiguity with labeling granularity.

At run time, a syntactic decoder must keep track of many alternative derivations, even if their target-language output strings are the same, because derivation pieces with different labels may allow for different rules to be applied on top of them. Rule applications drive reordering in syntax-based MT systems, so it is important to have parsed translation fragments with the correct labels to plug into the right-hand sides of larger rules. On the other hand, duplicating the same output string with many different labelings also leads to extra work for the decoder, and what would otherwise be a high-scoring translation fragment may instead have its probability mass split over a needlessly large collection of alternative labelings. Again we note the problem of spurious ambiguity — this time in opposition to a problem we call rule sparsity, when important rules cannot apply because their nonterminals do not match.

In sum, we frame what we call the "label problem" in syntax-based MT as a search over the space of possible SCFG labeling schemes given a particular language pair and choice of automatic parsers. This search must evaluate the fitness of a labeling scheme according to a number of dimensions. As introduced above, a labeling scheme must not lead to a grammar that allows too many possible labelings for a particular rule; this leads to spurious ambiguity in the grammar and wasted effort in the decoder. Nor, given a large overall label set, should there be too few possible labelings for an individual reordering pattern; this makes it less likely that the correctly labeled right-hand-side items will be available to plug into a reordering rule at run time. A third dimension is the notion of precision. While a small label set will largely avoid problems of spurious ambiguity or rule sparsity, it will also by necessity be less precise, smoothing out differences between syntactic categories and allowing reordering rules to overapply.

1.2 Thesis Statement

We have now introduced the setup for syntactic machine translation based on synchronous context-free grammars and quickly surveyed the space of labels that may be used in SCFGs for MT. We have also introduced the claim that finding the optimal point within the label space for a given language pair is important, as suboptimal points could have large effects on the performance and efficiency of syntactic MT systems. Although existing MT systems use grammar extraction algorithms that inhabit various points in the label space, there is much territory in between, and techniques geared towards finding better points in that space remain a largely unaddressed challenge in the field.

In this thesis, we therefore propose to explore the space of possible SCFG labels with a series of automatic relabeling methods that will create a better label set for a given language pair and choice of syntactic parsers. First of all, we propose the creation of a general-purpose grammar extractor to address a number of shortcomings in our baseline method (Section 3.1). We then intend to develop three techniques for automatic relabeling: collapsing of category labels based on alignment (Section 3.2), refining of category labels based on bilingual information (Section 3.3), and correction of mislabeled parse tree nodes based on an existing grammar (Section 3.4).

The automatic relabeling methods proposed in this thesis will result in an overall scheme for selecting an improved label set for a given language pair and choice of parsers (Section 3.5). We claim that the improved label set will demonstrate:

- a statistically significant increase in overall translation quality as measured by automatic metric scores;
- a significant reduction in spurious ambiguity as measured by properties of the grammar, the number of chart entries present during decoding, and overall system run time; and
- a significant reduction in rule sparsity as measured by properties of the grammar and test-set reference reachability.

As part of demonstrating these claims, we define spurious ambiguity (Section 2.1), rule sparsity (Section 2.2), and reordering precision (Section 2.3) as three characteristics of an SCFG that are significantly influenced by the choice of label set. Precise definition and meaningful measurements of these characteristics (Section 2.4) will be a contribution of this thesis. As listed above, they will also serve as part of our evaluation metrics for determining improvements.

1.3 Experimental Environment

We plan to carry out experiments in French–English, Chinese–English, and Arabic–English translation, plus the reverse direction (English–foreign) for one of these language pairs. All three language pairs have large parallel corpora and monolingual parsers available for our experimental work, and there are community-standard test sets on which to compare results. French, Arabic, and Chinese also represent increasing distances from English in terms of language typology and the amount of reordering required in translation. Each relabeling technique from Chapter 3 will be tested on at least two language pairs.

For French–English we have used, and plan to continue using, corpora released as part of the Workshop on Machine Translation (WMT) series. The most recent release, for WMT 2010, was made up of 31.5 million sentence pairs from four domains.[2] After throwing out Web-crawled data, this leaves 8.6 million sentence pairs of high-quality European Parliament, news commentary, and United Nations data. The WMT workshop series also includes an open competition each year for MT systems built using the provided data. We have already participated in WMT 2008, 2009, and 2010; going forward, we plan to contribute improved systems to WMT 2011 and 2012, which will allow a direct comparision of our relabeling techniques with the state of the art.

Large Arabic–English and Chinese–English corpora are provided periodically at another series of workshops, the NIST Open MT evaluations.[3] The NIST 2009 Arabic–English data, for example,

[2] See `www.statmt.org/wmt10/translation-task.html` for details on the WMT 2010 data.

[3] Details on the NIST Open MT evaluation series can be found at `www.itl.nist.gov/iad/mig/tests/mt`.

was made up of approximately 5 million parallel sentences. NIST evaluations also include the release of standard test sets for system-to-system comparison. We previously participated in the NIST 2009 Arabic–English evaluation; future plans are to contribute improved systems to either the Arabic–English or Chinese–English tracks of future NIST evaluations as they are announced.

For both Arabic and Chinese, we also plan to conduct more rapid prototyping using a subset of all available training data in order to run initial experiments and test techniques. In Chinese–English, we plan to make use of the FBIS corpus of approximately 300,000 sentence pairs. In Arabic–English, we will create a similarly sized corpus by making a selection from the most recent NIST data. Although performance on standard test sets will consequently be artificially low until construction of a full-sized system, the smaller prototypes will lead to more efficient experimentation in the early stages.

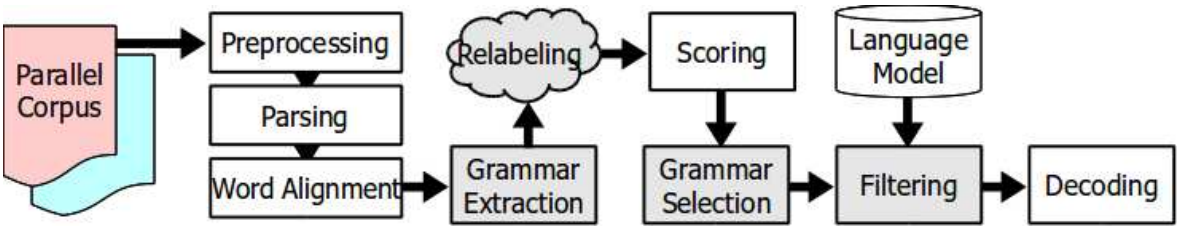


Figure 1.1: Our experimental setup for building SCFG-based MT systems. Components shaded in grey fall within the scope of this thesis.

We define a common experimental platform for our MT systems as follows, shown graphically in Figure 1.1. First, the initial parallel training data is preprocessed, statistically word-aligned with MGIZA++ (Gao and Vogel, 2008), and parsed with the applicable monolingual statistical parsers — typically those created at the University of California at Berkeley (Petrov and Klein, 2007) or Stanford University (Klein and Manning, 2003) that are freely available online. As an additional step, the resulting parse trees may be systematically transformed, such as by binarization or the removal of unary rule chains. Next, we extract SCFG rules according to the generalized method of Section 3.1.[4] Extracted rule instances are counted and scored according to the syntactic translation model features defined by Hanneman, Clark, and Lavie (2010) using Jon Clark's PhraseDozer package.[5]

Previous MT systems built in this way have included non-syntactic phrase pairs in addition to syntactic phrase pairs and grammar rules. Hanneman and Lavie (2009) extracted non-syntactic phrase pairs from the same training corpus according to standard phrase-based SMT extraction heuristics, then retained those that provided coverage for source sides that had not been extracted syntactically. The additional phrase pairs were given dummy syntactic labels, then dumped into the translation model and scored as if they were syntactic. Hanneman, Clark, and Lavie (2010) used a more sophisticated approach, sharing extraction counts between syntactic and non-syntactic copies of the same phrase pair and providing extra translation model features for distinguishing between

[4]Currently existing baseline systems referred to later in this document as being constructed according to this setup do not yet use the generalized rule extractor.

[5]With large amounts of training data, the construction of the translation model is a signficant computational challenge. In our pipeline, many of the data-intensive steps are carried out as MapReduce jobs on Yahoo's M45 research computing cluster or as multi-core jobs on the Language Technologies Institute's departmental cluster.

the use of syntactic and non-syntactic rules at decoding time. While including more phrase pairs in the system improved automatic metric scores in both cases, the non-syntactic phrases clouded the analysis of what derivations could be built using only syntactic rules. Since our focus in this thesis is very much on the power of different syntactic grammars, we choose to exclude non-syntactic phrases in our experimental systems in order to support a cleaner analysis of the syntactic rules' behavior and effectiveness during decoding.

Decoding itself will be carried out with the Joshua decoder (Li et al., 2009), an open-source platform for SCFG-based MT. Using the entire extracted grammar, particularly when its label set is large, presents a significant engineering and computational challenge to the decoder. We combat this in three ways: first by restricting the number of hierarchical grammar rules present in the final translation model, second by filtering terminal phrase pairs and the system's language model to only those portions relevant to the test set being translated, and third by running Joshua as a seven-threaded job on a server with 32 GB of RAM. In the case of rule filtering, recent experiments have settled on preserving the top 10,000 most frequently extracted hierarchical rules as a compromise between insufficiently complete grammars and prohibitiviely long decoding times.

Components shaded in grey in Figure 1.1 fall within the scope of this thesis. The central project, of course, is developing and testing the updated grammar extraction module and the series of relabeling techniques proposed in Chapter 3. We identify two further components whose improvement, in a practical sense, goes hand-in-hand with the success of improved grammar extraction and labeling. Grammar selection, as we have stated, currently involves skimming off some number of the most frequently extracted hierarchical rules. Unfortunately, this method does not distinguish between partially lexicalized rules, which may only apply in limited contexts, and fully abstract rules, which apply much more productively (or over-productively). Their all-nonterminal right-hand sides also cannot be filtered in advance to a particular test set. Additionally, our current filtering step does *not* filter even partially lexicalized rules given a test set; it only affects the (admittedly more numerous) phrase pairs. We envision making updates to both of these modules to allow for more sophisticated grammar selection and filtering.

The overall experimental pipeline is designed and managed with the LoonyBin workflow manager (Clark and Lavie, 2010). LoonyBin contributes to the re-use of code by wrapping each experimental step in a separate "tool descriptor" that can subsequently be dropped into any LoonyBin pipeline. We expect to release the final relabeling techniques developed in this thesis as individual LoonyBin tools that can be easily re-used by other experimenters.

1.4 Related Work

There has been surprisingly little work to date in syntax-based MT on analyzing, modifying, or optimizing the label set chosen for any language or pair of languages, despite the important role that the choice of labels plays in the construction of an MT system.

Working still in the domain of monolingual statistical parsing, Petrov et al. (2006) describe an EM-style algorithm for automatically splitting a set of category labels used in a monolingual treebank to learn a more informative parsing grammar. During each iteration's expectation step, rule probabilities for the current grammar are calculated from a Penn Treebank traning set; to match the training tree labels, any categories that have been refined in earlier iterations are collapsed together for the estimation. In the maximization step of the algorithm, each existing label is split into two copies, the rule probabilities are slightly perturbed, and then the grammar is retrained

using the larger set of symbols. The interesting result is a surprisingly detailed set of refined syntactic categories — for example, an NNP-15 subcategory for the first part of a two-word proper place name (*San* or *Wall*) and a separate NNP-3 category for the second part (*Francisco* or *Street*). At the nonterminal level, 32 different VP subcategories separately handle infinitive verb phrases, main-clause verb phrases, subordinate-clause verb phrases, and others. These promising results show that meaningful hidden categories can be automatically detected in a monolingual setting, but the technique has not yet been applied bilingually.

Huang and Knight (2006) discuss two methods for relabeling parse trees in order to boost the quality of an MT system with syntax on one side. Also taking the position that the baseline Penn Treebank tag set is too coarse, most of their experiments focus on splitting existing category labels by lexicalizing certain types of tree nodes. Nodes can be annotated either with head information from their descendents (such as renaming DT nodes to DT_this or DT_these, etc., depending on the exact determiner found under the node) or with contextual information from their surroundings in the tree (such as the label of the node's parent, or whether the node is its parent's left, right, or medial child). While the authors report significant gains in the output quality of a Chinese–English system for a refined model using five types of relabeling, their modifications were carried out monolingually in a string-to-tree translation system with parses on the English side only. In concentrating in this thesis on systems that model both source- and target-side syntactic structure, we believe further label refinements can be made.

Garera, Callison-Burch, and Yarowsky (2009) address the relabeling problem bilingually and at the part-of-speech level, but in the context of extracting a word-to-word translation lexicon from two monolingual corpora. Their tag-matching algorithm uses source- and target-side monolingual corpora that have been part-of-speech tagged in order to estimate a distribution of labels for each source or target word. Then, given a small gold-standard lexicon, the likelihood of mapping from a particular source-side part-of-speech tag to a target-side tag is computed by accumulating the monolingual tag probabilities over all entries in the lexicon.

Among MT applications, Venugopal et al. (2009) and Chiang (2010) implement on-the-fly grammar modification by allowing SCFG rules to apply in the decoder even when their labels do not match. In the first case (Venugopal et al., 2009), rules are extracted with the linguistically motivated expanded label set of the Syntax-Augmented MT (SAMT) system, but rules that are identical up to labelings are abstracted into a single rule using only the default X nonterminal as in Hiero. The unlabeled rule's distribution over possible labelings is saved as a "preference grammar," which can be used at decoding time to provide a soft measure of the label of any parsed subtree. Chiang's (2010) system, in the second case, also uses expanded SAMT-style labels, but they are explicitly preserved in the extracted rules. Instead, a rule may apply at a non-matching label site during decoding, and the translation model is augmented with a number of features keeping track of which labels have been substituted for which other labels. Weights for these numerous features are learned during tuning of the entire translation model, so that a test-time match or mismatch can be assigned a label-specific penalty or bonus score.

Chapter 2

Label-Based SCFG Properties

We introduced the "label problem" in SCFG-based MT in Section 1.1 as stemming from the fact that the category labels used in a monolingual statistical parser were developed independently from any machine translation task — and even from any monolingual statistical parsing task. Instead, the labels typically come from those used in monolingual treebanks: sets of sentences that have been manually labeled and parsed by human linguists. Treebanks exist for a growing number of languages, including English (Marcus, Santorini, and Marcinkiewicz, 1993), French (Abeillé, Clément, and Toussenel, 2003), and Chinese (Xue et al., 2005). Let us now investigate in more detail what happens when the syntactic category labels decided on in these treebanks are used in MT systems.

The Berkeley statistical parser (Petrov and Klein, 2007) trained on the English Penn Treebank produces a total of 71 different syntactic labels: 45 at the word level and 26 for multi-word constituents. Trained from the French Paris 7 Treebank, the parser produces 21 word-level and 12 constituent-level tags, for a total of 33. In simple SCFG grammar extraction, such as the method described in Section 3.1.1, this potentially leads to rules involving up to $33 \cdot 71 = 2343$ unique labels. The number can be reduced by adding a constraint to the SCFG extraction that prohibits preterminal (word-level) nodes from corresponding to nonterminal (multi-word) nodes, and vice versa. In that case, the number of possible unique labels drops to $(12 \cdot 26) + (21 \cdot 45) = 1257$. In a quick motivational experiment, we extracted a grammar using the correspondence constraint from 9.6 million French and English parallel sentences parsed with the Berkeley parsers.[1] In the results, we found 1078 of the 1257 possible label pairs, or 86 percent. The 31 most common among them each appeared more than a million times; however, 336 (31 percent) were seen fewer than 100 times each, a sort of conventional-wisdom cutoff for the number of data points needed to reliably estimate a model parameter.

But is 1078 unique category labels a reasonable number for French–English translation? If not, is it too high or too low? These are research questions that so far have been very little explored in the context of syntax-based MT, and in this chapter we claim that they are important.

The annotation guide for the Penn Treebank specifies three separate tags for adjectives. The tag for most adjectives (e.g. *good*, *large*) is JJ, but comparative adjectives (e.g. *better*, *larger*) are marked as JJR and superlative adjectives (e.g. *best*, *largest*) are marked as JJS (Santorini and MacIntyre, 1995). In the Paris 7 French treebank, on the other hand, adjectives of any type are simply given

[1] This corpus is derived from data resources made available for the 2009 Workshop on Machine Translation; see `http://www.statmt.org/wmt09`.

the A tag (Abeillé and Clément, 2003), with subtypes that are not reflected in treebank-derived parsers. Whether or not an SCFG-based MT system preserves six possible joint labels for adjective translation — as part of the grammar's overall inventory of labels for all types of words and phrases — can have far-reaching effects on the capabilities of the system.

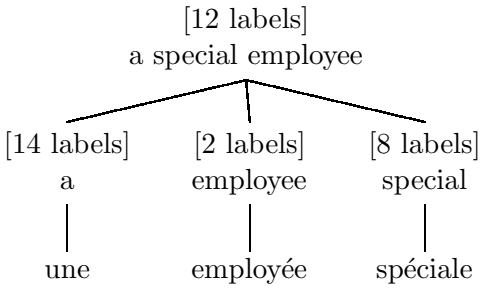
In the following sections, we identify three aspects of an SCFG-based MT system that are influenced by the system's choice of label set. We define and illustrate each one precisely, and then we conclude this chapter by proposed methods by which to measure them meaningfully for a given grammar.

2.1 Spurious Ambiguity

Spurious ambiguity is a well-known problem in both monolingual parsing (as stated by e.g. Pareschi and Steedman (1987)) and in parsing-based MT. Chiang (2005) defined it as the condition of having "many derivations that are distinct yet have the same model feature vectors and give the same translation," but the term is more generally used without the constraint that the derivations must appear model-identical in terms of score. Examples of spurious ambiguity include a phrase pair with an excessive number of possible left-hand-side labelings, or the instantiation of the same reordering pattern on the right-hand side of a rule with an overly large number of possible right-hand-side labelings. In brief, it is the problem of having "too many labels." Spurious ambiguity has its roots in the creation of the parsed parallel corpus itself, as word alignment errors, tagging errors, bracketing errors, and non-compositional translations combine to produce evidence for incorrect labelings of a right-hand side. The effects of spurious ambiguity are felt both in the estimation of the translation model at training time and in the application of that model at run time.

In model estimation, each additional left-hand-side label for a given right-hand side weakens any model score that conditions on all or part of the right-hand side, as that right-hand side's probability mass is further and further divided. Assume a generic SCFG rule of the form $\ell_s :: \ell_t \rightarrow [r_s] :: [r_t]$. The Stat-XFER decoder used left-hand-side labels as a component of its joint probability distributions, computing $P(\ell_s, \ell_t, r_s \,|\, r_t)$ and $P(\ell_s, \ell_t, r_t \,|\, r_s)$. More recent work by the Carnegie Mellon statistical transfer MT group (Hanneman, Clark, and Lavie, 2010) has used label probability features $P(\ell_s, \ell_t \,|\, r_s)$, $P(\ell_s, \ell_t \,|\, r_t)$, and $P(\ell_s, \ell_t \,|\, r_s, r_t)$. Spurious ambiguity can also lead to flattening of distributions conditioned on the left-hand side, as more right-hand sides are vying for a left-hand side's probability mass. The Syntax-Augmented MT system (Zollmann and Venugopal, 2006), for example, computes $P(r_s, r_t \,|\, \ell_t)$ as one of its features.

At run time, spurious ambiguity forces the decoder to maintain additional chart entries for each variant labeling of a translation fragment. Chart cells, however, are typically limited to a maximum number of highest-scoring entries, to only those entries whose scores are within a certain margin from the cell's best-scoring entry, or to some combination of both thresholds. Other possible entries whose scores fall below the cutoffs are pruned out of the search space and not stored. Thus, filling chart cells with a large number of variant labelings for the same translation fragment may result in other entries falling out of the search space. With large chart size limits, it will result in much extra work in applying rules. The sample French phrase *une employée spéciale*, for instance, may be translated to *a special employee* in English with a large variety of labels, as in Figure 2.1. In a baseline extracted French–English SCFG using the WMT 2010 data from Section 1.3, the translation of *une* to *a* can be done with 14 possible syntactic labels, *employée* to *employee* with two, and *spéciale* to *special* with eight. From those building blocks, there are 48 applicable rules to



une :: **a**
D::DT, PRO::DT, D::SYM, D::FW, D::NNP, ADV::DT, A::DT, N::DT, PRO::SYM, D::LS, PRO::NNP, NP::X, PRO::FW, ADV::SYM
employée :: **employee**
N::NN, V::NN
spéciale :: **special**
A::JJ, N::JJ, A::NN, AP::ADJP, AP::NP, NP::NP, V::JJ, N::NN
une employée spéciale :: **a special employee**
NP::NP, AP::NP, AdP::NP, PP::NP, VPpart::NP, VN::NP, Srel::NP, NP::X, NP::S, NP::PP, NP::NX, NP::FRAG

Figure 2.1: The translation of French *une employée spéciale* into English *a special employee* is derivationally very ambiguous in a baseline French–English SCFG.

produce a parse node with the desired English output — and that node itself may have 12 different labels!

Even if pruning is completely avoided, additional labels also weaken probabilities of translation *strings* under the Viterbi approximation. That is, when calculating the scores of possible outputs, most decoders do not sum over all possible derivations producting the same output string; instead, they merely search for the output string that has the highest scoring single derivation (cf. Li, Eisner, and Khudanpur (2009)). Therefore, one output string that can be produced by many possible derivations of moderate score will be ignored in favor of a string that can be produced in exactly one way with a higher score.

2.2 Rule Sparsity

Rule sparsity is a complementary problem to spurious ambiguity: instead of the grammar containing too many rules, it rather does not have enough. Rule sparsity occurs when a rule carrying out a particular reordering pattern should be applied, but the right-hand-side labels for the available rules encoding that pattern do not match the parsed fragments that have been built. This notion of not having the right rule has been informally referred to in the literature as general "data sparsity" (Setiawan and Resnik, 2010) or identified as fallout from an overly strong "matching constraint" at rule substitution nodes (Chiang, 2010). Here, however, we prefer our own term "rule sparsity" because it localizes the blame for missing information to the thing that is actually missing — a grammar rule.

Consider the French noun phrase *l' office monétaire de Hongkong* and its desired English translation *the Hong Kong monetary office*. In order for this translation to be produced by an SCFG-based MT system, a large number of rules must be present and applied in the decoder. First, the individual French words must be translated to the correct English equivalents, requiring for example Rules 1 through 4 in Figure 2.2. Next, in order to carry out the correct reording for *office monétaire* $\leftrightarrow$ *monetary office*, something like Rule 5 is needed. Obviously, Rule 5 must exist in the translation model, and its right-hand-side nonterminals N :: NN and A :: JJ must match labels that have been assigned to *office* $\leftrightarrow$ *office* and *monétaire* $\leftrightarrow$ *monetary*, respectively. A similar constraint is in effect

(1) D :: DT → [l'] :: [the]
(2) N :: NN → [office] :: [office]
(3) A :: JJ → [monétaire] :: [monetary]
(4) NP :: NP → [Hongkong] :: [Hong Kong]
(5) NP :: NP → [N^1 A^2] :: [JJ^2 NN^1]
(6) NP :: NP → [NP^1 de NP^2] :: [NP^2 NP^1]

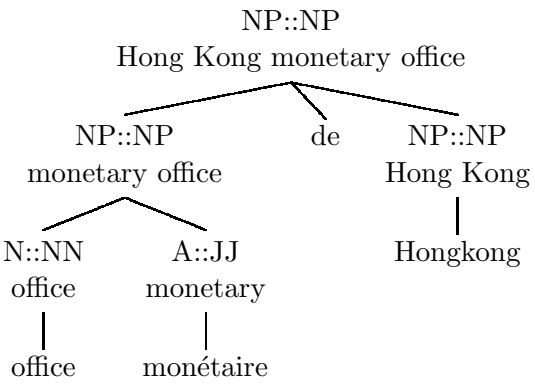


Figure 2.2: Necessary grammar rules and their required applications to produce the English translation *the Hong Kong monetary office* from the French input *l' office monétaire de Hongkong*.

for making the final reordering: both *office monétaire* ↔ *monetary office* and *Hongkong* ↔ *Hong Kong* must have been identified as NP :: NP in order to apply Rule 6.

If any of these requirements are not met, then the English translation cannot be reordered correctly. For example, if Rule 4 were replaced with the alternative — and still reasonable — Rule 4′ below

$$(4') \text{ N :: NP} \rightarrow [\text{Hongkong}] :: [\text{Hong Kong}]$$

then instead of Rule 6, we must now have an alternative rule

$$(6') \text{ NP :: NP} \rightarrow [\text{NP}^1 \text{ de N}^2] :: [\text{NP}^2 \text{ NP}^1]$$

in order to successfully create the desired English translation. Without Rule 6′, the necessary reordering is not carried out due to rule sparsity.

The key point with respect to SCFG labeling schemes is that the labels in use on the right-hand sides of rules control which derivations may be built, particularly with regard to reordering patterns. What derivations may be built, in turn, specify the space of possible output translations.

2.3 Reordering Precision

As we have seen, the criterion that labels must match in rule application can lead to rule sparsity problems when the labeled right-hand side required to carry out a specific reordering is not found. However, this label-matching criterion also plays an important watchdog role in making sure that reordering rules are not over-applied in contexts where their reordering patterns are not licensed by the original data. An SCFG labeling scheme therefore must also enforce a certain amount of precision in the grammar in terms of which rules may apply in which situations. Our discussion of reordering precision as a grammar property is founded on one of the basic arguments for applying linguistic syntax to the MT task: that a set of hierarchical categories can do a better job of explaining word reordering between two languages than properties of the word strings alone.

Consider the generic French–English reordering pattern

$$\text{X :: X} \rightarrow [\text{X}^1 \text{ de X}^2] :: [\text{X}^2 \text{ X}^1] \tag{2.1}$$

where X can stand for any nonterminal. In our baseline WMT 2010 extracted grammar, it is instantiated by 216 unique rules, where the choice of the set of labels $\{X^1, X^2\}$ is restricted to 162 possible pairs. The entire grammar, however, uses an inventory of 1134 total labels, meaning that there are $1134 \cdot 1134 = 1{,}285{,}956$ possible ways to draw any pair of them. Thus, the reordering pattern specified in Rule 2.1 is only allowed to be carried out in a very limited range of circumstances — mainly when X^1 represents a French noun translated to an English noun and when X^2 represents a French noun or noun phrase translated to an English noun, noun phrase, or adjective. This enforced precision places a major control on the decoder's search space.

The granularity of this control is obviously dependent on the granularity of the grammar's labels. Figure 2.3 compares the most frequent instantiations of the reordering pattern in Rule 2.1 with those of the similar Rule 2.2:

$$\text{X :: X} \rightarrow [\text{X}^1 \text{ de } \text{X}^2] :: [\text{X}^1 \text{ X}^2] \tag{2.2}$$

While both reordering patterns are frequently applied to nouns, it is the finer-grained labeling derived from the target side that more effectively discriminates between preferring to apply Rule 2.1 when the head noun is a plural common noun ($X^1 = N :: NNS$) and Rule 2.2 when the structure of the target-side translation more resembles a verb phrase. The use or non-use of such finer-grained labels represents a trade-off between increased precision in reordering rules on the one hand and decreased problems of rule sparsity on the other.

Instantiations of Rule 2.1:		**Instantiations of Rule 2.2:**	
$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{N}^2] :: [\text{NN}^2 \text{ NNS}^1]$	9304	$\text{PP :: PP} \rightarrow [\text{P}^1 \text{ de } \text{NP}^2] :: [\text{IN}^1 \text{ NP}^2]$	14,263
$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{N}^2] :: [\text{JJ}^2 \text{ NNS}^1]$	1833	$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{N}^2] :: [\text{NNP}^1 \text{ NNP}^2]$	3809
$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{N}^2] :: [\text{NN}^2 \text{ NN}^1]$	1797	$\text{NP :: VP} \rightarrow [\text{N}^1 \text{ de } \text{NP}^2] :: [\text{VBG}^1 \text{ NP}^2]$	3471
$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{NP}^2] :: [\text{NP}^2 \text{ NNS}^1]$	1535	$\text{NP :: VP} \rightarrow [\text{N}^1 \text{ de } \text{NP}^2] :: [\text{VB}^1 \text{ NP}^2]$	2465
$\text{NP :: NP} \rightarrow [\text{N}^1 \text{ de } \text{N}^2] :: [\text{NNP}^2 \text{ NNP}^1]$	1173	$\text{NP :: NP} \rightarrow [\text{ADV}^1 \text{ de } \text{N}^2] :: [\text{JJR}^1 \text{ NN}^2]$	1987

Figure 2.3: The most frequent instantiations, by number of extracted instances, of two related French–English SCFG reordering patterns.

2.4 Measuring These Properties

In our thesis statement (Section 1.2), we proposed using properties of an SCFG as a measurement for determining improvements made to that grammar by relabeling techniques. We have so far examined three such properties in the preceding sections; to conclude this chapter, we propose methods by which to measure them.

Spurious ambiguity, rule sparsity, and reordering precision can all to some extent be measured by simple counting. In the case of spurious ambiguity, we can compute the number of variant labelings for a given right-hand side or reordering pattern, then aggregate counts for an entire extracted grammar or for the portion of it used on a particular test set. Reordering precision can be measured by the fraction of possible labeled instantiations present in the grammar for a given reordering pattern: having fewer instantiations means a more precise grammar, but also one that suffers more from rule sparsity. We can also count up the number of total rules, or reordering rules, that are applied in the decoder when translating a given test set and compare it to the

number of words that are still out of order when the output translations are aligned to reference translations. This way, rule sparsity is indicated when few rules apply during decoding and the output is produced in an incorrect order.

However, a deeper understanding of these properties' effect on an actual MT task requires a deeper look into the decoding process. In addition to simple counts, we propose to measure spurious ambiguity, rule sparsity, and reordering precision within a run-time system. Practically, spurious ambiguity is affected by the total model score of a translation fragment in addition to any pruning heuristics applied at decoding time. We aim to take these factors into account by examining the completed output chart or hypergraph produced by the Joshua decoder on a sentence-by-sentence basis and counting in each cell the number of variant labelings given to identical string outputs.

Rule sparsity and reordering precision are both related to the notion of what derivations can and cannot be built at translation time. Auli et al. (2009) used the term "induction error" to indicate when the correct target-language output is not reachable within an MT system's search space. Induction error is measurable via constrained decoding: given an input sentence for which the reference translation is known, the translation model is pruned to only those rules that are useful in generating the reference from the source sentence. Such strict pruning in practice negates the effect of any pruning heuristics in the decoder, so the absolute reachability of the reference translation can be verified. We plan to use constrained decoding in Joshua to similar ends. Not only will this allow us to use reference reachability as an explicit measure of grammar quality, but it will also provide a platform for meaningful manual error analysis as the techniques from the next chapter are being developed.

Chapter 3

Grammar Extraction and Relabeling

3.1 A General-Purpose Rule Extractor

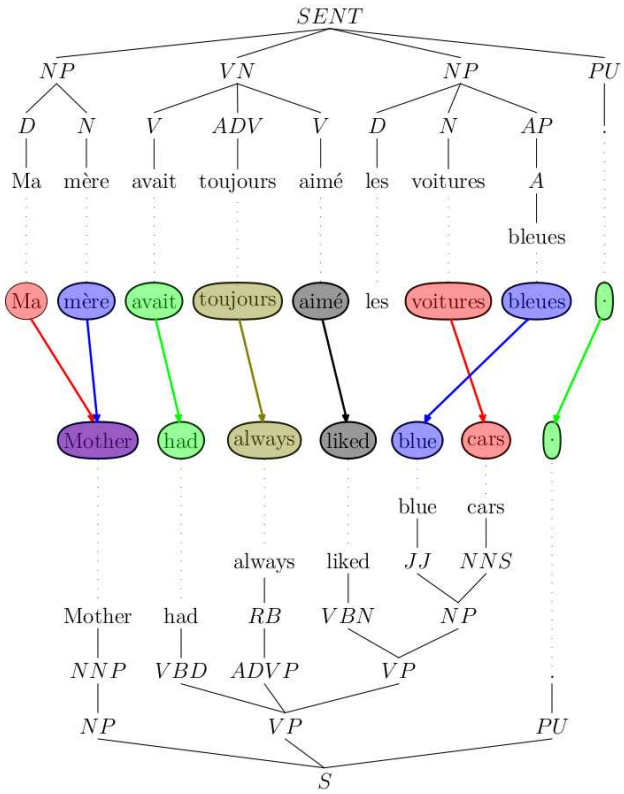
3.1.1 Motivation

Our baseline method of node alignment and SCFG grammar extraction is the method of Lavie, Parlikar, and Ambati (2008). Given a word-aligned and parsed sentence pair, this method identifies translational equivalents ("node alignments") between the two parse trees according to support from the word alignments. A node n_s in one parse tree S will be aligned to a node n_t in the other parse tree T if all the words in the yield of n_s are either all aligned to words within the yield of n_t or have no alignments at all. (This is analagous to the word alignment consistency constraint in phrase-based statistical MT's phrase extraction heuristics.) If there are multiple nodes n_t satisfying this constraint, the node in T closest to the leaves will be selected. Then SCFG rules can be extracted from adjacent levels of aligned nodes, which specify points at which the tree pair can be decomposed into minimal SCFG rules. In addition to producing a minimal SCFG rule, each decomposition point also produces a phrase pair rule with the node pair's terminal yields as the right-hand side as long as the number of tokens in the yield is less than a maximum-length threshold.

Figure 3.1 is an illustration of the method for a French–English parallel sentence. Since the French terminal *voitures* is aligned to the English terminal *cars*, for example, the French N node above *voitures* can be aligned to the English NNS node above *cars*. The resulting SCFG rule is $\text{N} :: \text{NNS} \rightarrow [\textit{voitures}] :: [\textit{cars}]$. Further up in the tree, the French NP node produces a yield of three words, all of which are aligned within the yield of the NP node on the English side (or not aligned at all). Looking down from the NP nodes, we find that the next level of node alignment occurs between N and NNS and between A and JJ. These two levels of node alignment produce the SCFG rule $\text{NP} :: \text{NP} \rightarrow [\text{les N}^1\ \text{A}^2] :: [\text{JJ}^2\ \text{NN}^1]$. Since the source-side terminal *les* is unaligned, it is passed as is into the source right-hand side of the rule.

We find a number of shortcomings in this extraction technique, however. When multiple nodes n_t equally well align to n_s according to the word alignments, the hard-coded choice of selecting the "lowest" n_t ignores real ambiguity in the resulting labeled rules. In Figure 3.1, the English JJ node above *blue* is aligned only to the French A node above *bleues*, leaving the AP node unaligned and preventing it from appearing in any extracted rule. We would prefer to preserve such ambiguity through the rule extraction stage.[1] Lavie, Parlikar, and Ambati (2008) also specify a hard constraint

[1] We note, however, a trade-off in this decision: preserving labeling ambiguity due to chains of unary rules decreases



Node Alignments:

avait	↔	had
toujours	↔	always
aimé	↔	liked
voitures	↔	cars
bleues	↔	blue
.	↔	.
V	↔	VBD
ADV	↔	RB
V	↔	VBN
N	↔	NNS
A	↔	JJ
PU	↔	PU
NP	↔	NP
NP	↔	NP
SENT	↔	S

SCFG Rules:

V :: VBD	→	[avait] :: [had]
ADV :: RB	→	[toujours] :: [always]
V :: VBN	→	[aimé] :: [liked]
N :: NNS	→	[voitures] :: [cars]
A :: JJ	→	[bleues] :: [blue]
PU :: PU	→	[.] :: [.]
NP :: NP	→	[Ma mère] :: [Mother]
NP :: NP	→	[les N^1 A^2] :: [JJ^2 NNS^1]
SENT :: S	→	[NP^1 V^2 ADV^3 V^4 NP^5 PU^6] :: [NP^1 VBD^2 RB^3 VBN^4 NP^5 PU^6]

Figure 3.1: Syntactic SCFG grammar extraction from a parsed and word-aligned sentence pair.

that prevents preterminal nodes (i.e. parts of speech) from aligning to multi-word nonterminal nodes (i.e. constituents). In the figure, this forces the first node alignment above *Ma mère* ↔ *Mother* to occur between the French NP and the English NP nonterminals, even though the English preterminal NNP also covers the same phrase. While this choice reduces the complexity of node alignment, we feel it is poorly motivated linguistically and again removes ambiguity we may wish to preserve.

3.1.2 Completed Work

As a replacement, we are in the process of re-implementing a general-purpose rule extractor that will address these and other concerns. While preserving the word alignment support criterion in the node alignment stage, our implementation allows for ambiguous node alignments in cases where multiple nodes have the same word alignment coverage. We also include the method of introducing virtual nodes of Ambati and Lavie (2008), extending it similarly with an allowance for ambiguous alignments. Virtual nodes may be inserted in either parse tree in cases where they would allow additional node alignments to be made without violating surrounding tree structure. In Figure 3.1, for example, it is possible to create a virtual node in the French tree dominating the existing VN

the rule sparsity problem (since in this example *bleues* ↔ *blue* could fill both an A :: JJ and an AP :: JJ slot), but it increases spurious ambiguity (since the phrase pair is potentially inserted into the decoding chart with both labels).

and NP nodes; it will be aligned to the English side's VP node, resulting in additional extracted rules and shortening the length of the right-hand side of the extracted SENT :: S rule. We also remove the prohibition on the inter-alignment of preterminal and nonterminal nodes.

The output of the node alignment stage can be fed directly to rule extraction, or it can be written out to text for use by other applications. (See, for example, the tree relabeling scheme of Section 3.4.) Figure 3.2 shows the text-format output after performing node alignment on the sentence of Figure 3.1. Each line of output represents one alignment. The second and third columns give the source- and target-side word spans of the nodes that make up the alignment. The fourth column specifies the extraction heuristics that support the node alignment, with codes for the source and target sides separated by the digit "2." The notation "S" means that the alignment is supported by only the string-based non-syntactic phrase pair extraction of statistical MT (Koehn, Och, and Marcu, 2003), "T" indicates an exact node alignment according to the baseline method of Section 3.1.1, and "TS" stands for an alignment using virtual nodes as described above.

```
1       7-7     4-4     T2T T2TS TS2T T2S S2T
1       6-7     4-5     TS2T S2T
1       5-7     4-5     T2T T2TS TS2T T2S S2T
1       4-7     3-5     S2T
1       2-7     1-5     TS2T S2T
1       0-1     0-0     T2T T2TS TS2T T2S S2T
1       8-8     6-6     T2T T2TS TS2T T2S S2T
1       2-2     1-1     T2T T2TS TS2T T2S S2T
1       5-6     5-5     TS2T S2T
1       6-6     5-5     T2T T2TS TS2T T2S S2T
1       4-4     3-3     T2T T2TS TS2T T2S S2T
1       4-5     3-3     S2T
1       3-3     2-2     T2T T2TS TS2T T2S S2T
1       2-4     1-3     T2S
1       0-8     0-6     T2T T2TS TS2T T2S S2T
```

Figure 3.2: Text representation for the node alignments found in the sentence of Figure 3.1 using the general-purpose rule extractor under development.

Thus, while the baseline technique only extracts nine node alignments from the sentence in Figure 3.1 (those marked "T2T" in Figure 3.2), the generalized node aligner finds a total of 15 aligned spans and 18 aligned nodes.

3.1.3 Proposed Work

In the rule extraction phase, we plan to broaden single-decomposition extraction to instead use subphrase subtraction heuristics more similar to those of Hiero (Chiang, 2005). Given a node alignment as a rule's left-hand side, any subset of aligned nodes within its subtree may be chosen as right-hand side nonterminals, while the remainder of the yield is left as terminal strings. Under this extraction method, a tree pair will yield multiple valid decompositions instead of just one. The resulting grammar is similar to that extracted by the GHKM "composed rules" of Galley et al. (2006), but to arbitrary depth and with both source- and target-side syntax. As in Hiero and

GHKM extraction, we expect to counteract the explosion in computational complexity and number of rule instances with appropriate controls on maximum rule size, maximum subtree size, etc.

The more complete rule extractor described here will serve as the source of extracted grammars for the grammar relabeling techniques described in the next sections.

3.2 Label Collapsing via Alignment Distribution

3.2.1 Motivation

We introduced spurious ambiguity and rule sparsity in Chapter 2 as two of the problems that must be taken into account when developing an SCFG labeling scheme. At a certain level, both can be thought of as problems of large label sets — such as those introduced in Section 3.1 where nonterminals are composed of pairs of labels from source and target parse trees — compounded by errors in parsing and word alignment. Rule sparsity can be made worse when exactly the right sequence of labels must exist from among many possibilities, and spurious ambiguity is exacerbated when ambiguous translations or patterns are instantiated by many variant labelings. Therefore, we hypothesize that a method of automatically clustering and collapsing large label sets may prove useful in reducing these problems' negative effects.

Large labels sets may exist in part because different monolingual parsers may be modeling different granularities of syntactic categories, but not all distinctions may be needed for a particular bilingual translation scenerio. We provided earlier, at the beginning of Chapter 2, the example of adjective tags in French and English according to treebank guidelines for those two languages. In French, a single tag (A) is used, but English parsers typically separate basic adjectives (JJ) from comparatives (JJR) and superlatives (JJS). A logical question for French–English translation is whether or not the different English labels encode some sort of behavioral difference in translation as well, or whether the three subtypes merely add spurious ambiguity among rules involving adjectives.

Our initial insight is that statistical word alignments, and the node alignments derived from them, provide a source of information about behavioral differences of syntactic categories in translation. Much as an English–French bilingual speaker could conclude that English base adjectives are likely to translate as French base adjectives (*the large car* $\leftrightarrow$ *la grande voiture*), while English comparatives and superlatives require French adjective phrases (*the larger car* $\leftrightarrow$ *la plus grande voiture*; *the largest car* $\leftrightarrow$ *la voiture la plus grande*), we expect to find the same sort of information encoded in word or node alignments. In this case, we would expect to see many instances of JJ nodes aligned to A nodes, many instances of JJR or JJS nodes aligned to AP nodes, and few instances of JJR or JJS nodes aligned to A nodes. We can conclude from such distributions that, among English labels, JJR and JJS behave similarly in translation and that they behave differently from JJ.

3.2.2 Label Collapsing Algorithm

We begin with an initial set of SCFG rules extracted from a parallel parsed corpus, where S denotes the set of labels used on the source side and T denotes the set of labels used on the target side. Each rule has a left-hand side of the form $s :: t$, where $s \in S$ and $t \in T$, meaning that a parse node labeled s was aligned to a node labeled t in a parallel sentence. From the left-hand sides of all extracted rule instances, we compute label alignment distributions $P(s \mid t)$ and $P(t \mid s)$ by simple

counting and normalizing:

$$P(s \mid t) = \frac{\#(s :: t)}{\#(t)} \tag{3.1}$$

$$P(t \mid s) = \frac{\#(s :: t)}{\#(s)} \tag{3.2}$$

For two source-language labels s_1 and s_2, we have an equally simple metric of alignment distribution difference d: the total of the absolute differences in likelihood for each target-language label. The distribution difference can be defined analogously for target-side labels t_1 and t_2.

$$d(s_1, s_2) = \sum_{t \in T} |P(t \mid s_1) - P(t \mid s_2)| \tag{3.3}$$

$$d(t_1, t_2) = \sum_{s \in S} |P(s \mid t_1) - P(s \mid t_2)| \tag{3.4}$$

If s_1 and s_2 are plotted as points in $|T|$-dimensional space such that each point's position in dimension t is equal to $P(t \mid s)$, then this metric is equivalent to the L_1 distance between s_1 and s_2. An equivalent interpretation exists for t_1 and t_2. In either case, the resulting value between 0 and 2 provides a metric for the "closeness" of any two labels from one language based on what labels in the other language they are paired with in rules.

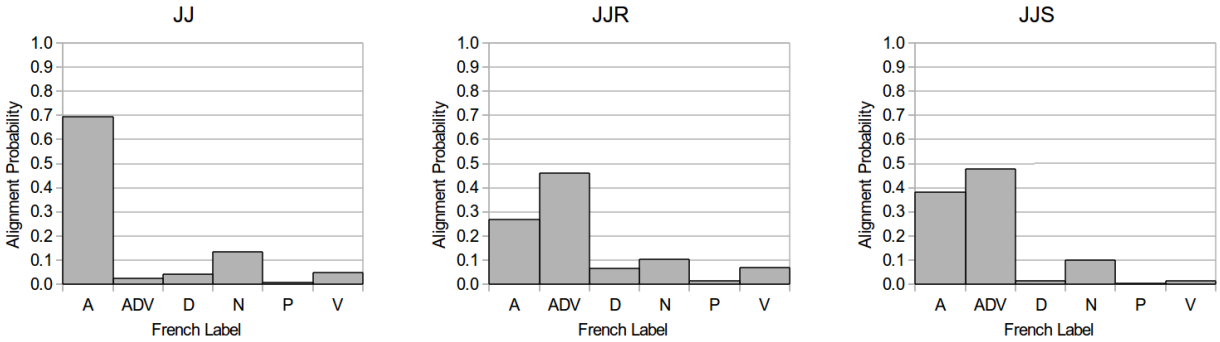


Figure 3.3: Partial alignment distributions for English adjective labels JJ, JJR, and JJS in a French–English parsed parallel corpus.

Figure 3.3 shows partial alignment distributions $P(s \mid \text{JJ})$, $P(s \mid \text{JJR})$, and $P(s \mid \text{JJS})$ for the three English adjective labels in the WMT 2010 French–English parallel parsed corpus of Section 1.3. It is clear from the figure that all three labels are somewhat related in terms of what their equivalents are in French, but it is also clear that JJR and JJS are much more closely related to each other than either is to JJ. Using Equation 3.4 to calculate alignment distribution differences, we find that $d(\text{JJR}, \text{JJS}) = 0.2688$, while $d(\text{JJ}, \text{JJR}) = 0.9952$ and $d(\text{JJ}, \text{JJS}) = 0.9114$. These results all tally with the intuitive conclusions expressed at the end of Section 3.2.1.

Given the above method for computing an alignment distribution difference for any pair of labels, we develop an iterative greedy method for label collapsing. At each step, we compute d for all pairs of labels, then collapse the pair with the smallest d value into a single label. Then alignment distribution differences are recomputed over the new, smaller label set, and again the

label pair with the smallest distribution difference is collapsed. This process continues until some stopping criterion is reached, or in the limit until all labels have been combined into one. Label pairs being considered for collapsing may be only source-side labels, only target-side labels, or both. In practice, we generally choose to allow label collapsing to apply on either side during each iteration of our greedy algorithm — whichever label pair is closest according to either Equation 3.3 or Equation 3.4 at each step will be merged.

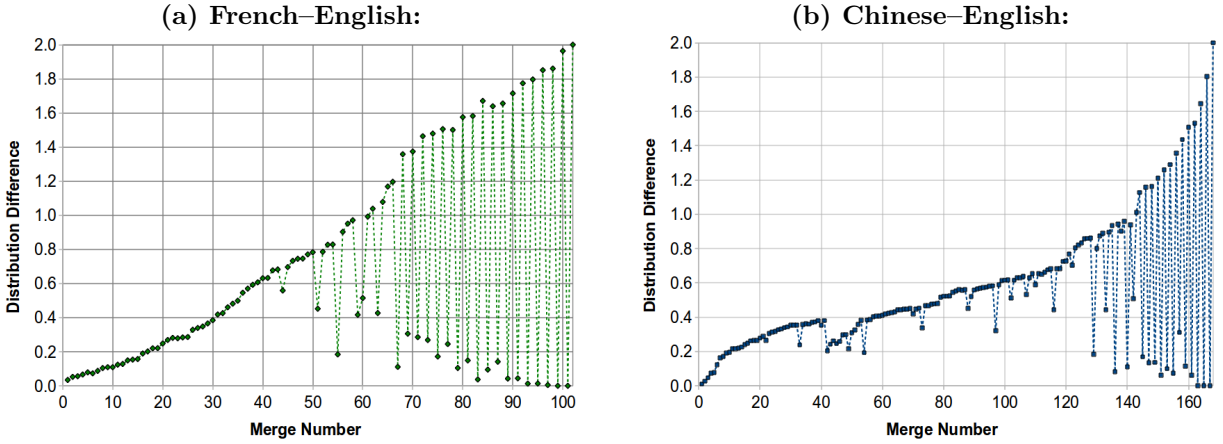


Figure 3.4: Distribution difference values for each merge in complete runs of our algorithm on French–English and Chinese–English SCFGs.

We observe an interesting behavior of the algorithm when looking for a choice of stopping criterion. Figure 3.4(a) shows a complete run of the label collapsing algorithm for the WMT 2010 French–English SCFG until all labels on each side have been collapsed to a single category. For each of the 102 merges, the graph shows the distribution difference value of the two labels that participated in the merge. These values are very close to monotonically increasing for the first 43 merges, but after that point the distribution difference of successive merges quickly falls into a see-saw pattern, alternating between very high and very low values. Alternations are merely the result of a label pair in one language suddenly scoring much lower on the distribution difference metric than previously, thanks to some change that has occurred in the label set of the other language. Figure 3.4(b), on the other hand, shows a complete run of the algorithm for a Chinese–English SCFG extracted from the FBIS corpus. In this case, large drops in distribution difference values are more prevalent throughout the entire set of merges, but another prominent feature of the graph is a significant discontinuity in the curve after the 41st merge. This represents an even larger type of see-saw, in which multiple label pairs suddenly appear closer together according to alignment distribution difference.

Although such sudden drops in distribution difference value are expected, they may provide an indication of when the label collapsing algorithm has progressed too far, since we have so reduced the label set in one language that categories previously very different have become almost indistinguishable. We thus propose the beginning of unstable behavior or the first large discontinuity, as seen in graphs like those in Figure 3.4, as a reasonable stopping point for our algorithm. At present this represents a reasonable, though manually chosen, heuristic; our proposed work in Section 3.2.4 will explore other alternatives.

3.2.3 Experiments

We ran initial label collapsing experiments on 8.6 million parsed sentence pairs of the French–English WMT 2010 data introduced in Section 1.3; the baseline system was built similarly to the baseline system described in that section and by Hanneman, Clark, and Lavie (2010). The results show a significant increase in overall translation quality. They also lead to a grammar that suffers less from spurious ambiguity and rule sparsity in two key areas.

French:		**English:**	
C-A	A, PREF, X	C-QUO	“, ”
C-I	ET, I	C-SYM	#, SYM
C-P	P, PC, S	C-N	$, NN, NNS, NNP, NNPS
C-S	SENT, Sint	C-DT	DT, PRP$
C-Srel	Srel, Ssub	C-P	IN, TO
C-VP	VPinf, VPpart	C-JJ	JJ, PDT
		C-JJX	JJR, JJS
		C-UH	LS, UH
		C-V	MD, VB, VBD, VBN, VBP, VBZ
		C-RB	RB, RBR, RBS
		C-WDT	WDT, WP, WP$
		C-ADJP	ADJP, UCP
		C-S	FRAG, S, SBARQ, SINV, SQ
		C-INTJ	INTJ, PRT
		C-NP	LST, NAC, NP, NX, QP, WHADVP, WHNP
		C-PP	PP, WHPP
		C-RRC	PRN, RRC, WHADJP

Figure 3.5: French and English category labels that are clustered together and collapsed according to our algorithm. Names of collapsed labels (starting with “C-”) are manually assigned for clarity.

A grammar extracted from the original parse trees according to the baseline grammar extraction process of Lavie, Parlikar, and Ambati (2008) uses 33 category labels on the French side and 72 on the English side, for a total of 1134 unique joint lables of the form $s :: t$. Label collapsing reduces this to 25 French, 37 English, and a new total of 502 joint labels, a reduction of 56 percent. The new coarser labels in each language represent collapsed subtypes of the original labels, partially collapsed subtypes, or new latent categories that were discovered automatically based on similar alignment distributions. A full list of the collapsed labels is given in Figure 3.5. Full collapsing of subtypes is seen in French, for example, with the combination of the VPinf and VPpart categories for verb phrases into the single category C-VP. As hinted earlier, the English adjective labels are partially collapsed: JJR and JJS are combined into one, but JJ is left distinct. A latent category is also discovered that combines the English labels PRN (parenthetical expressions), RRC (reduced relative clauses), and WHADJP (adjective phrases headed by a *wh*-adjective).

For spurious ambiguity, the size of the entire grammar is reduced by approximately 1 percent, but we find a more substantial effect on fully abstract grammar rules and on very frequent terminal phrase pairs. The 1000 most frequent phrase pairs in the grammar, for example, have 21 percent fewer possible left-hand-side labels than before label collapsing. The number of left-hand-side labels for the 1000 most frequent hierarchical grammar rules is decreased by 10 percent. We also find a 20 percent reduction in the number of unique fully abstract grammar rules.

Monotonic Patterns		Reordering Patterns	
$[X^1\ X^2] :: [X^1\ X^2]$	13.7	$[X^1\ X^2\ X^3] :: [X^1\ X^3\ X^2]$	6.0
$[X^1] :: [X^1]$	1.3	$[X^1\ X^2] :: [X^2\ X^1]$	2.8
$[X^1\ X^2\ X^3] :: [X^1\ X^2\ X^3]$	7.0	$[X^1\ X^2\ X^3\ X^4] :: [X^1\ X^3\ X^2\ X^4]$	15.7
$[w\ X^1] :: [w\ X^1]$	1.3	$[w\ X^1] :: [X^1\ w]$	1.3
$[X^1\ X^2\ X^3\ X^4] :: [X^1\ X^2\ X^3\ X^4]$	17.0	$[w\ X^1\ X^2] :: [X^2\ X^1]$	3.0

Figure 3.6: Label collapsing increases by 1.3 to 17 times the likelihood that a randomly drawn label sequence will fit the most commonly extracted monotonic and reordering rule patterns.

Rule sparsity is more difficult to measure directly without gold-standard rule applications or constrained decoding on a particular test set, but we can indirectly determine improvement by examining reordering performance. On two test sets, the number of reordering rules applied during decoding increases by 5.6 and 9.3 percent. Underlying this is a significant increase in the number of label sequences that instantiate basic reordering patterns in the grammar. Figure 3.6 shows the five most frequently extracted monotonic and reordered right-hand-side patterns in the grammar, along with how much more likely it is that a randomly drawn label sequence for each one will instantiate an actually extracted rule.[2] For example, a random label sequence is 2.8 times as likely to create a valid rule of the form $X :: X \rightarrow [X^1\ X^2] :: [X^2\ X^1]$ after label collapsing than before it.

	news-test2009			news-test2010		
System	**METEOR**	**BLEU**	**TER**	**METEOR**	**BLEU**	**TER**
Baseline	51.96	21.75	59.44	53.13	22.03	58.00
Label collapsing	53.00	22.78	59.16	54.03	23.14	57.86

Figure 3.7: French–English automatic metric results with and without a label collapsing step.

Finally, we observe significant improvements on automatic metric scores on two WMT test sets after label collapsing. We test versions of our system with and without a label collapsing step on the "news-test2009" and "news-test2010" sets, previously used as the official evaluation sets of WMT 2009 and WMT 2010. Among evaluation metrics, we perform case-insensitive scoring on BLEU (Papineni et al., 2002) as implemented by the NIST `mteval-v13.pl` script, METEOR v. 1.0 (Lavie and Denkowski, 2009), and TER v. 0.7 (Snover et al., 2006). As shown in Figure 3.7, label collapsing improves scores across metrics on both test sets. The smaller change in TER could be explained by a change in length between the baseline and collapsed system — the metric notoriously prefers shorter translations, and the output from the label-collapsed system is slightly longer than the baseline.

3.2.4 Proposed Work

We propose a number of natural extensions to the label collapsing setup described so far.

First of all, we plan to investigate the heuristic stopping criterion currently in use based on see-saws or large discontinuities in distribution difference values. Though the current criterion

[2] X represents any nonterminal, and w represents the appearance of one or more terminal words.

can be calculated automatically, it is unclear whether it represents the most meaningful method of determining when label collapsing has progressed too far. At the cost of a more expensive calculation, a more refined stopping criterion could take into account some property of the collapsed grammar we wish to improve — the change in conditional entropy of $s :: t$ for a given s_i or t_i, for example, could be used to monitor when label collapsing has flattened the alignment distribution for individual labels. Monitoring the count of SCFG left-hand-side labels for a given right-hand side as the algorithm progresses could lead to a stopping criterion based on improvements in spurious ambiguity.

The Berkeley statistical parser for English, French, German, Chinese, Arabic, and Bulgarian (Petrov and Klein, 2007) is trained to produce two different label sets for each language: the default label set as learned directly from each language's treebank, or a fragmented label set following the label splitting algorithm of Petrov et al. (2006). (See Section 1.4.) The fragmented label sets provide an intriguing alternative starting point for the label collapsing algorithm described in this section, as the monolingual fragmented grammars typically contain hundreds instead of tens of labels each. We propose to carry out label collapsing from such an extreme baseline in order to test the range of our algorithm. Our hypothesis is that the ability to cluster and collapse finer-grained subtypes will result in a significantly different and more customized label set for a given language pair than either the default treebank labels or our initial label collapsing results.

We also propose to explore different formulations of the alignment distribution different metric and the exact procedure for collapsing. The L_1 distance metric in Equations 3.3 and 3.4 combined with the greedy collapsing algorithm of Section 3.2.2 reflects a simple and intuitive approach, but it may not be optimal. In particular, the choice of L_1 distance can be contrasted with the use of Euclidean (L_2) distance or KL divergence (Kullback and Leibler, 1951). Greedy collapsing of both source and target labels at once can be contrasted with a clustering algorithm such as K-means, where either the source- or target-side labels can be fully clustered in one step.

All of the above extensions are straightforward modifications of the baseline algorithm from Section 3.2.2. They can be carried out experimentally one at a time by swapping out the current stopping criterion, input grammar, or clustering mechanism, and the results can be analyzed and compared to the baseline as we did in Section 3.2.3. We will design more complicated experiments based on the success or non-success of the experiments already proposed here. Finally, we will consider the introduction of new features beyond alignment distribution, as warranted by analysis of the strengths and weaknesses of the current approach. The perceptron approach of Fossum, Knight, and Abney (2008), applied in their case to the problem of word-alignment modification, can serve as a possible inspiration for a simple multi-feature framework.

3.3 Label Refining via Right-Hand-Side Context

3.3.1 Motivation

Label sets developed monolingually for non-MT tasks may not reflect a large amount of potentially useful information that could be modeled in a bilingual MT framework. For instance, a monolingual treebank for English is extremely unlikely to provide two classes of adjectives for words that do or do not preceed their head nouns when they are translated into French. A monolingual French treebank, perhaps considering the pre-noun vs. post-noun distinction unimportant or obvious, may

not mark it either.[3] A syntax-based MT system trained from monolingual English and French parsers will thus end up missing information with respect to adjective positions. With the same set of adjective tags appearing both pre- and post-noun, the grammar cannot encode the fact that certain English adjectives are very likely to be re-ordered in French while others are not. Making adjective positioning information explicit and including it in an MT system for English–French translation would allow for a more precise and less ambiguous translation model. Instead of a single adjective class whose members all "sometimes" change position in French output, words can be explicitly assigned with individual probabilities to "in-order" and "reorder" adjective types that encapsulate finer-grained information about the behavior of each word.

SCFG Rule:	p_{trans}**:**	p_{label}**:**
(1) DT :: D → [the] :: [la]	0.3309	0.9980
(2) NN :: N → [car] :: [voiture]	0.4003	0.9679
(3) JJ :: A → [large] :: [grande]	0.1165	0.9968
(4) JJ :: A → [blue] :: [bleue]	0.2716	0.8864
(5) NP :: NP → [DT^1 JJ^2 NN^3] :: [D^1 N^3 A^2]	0.5622	0.9974
(6) NP :: NP → [DT^1 JJ^2 NN^3] :: [D^1 A^2 N^3]	0.2247	0.9999

Figure 3.8: SCFG rules and simplified model scores for translating the English noun phrases *the large car* and *the blue car* into French.

As an illustration of this, we consider the two English noun phrases *the large car* and *the blue car*, which we would like to translate into the French NPs *la grande voiture* and *la voiture bleue*, respectively. Note the difference in position of the adjective in French. Producing the target output requires the SCFG rules presented in Figure 3.8, which also includes their rule scores as calculated from a baseline grammar extraction using the WMT 2010 data of Section 1.3. For simplicity, we reduce the translation model to two features for a generic rule of the form $\ell_s :: \ell_t \rightarrow [r_s] :: [r_t]$: the translation probability $p_{trans} = P(r_t \,|\, r_s)$ and the label probability $p_{label} = P(\ell_s, \ell_t \,|\, r_s, r_t)$. Once the appropriate lexical rules 1–4 have been applied to the English input, the model has a choice between Rule 5 and Rule 6 for both of our proposed noun phrases. Their label probabilities are very similar, but the reordered translation going into French in Rule 5 is two and a half times more likely than the in-order version of Rule 6 according to their translation probabilities. Thus, this translation model will have a strong preference for producing both *la voiture bleue*, which is correct, and *la voiture grande*, which is not.

Although it is quite correct that Rule 5 predominates over Rule 6 for English–French adjective translation as a whole, the class of French adjectives can actually be decomposed into two subgroups: a majority group of those that almost always follow their head noun, and a minority group of those that almost always precede it. Given the lexical identity of any adjective, the correct choice between Rule 5 and Rule 6 is more obvious.

[3] As we have seen, the Paris 7 Treebank in fact does not mark the distinction, using various adjective subtype tags to indicate gender, number, etc., but not position. Further, the Berkeley and Stanford French parsers use the single tag A for all adjectives.

3.3.2 Proposed Work

We propose using the bilingual SCFG setting to detect hidden subtypes of syntactic categories that are not expressed monolingually by treebanks or parsers. Our intended technique begins by considering sets of rules with the same left-hand side and source right-hand side, but differing target right-hand sides. Rules 5 and 6 from Figure 3.8 are such a set for English–French. Rules 3.5 and 3.6 below are examples for French–English.

$$\text{NP} :: \text{NP} \rightarrow [\text{N}^1\ \text{A}^2\ \text{A}^3] :: [\text{JJ}^2\ \text{JJ}^3\ \text{NN}^1] \tag{3.5}$$

$$\text{NP} :: \text{NP} \rightarrow [\text{N}^1\ \text{A}^2\ \text{A}^3] :: [\text{JJ}^3\ \text{JJ}^2\ \text{NN}^1] \tag{3.6}$$

From such a set, we suppose that a hidden subtype for one of the reordered categories might explain the reordering. One source of evidence for a hidden preterminal category is an underlying lexical difference, as we saw in the motivating example in Section 3.3.1. For a right-hand-side nonterminal that changes position, it may be because of an underlying derivational difference one level lower in the tree. We propose to search for both types of differences by returning to the training corpus and tabulating, for each extracted instance of a rule in our set of interest, the lexical identities or the derivation of each constituent plugging into it. Items much more likely to fit one rule in the set than another form a category subtype, and their labels will be re-written throughout the training corpus accordingly.

$\mathbf{D}^1$	**Rule 5**	**Rule 6**	$\mathbf{N}^3$	**Rule 5**	**Rule 6**	$\mathbf{A}^2$	**Rule 5**	**Rule 6**
un	0.3165	0.2067	rôle	0.0349	0.0023	commune	0.0338	—
une	0.2654	0.2545	position	0.0241	0.0017	important	0.0286	0.0078
la	0.1228	0.1688	question	0.0213	0.0199	politique	0.0285	—
le	0.0815	0.1054	communauté	0.0212	0.0002	internationale	0.0267	—
cette	0.0428	0.0563	solution	0.0172	0.0064	européenne	0.0218	—
			fois	0.0001	0.0397	même	0.0004	0.0957
			temps	0.0004	0.0355	nouvelle	0.0018	0.0837
			point	0.0065	0.0258	première	0.0003	0.0676
			majorité	0.0025	0.0212	bonne	—	0.0593
						nouveau	0.0012	0.0521

Figure 3.9: The most likely French words to plug into each right-hand-side element of Rule 5 (French side $[\text{D}^1\ \text{N}^3\ \text{A}^2]$) and Rule 6 (French side $[\text{D}^1\ \text{A}^2\ \text{N}^3]$) from Figure 3.8.

Let us assume, for example, that our proposed system has identified Rules 5 and 6 from Figure 3.8 as a rule set of interest. Figure 3.9 shows, from a sample of nearly 47,000 applications of those rules, the five most likely French words to plug into each slot of each rule. (That is, 31.65 percent of the D^1 slots in instances of Rule 5 are filled by the word *un*, while 20.67 percent of the D^1 slots in Rule 6 are.) The top five words used as the determiner in Rule 5 are also the top five words used as the determiner in Rule 6; the probabilities are somewhat perturbed, but the overall ranking is almost identical. The situation at the noun (N) node is more clouded. The word *question* is among the top five possibilities for both rules, but otherwise Ns that are more likely in one rule are rather less likely in the other, both in terms of probability and overall ranking. The biggest difference between rules, however, is shown in the adjective (A) slot. Of the five most likely adjectives to be involved in Rule 5, only one of them (*important*) appears at all in Rule 6 — and then with a very

low probability. Adjectives that are the most likely in Rule 6, in turn, appear in Rule 5 with either very low probabilities or not at all (*bonne*).

We can conclude from all this that it is most likely a latent subcategory of adjectives that gives rise to the varient target-side orderings seen in Rules 5 and 6. The words in the A^3 section of Figure 3.9 now become evidence for the new syntactic category, and their labels are reassigned. In every rule extraction instance where one of these words plugs into Rule 6, the label JJ :: A is modified to, say, JJ-A :: A-A. This establishes in the grammar new rules

$$\text{JJ-A :: A-A} \rightarrow [\text{same}] :: [\text{même}],\ \textit{etc.} \tag{3.7}$$

$$\text{NP :: NP} \rightarrow [\text{DT}^1\ \text{JJ-A}^2\ \text{NN}^3] :: [\text{D}^1\ \text{A-A}^2\ \text{N}^3] \tag{3.8}$$

with high label and translation probabilities in the case of the hierarchical rule. At the risk of generating some additional spurious ambiguity and rule sparsity, we posit that we may gain a larger increase in reordering precision.

Further exploratory research will be required in order to determine the scope of this technique appropriately, such as by the inclusion of cutoffs on the number of rule sets considered, the strength of evidence required to create a new category label, or the maximum number of new categories that should be introduced overall. The setup of Figure 3.9 with comparison of probability values could point towards the use of a distribution difference metric, as in Section 3.2, to judge the strength of evidence in a rule set for a hidden category.

3.4 EM-Based Tree Relabeling

3.4.1 Motivation

Variant labelings in an SCFG rule are in part caused by labeling errors in the monolingual statistical parsers that originally parsed each side of the parallel corpus from which rules are extracted. Each sentence in the corpus is parsed independently, both with respect to other sentences in the same language and with respect to the particular foreign-language sentence aligned to it. It is thus possible that a number of local labeling errors could be corrected if global information were taken into account.

We consider the treatment of cardinal numbers as a simple example. From the English Penn Treebank tagging guidelines, it is clear that any terminal made up of purely digits should be labeled a CD (Santorini and MacIntyre, 1995). This is the result for 96.4 percent of the all-digit terminals in 8.6 million English sentences that have been parsed with the Penn Treebank-trained Berkeley parser. However, the remaining 3.6 percent are labeled with 22 different incorrect tags, ranging from nouns and adjectives to prepositions and verbs. While one of these other labels may lead to a higher parse-model score in any one sentences, the evidence from the overall corpus strongly indicates that they are erroneous.

Cases of single-tag errors can be resolved with simple preprocessing — in this case, to blindly overwrite any tag for an all-digit terminal with CD — but a more difficult scenario awaits in French. The Paris 7 French Treebank tagging guide specifies that numbers may be tagged as D, N, A, or PRO depending on how they are used in the sentence (Abeillé and Clément, 2003), yet errors persist in scattered instances. We can again use the overwhelming weight of correctly tagged instances to fix individual errors given their contexts.

3.4.2 Proposed Work

We propose an EM algorithm (Dempster, Laird, and Rubin, 1977) to relabel a parsed parallel corpus after an initial process of node alignment has been run on it. Node alignment results in a baseline grammar with rules of the form $\ell \rightarrow r$.

Each node-aligned tree pair is then considered separately in the E step. We remove the labels from the aligned nodes, then apply rules from the current grammar bottom-up in order to reproduce the most likely labeled version of the same tree. We define a recursive formula for keeping track of the probability $\alpha(\ell_i, n)$ of having label ℓ_i at aligned node n. The probability of reaching an ℓ_i at n via a rule $R = \ell_i \rightarrow r$ is equal to the probability of having that rule's right-hand-side nonterminal labels $\ell_1 ... \ell_k$ in existence at the appropriate subtree nodes $n_1 ... n_k$ times the probability of applying the rule on top of such a right-hand side:

$$\alpha_R(\ell_i, n) = P(\ell_i \,|\, r) \prod_{j=1}^{k} \alpha(\ell_j, n_j) \tag{3.9}$$

These probabilities are accumulated for all rules R that may apply at n to produce label ℓ_i, giving the overall probability for creating ℓ_i at n by any combination of rule applications:

$$\alpha(\ell_i, n) = \sum_{R} \alpha_R(\ell_i, n) \tag{3.10}$$

Once the maximally likely labelings of all tree pairs have been computed, new SCFG rules are extracted from the corpus in the algorithm's M step to form an updated grammar for the next EM iteration. Updated rule probabilities $P(\ell \,|\, r)$ are computed from the extracted rule instances by simple maximum-likelihood estimates

$$P(\ell \,|\, r) = \frac{\#(\ell \rightarrow r)}{\#(r)} \tag{3.11}$$

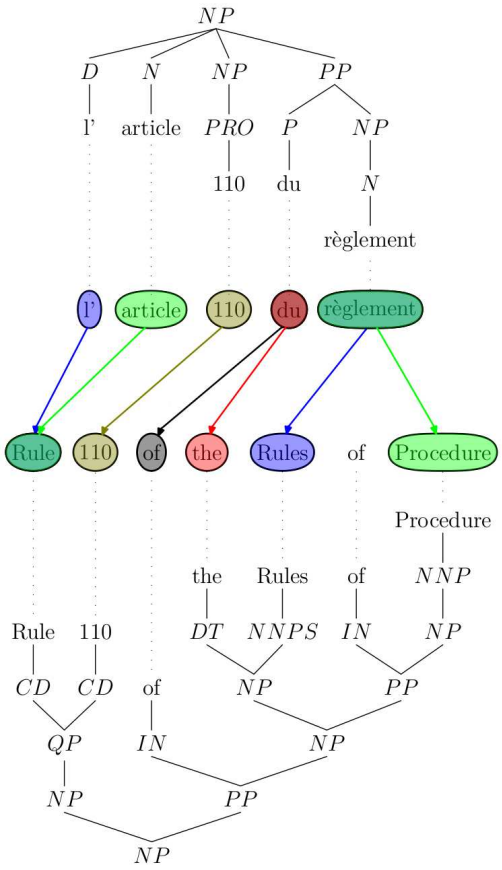
We illustrate one iteration of the algorithm using the tree fragment in Figure 3.10 and the baseline extracted grammar of Section 1.3. Figure 3.10(a) shows the original tree fragment as it appears in the corpus; however, there are only three node alignments and only three SCFG rules extracted from it. Thus, from the point of view of reproducing the tree from rules in our EM algorithm, the structure can be simplified to the one in Figure 3.10(b). At the algorithm's initialization, the rules extracted from the tree are those shown in Figure 3.10(c), where the number *110* has been incorrectly tagged in French. Node labels are removed, and we now try to reproduce the most likely labeled version of the tree according to rules extracted from the entire grammar.

Let us begin at node n_1. We find in our baseline grammar 37 possible labelings for a rule of the form X :: X $\rightarrow$ [110] :: [110]; one of them has the left-hand-side label D :: CD. There are no right-hand-side constituents in the rule, so in this case Equation 3.9 becomes

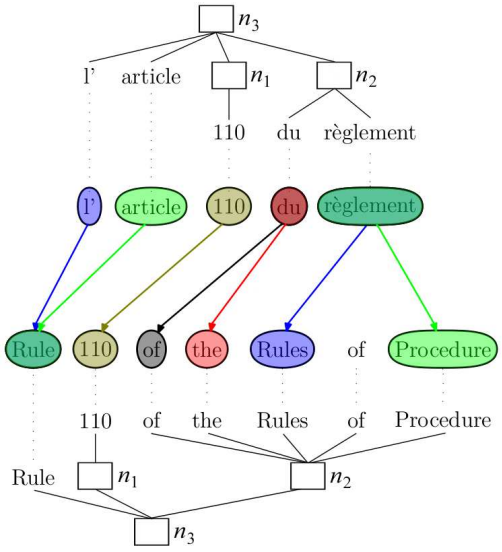
$$\alpha_{\text{D::CD}\rightarrow[110]::[110]}(\text{D :: CD}, n_1) = P(\text{D :: CD} \,|\, [110] :: [110]) = 0.5081 \tag{3.12}$$

with the label probability $P(\text{D :: CD} \,|\, [110] :: [110])$ being computed from the full extracted grammar. Similar calculations are performed for other possible labelings at n_1, resulting in a matrix of label probabilities $\alpha(\text{D :: CD}, n_1) = 0.5081$, $\alpha(\text{PRO :: CD}, n_1) = 0.2789$, $\alpha(\text{N :: CD}, n_1) = 0.1004$, and so on.

(a) Original Tree



(b) Reproduced Tree



(c) Original Extracted Rules

PRO :: CD → [110] :: [110]
PP :: PP → [du règlement] :: [of the Rules of Procedure]
NP :: NP → [l' article PRO^1 PP^2] :: [Rule CD^1 PP^2]

(d) Second Iteration Rules Extracted

N :: CD → [110] :: [110]
PP :: PP → [du règlement] :: [of the Rules of Procedure]
NP :: NP → [l' article N^1 PP^2] :: [Rule CD^1 PP^2]

Figure 3.10: A tree fragment set up for our EM-based tree relabeling algorithm. After the algorithm's first iteration, the labeling error PRO :: CD is corrected to N :: CD.

The same line of calculations is carried out at node n_2, this time using rules of the form X :: X → [du règlement] :: [of the Rules of Procedure]. There are only two of them in the grammar, leading to a second column of the label probability matrix that contains $\alpha(\text{PP :: PP}, n_2) = 0.7736$ and $\alpha(\text{NP :: PP}, n_2) = 0.2264$.

At node n_3, we are looking for rules of the form X :: X → [l' article X^1 X^2] :: [Rule X^1 X^2], of which there are 13 in the grammar. One of them, NP :: NP → [l' article D^1 PP^2] :: [Rule CD^1 PP^2], requires right-hand-side constituents labeled D :: CD and PP :: PP. Thus, for this rule, Equation 3.9 becomes

$$\begin{aligned}&\alpha_{\text{NP::NP}\rightarrow\text{[l' article D}^1\text{ PP}^2\text{]::[Rule CD}^1\text{ PP}^2\text{]}}(\text{NP :: NP}, n_3)\\ &\quad = P(\text{NP :: NP} \mid \text{[l' article D}^1\text{ PP}^2\text{]} :: \text{[Rule CD}^1\text{ PP}^2\text{]}) \cdot \alpha(\text{D :: CD}, n_1) \cdot \alpha(\text{PP :: PP}, n_2)\\ &\quad = 0.0080 \cdot 0.5081 \cdot 0.7736 = 0.0031 \qquad (3.13)\end{aligned}$$

Eleven of the 13 rules that could apply at n_3 have NP :: NP as their left-hand sides, so their

probabilities calculated as above are all summed together according to Equation 3.10 to produce the total value for $\alpha(\text{NP} :: \text{NP}, n_3)$ in the label probability matrix. The remaining two rules are headed by NP :: VP; they are summed together to create $\alpha(\text{NP} :: \text{VP}, n_3)$. At the end of the procedure, the two label probabilities for the top of the tree are $\alpha(\text{NP} :: \text{NP}, n_3) = 0.0343$ and $\alpha(\text{NP} :: \text{VP}, n_3) = 0.0059$.

To find the Viterbi labeling of the entire tree, we now trace downward from the most likely label at the root. As we just saw, the most likely label at n_3 is NP :: NP. Within the rules for generating an NP :: NP at that node, the most likely is NP :: NP $\rightarrow$ [l' article N^1 PP^2] :: [Rule CD^1 PP^2]. Thus, we assign label N :: CD to n_1 and label PP :: PP to n_2. When SCFG rules are extracted from this tree fragment in the EM algorithm's M step, the rule previously extracted as PRO :: CD $\rightarrow$ [110] :: [110] will instead become N :: CD $\rightarrow$ [110] :: [110], as in Figure 3.10(d). The local labeling error has been fixed.

3.5 Combining Relabeling Techniques

So far we have described three individual methods for automatically relabeling parse trees or extracted SCFG rules. However, aside from testing each method individually, our goal in Section 1.2 was to create one overall scheme to improve the label set for any language pair or choice of automatic parsers. In this section, we specify that that will be accomplished by a pipeline combining one or more of the techniques from the last three sections in sequence, and we propose specific sequences that make sense under certain conditions.

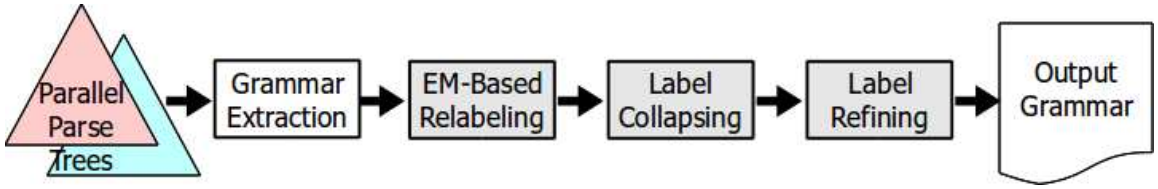


Figure 3.11: A proposed default combination of relabeling techniques for large-data scenarios. Steps representing relabeling techniques are shaded in grey.

The data resources and experimental setup we outlined in Section 1.3 are geared towards large-scale systems. In such cases, where the extracted grammars should be large enough to permit good statistical modeling, we propose a final sequence made up of all three relabeling techniques, shown in Figure 3.11. After running the more generalized method of grammar extraction from Section 3.1, we propose first using the EM-based tree relabeling technique of Section 3.4. Its goal is to remove occasional monolingual labeling errors by making use of the entire extracted bilingual grammar, and it is suitable to remove such easily correctable sources of spurious ambiguity right away. Once some variant labelings have been cleared away, the label collapsing procedure of Section 3.2 is a logical second step. Its goal is to further reduce spurious ambiguity and rule sparsity by removing monolingual labels that are in a sense redundant because they have the same alignment behavior as other labels. This should result in an improved, smaller set of labels, but so far little attention has been paid to bilingual categories. Thus, we propose a final step of label refining (Section 3.3) to re-enlarge the label set with useful categories that are motivated by the particular bilingual MT task being modeled. The output of this third step will be the final improved grammar.

Further experimentation will be necessary to determine whether this pipeline should be modified in the event that the input parse trees use very fine-grained categories, such as those returned by the refinement technique of Petrov et al. (2006). With such detailed labels, it is possible that the rule and label probabilities in EM-based relabeling will become overly fragmented. It is also possible that some latent bilingual categories will already be modeled in the fine-grained joint labels, since monolingual labels will already be distinguishing between a large variety of category subtypes. In these cases, better final results might be obtained by moving the label collapsing step forward to the beginning of the pipeline, or by removing one of the other two relabeing steps.

Section 1.3's prototype Arabic–English and Chinese–English systems may also require a different combination of relabeling techniques. With a much smaller training corpus, it may be desirable to have an overall smaller label set in order to avoid fragmenting rule scores and translation probabilities across many variant labeling. There is also a benefit to fixing label errors for the same reason. These two facts would put the emphasis on label collapsing and EM-based relabeling among our proposed techniques, possibly at the expense of including the label refining module.

Chapter 4

Summary and Timeline

4.1 Summary

We defined and illustrated three properties of an SCFG that are highly dependent on the grammar's label set:

- **Spurious ambiguity.** Intuitively the case of having "too many labels," spurious ambiguity occurs when a given right-hand side can be labeled with a large number of left-hand sides, or when a reordering pattern can be instantiated by a large number of right-hand-side labels. It leads to weaker model scores for outputs that can be formed in multiple ways.
- **Rule sparsity.** The case of having "too few rules," rule sparsity is the inability of a grammar rule to apply because the nonterminal labels on its right-hand side do not match the labels of the constituents actually found at the desired application sites. It restricts the space of possible derivations that can be created at run time, which in turn restricts the space of possible output translations.
- **Reordering precision.** Reordering precision defines the spectrum between extreme rule sparsity on one hand and an overly permissive (or overly ambiguous) grammar on the other. It encapsulates the notion that certain reorderings are permissible only in certain situations by forcing the constituents that plug into a rule to match the categories in the rule's right-hand side. The granularity of the categories controls the degree of precision.

We sketched out ways to evaluate a grammar based on these characterisitcs, which will be used — along with additional criteria, such as automatic MT metrics and decoding run time — to judge improvements from a series of proposed automatic SCFG relabeling techniques. These relabeling techniques will be carried out on SCFGs extracted by a new general-purpose rule extractor that incorporates a variety of existing grammar extraction methods.

- **Label collapsing.** For category labels in one language, we use their distributions over aligned categories in the other language to cluster and collapse monolingual labels according to similar alignment distribution. We will show that this technique reduces both spurious ambiguity and rule sparsity by removing redundant labels and allowing important reordering patterns to be instantiated by a larger fraction of labelings.

- **Label refining.** We use sets of rules with the same left-hand side and source right-hand side, but different orderings on the target right-hand side, as evidence for a hidden subtype in one of the right-hand-side labels that explains the reordering. We will develop a model, based on underlying compositional differences of the reordered constituents, for creating new subtype labels with a view to increasing the reordering precision of the grammar.

- **Tree relabeling.** We propose an EM algorithm to correct local labeling errors in individual parse trees by using the grammar extracted from the entire parallel corpus to bias towards the correct label. The expected result is a reduction in spurious ambiguity.

4.2 Timeline

Work on this thesis project should be completed by May 2012, with individual tasks being carried out according to the timeline below.

January to March 2011:
Implementation of general-purpose rule extractor.
Full label collapsing experiments.
Tests of extensions to label collapsing algorithm.
Checkpoint: Submission of EMNLP paper on label collapsing (March 23).

April to July 2011:
Development of constrained decoding setup.
Development of label refining techniques.
Full label refining experiments.
Checkpoint: Submission of French–English system to WMT 2011 (expected April–June).

August to October 2011:
Development of framework for EM-based relabeling.
Full EM-based relabeling experiments.
Checkpoint: Submission of Arabic or Chinese system to NIST 2011 (assumed).

November to December 2011:
Finalization of all experiments on individual techniques.
Begin job search.
Checkpoint: Submission of ACL paper on relabeling techniques (expected December).

January to March 2012:
Development of combined technique pipelines.
Finalization of systems and experimentation.

April to May 2012:
Thesis writing.
Checkpoint: Thesis defense (May).

References

Abeillé, Anne and Lionel Clément, 2003. *Annotation Morpho-Syntaxique: Les mots simples, les mots composés*. Université Paris 7, Paris, France. `http://www.llf.cnrs.fr/Gens/Abeille/guide-morpho-synt.02.pdf`.

Abeillé, Anne, Lionel Clément, and François Toussenel. 2003. Building a treebank for French. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers, pages 165–187.

Aho, A.V. and J.D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56.

Ambati, Vamshi and Alon Lavie. 2008. Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 235–244, Waikiki, HI, October.

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

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270, Ann Arbor, MI, June.

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

Clark, Jonathan and Alon Lavie. 2010. LoonyBin: Keeping language technologists sane through automated management of experimental (hyper)workflows. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 1301–1308, Valletta, Malta, May.

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

Fossum, Victoria, Kevin Knight, and Steven Abney. 2008. Using syntax to improve word alignment precision for syntax-based machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 44–51, Columbus, OH, June.

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

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

Gao, Qin and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, OH, June.

Garera, Nikesh, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, pages 129–137, Boulder, CO, June.

Hanneman, Greg, Jonathan Clark, and Alon Lavie. 2010. Improved features and grammar selection for syntax-based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 82–87, Uppsala, Sweden, July.

Hanneman, Greg and Alon Lavie. 2009. 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*, pages 1–9, Boulder, CO, June.

Huang, Bryant and Kevin Knight. 2006. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 240–247, New York, NY, June.

Klein, Dan and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, pages 3–10.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 48–54, Edmonton, Alberta, May–June.

Kullback, Solomon and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Lavie, Alon and Michael J. Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23(2–3):105–115.

Lavie, Alon, Alok Parlikar, and Vamshi Ambati. 2008. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation*, pages 87–95, Columbus, OH, June.

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

Li, Zhifei, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proceedings of the 47th Annual Meeting of the ACL and the Fourth IJCNLP of the AFNLP*, pages 593–601, Suntec, Singapore, August.

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

Liu, Yang, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translations with packed forests. In *Proceedings of the 47th Annual Meeting of the ACL and the Fourth IJCNLP of the AFNLP*, pages 558–566, Suntec, Singapore, August.

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

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evalution of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, July.

Pareschi, Remo and Mark Steedman. 1987. A lazy way to chart-parse with categorial grammars. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 81–88, Stanford, CA, July.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 433–440, Sydney, Australia, July.

Petrov, Slav and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411, Rochester, NY, April.

Santorini, Beatrice and Robert MacIntyre, 1995. *Part-of-Speech Tagging Guidelines for the Penn Treebank Project*. University of Pennsylvania, Philadelphia, PA. `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`.

Setiawan, Hendra and Philip Resnik. 2010. Generalizing hierarchical phrase-based translation using rules with adjacent nonterminals. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 349–352, Los Angeles, CA, June.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Seventh Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA, August.

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

Xue, Nianwen, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.

Zhechev, Ventsislav and Andy Way. 2008. Automatic generation of parallel treebanks. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1105–1112, Manchester, England, August.

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