Improving Syntax-Augmented Machine Translation by Coarsening the Label Set

Greg Hanneman and **Alon Lavie**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{ghannema, alavie}@cs.cmu.edu

Abstract

We present a new variant of the Syntax-Augmented Machine Translation (SAMT) formalism with a category-coarsening algorithm originally developed for tree-to-tree grammars. We induce bilingual labels into the SAMT grammar, use them for category coarsening, then project back to monolingual labeling as in standard SAMT. The result is a "collapsed" grammar with the same expressive power and format as the original, but many fewer nonterminal labels. We show that the smaller label set provides improved translation scores by 1.14 BLEU on two Chinese–English test sets while reducing the occurrence of sparsity and ambiguity problems common to large label sets.

1 Introduction

The formulation of statistical machine translation in terms of synchronous parsing has become both theoretically and practically successful. In a parsing-based MT formalism, synchronous context-free grammar rules that match a source-language input can be hierarchically composed to produce a corresponding target-language output. SCFG translation grammars can be extracted automatically from data. While *formally* syntactic approaches with a single grammar nonterminal have often worked well (Chiang, 2007), the desire to exploit linguistic knowledge has motivated the use of translation grammars with richer, *linguistically* syntactic nonterminal inventories (Galley et al., 2004; Liu et al., 2006; Lavie et al., 2008; Liu et al., 2009).

Linguistically syntactic MT systems can derive their label sets, either monolingually or bilingually, from parallel corpora that have been annotated with source- and/or target-side parse trees provided by a statistical parser. The MT system may exactly adopt the parser's label set or modify it in some way. Larger label sets are able to represent more precise, fine-grained categories. On the other hand, they also exacerbate a number of computational and modeling problems by increasing grammar size, derivational ambiguity, and data sparsity.

In this paper, we focus on the Syntax-Augmented MT formalism (Zollmann and Venugopal, 2006), a monolingually labeled version of Hiero that can create up to 4000 "extended" category labels based on pairs of parse nodes. We take a standard SAMT grammar with target-side labels and extend its labeling to a bilingual format (Zollmann, 2011). We then coarsen the bilingual labels following the "label collapsing" algorithm of Hanneman and Lavie (2011). This represents a novel extension of the tree-to-tree collapsing algorithm to the SAMT formalism. After removing the source-side labels, we obtain a new SAMT grammar with coarser target-side labels than the original.

Coarsened grammars provide improvement of up to 1.14 BLEU points over the baseline SAMT results on two Chinese–English test sets; they also outperform a Hiero baseline by up to 0.60 BLEU on one of the sets. Aside from improved translation quality, in analysis we find significant reductions in derivational ambiguity and rule sparsity, two problems that make large nonterminal sets difficult to work with.

Section 2 provides a survey of large syntax-based

Proceedings of NAACL-HLT 2013, pages 288–297,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

MT label sets, their associated problems of derivational ambiguity and rule sparsity, and previous attempts at addressing those problems. The section also summarizes the tree-to-tree label collapsing algorithm and the process of SAMT rule extraction. We then describe our method of label collapsing in SAMT grammars in Section 3. Experimental results are presented in Section 4 and analyzed in Section 5. Finally, Section 6 offers some conclusions and avenues for future work.

2 Background

2.1 Working with Large Label Sets

Aside from the SAMT method of grammar extraction, which we treat more fully in Section 2.3, several other lines of work have explored increasing the nonterminal set for syntax-based MT. Huang and Knight (2006), for example, augmented the standard Penn Treebank labels for English by adding lexicalization to certain types of nodes. Chiang (2010) and Zollmann (2011) worked with a bilingual extension of SAMT that used its notion of "extended categories" on both the source and target sides. Taking standard monolingual SAMT as a baseline, Baker et al. (2012) developed a tagger to augment syntactic labels with some semantically derived information. Ambati et al. (2009) extracted tree-to-tree rules with similar extensions for sibling nodes, resulting again in a large number of labels.

Extended categories allow for the extraction of a larger number of rules, increasing coverage and translation performance over systems that are limited to exact constituent matches only. However, the gains in coverage come with a corresponding increase in computational and modeling complexity due to the larger label set involved.

Derivational ambiguity — the condition of having multiple derivations for the same output string — is a particular problem for parsing-based MT systems. The same phrase pair may be represented with a large number of different syntactic labels. Further, new hierarchical rules are created by abstracting smaller phrase pairs out of larger ones; each of these substitutions must also be marked by a label of some kind. Keeping variantly labeled copies of the same rules fragments probabilities during grammar scoring and creates redundant hypotheses in the decoder at run time.

A complementary problem — when a desired rule application is impossible because its labels do not match — has been variously identified as "data sparsity," the "matching constraint," and "rule sparsity" in the grammar. It arises from the definition of SCFG rule application: in order to compose two rules, the left-hand-side label of the smaller rule must match a right-hand-side label in the larger rule it is being plugged in to. With large label sets, it becomes less likely that two arbitrarily chosen rules can compose, making the grammar less flexible for representing new sentences.

Previous research has attempted to address both of these problems in different ways. Preference grammars (Venugopal et al., 2009) are a technique for reducing derivational ambiguity by summing scores over labeled variants of the same derivation during decoding. Chiang (2010) addressed rule sparsity by introducing a soft matching constraint: the decoder may pay a learned label-pair-specific penalty for substituting a rule headed by one label into a substitution slot marked for another. Combining properties of both of the above methods, Huang et al. (2010) modeled monolingual labels as distributions over latent syntactic categories and calculated similarity scores between them for rule composition.

2.2 Label Collapsing in Tree-to-Tree Rules

Aiming to reduce both derivational ambiguity and rule sparsity, we previously presented a "label collapsing" algorithm for systems in which bilingual labels are used (Hanneman and Lavie, 2011). It coarsens the overall label set by clustering monolingual labels based on which labels they appear joined with in the other language.

The label collapsing algorithm takes as its input a set of SCFG rule instances extracted from a parallel corpus. Each time a tree-to-tree rule is extracted, its left-hand side is a label of the form s::t, where s is a label from the source-language category set S and t is a label from the target-language category set T. Operationally, the joint label means that a source-side subtree rooted at s was the translational equivalent of a target-side subtree rooted at t in a parallel sentence. Figure 1 shows several such subtrees, highlighted in grey and numbered. Joint left-hand-side labels for the collapsing algorithm,

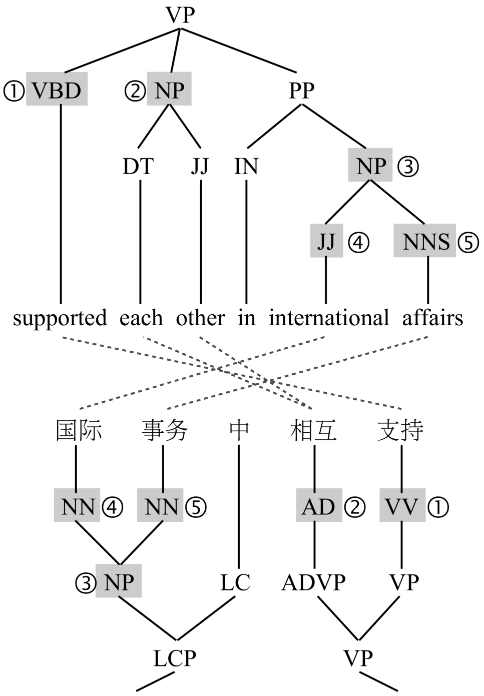

Figure 1: Sample extraction of bilingual nonterminals for label collapsing. Labels extracted from this tree pair include VBD::VV and NP::AD.

such as VBD::VV and NP::AD, can be assembled by matching co-numbered nodes.

From the counts of the extracted rules, it is thus straightforward to compute for all values of s and t the observed $P(s \mid t)$ and $P(t \mid s)$, the probability of one half of a joint nonterminal label appearing in the grammar given the other half. In the figure, for example, $P(\text{JJ} \mid \text{NN}) = 0.5$. The conditional probabilities accumulated over the whole grammar give rise to a simple L_1 distance metric over any pair of monolingual labels:

$$d(s_1, s_2) = \sum_{t \in T} |P(t \mid s_1) - P(t \mid s_2)| \qquad (1)$$

$$d(t_1, t_2) = \sum_{s \in S} |P(s \mid t_1) - P(s \mid t_2)| \qquad (2)$$

An agglomerative clustering algorithm then combines labels in a series of greedy iterations. At each step, the algorithm finds the pair of labels that is currently the closest together according to the distance metrics of Equations (1) and (2), combines those two labels into a new one, and updates the set of $P(s \mid t)$ and $P(t \mid s)$ values appropriately. The choice of label pair to collapse in each iteration can be expressed formally as

$$\operatorname*{arg\,min}_{(s_i, s_j) \in S^2, (t_k, t_\ell) \in T^2} \{d(s_i, s_j), d(t_k, t_\ell)\} \qquad (3)$$

That is, either a source label pair or a target label pair may be chosen by the algorithm in each iteration.

2.3 SAMT Rule Extraction

SAMT grammars pose a challenge to the label collapsing algorithm described above because their label sets are usually monolingual. The classic SAMT formulation (Zollmann and Venugopal, 2006) produces a grammar labeled on the target side only. Nonterminal instances that exactly match a target-language syntactic constituent in a parallel sentence are given labels of the form t. Labels of the form $t_1{+}t_2$ are assigned to nonterminals that span exactly two contiguous parse nodes. Categorial grammar labels such as t_1/t_2 and $t_1 \backslash t_2$ are given to nonterminals that span an incomplete t_1 constituent missing a t_2 node to its right or left, respectively. Any nonterminal that cannot be labeled by one of the above three schemes is assigned the default label X.

Figure 2(a) shows the extraction of a VP-level SAMT grammar rule from part of a parallel sentence. At the word level, the smaller English phrase *supported each other* (and its Chinese equivalent) is being abstracted as a nonterminal within the larger phrase *supported each other in international affairs*. The larger phrase corresponds to a parsed VP node on the target side; this will become the label of the extracted rule's left-hand side. Since the abstracted sub-phrase does not correspond to a single constituent, the SAMT labeling conventions assign it the label VBD+NP. We can thus write the extracted rule as:

$$\begin{aligned}\text{VP} \rightarrow [\ &\text{国际 事务 中 VBD+NP}^1]:: \\ &[\text{VBD+NP}^1 \text{ in international affairs}]\end{aligned} \qquad (4)$$

While the SAMT label formats can be trivially converted into joint labels X::t, X::$t_1{+}t_2$, X::t_1/t_2, X::$t_1 \backslash t_2$, and X::X, they cannot be usefully fed into the label collapsing algorithm because the necessary conditional label probabilities are meaningless. To acquire meaningful source-side labels, we turn to a

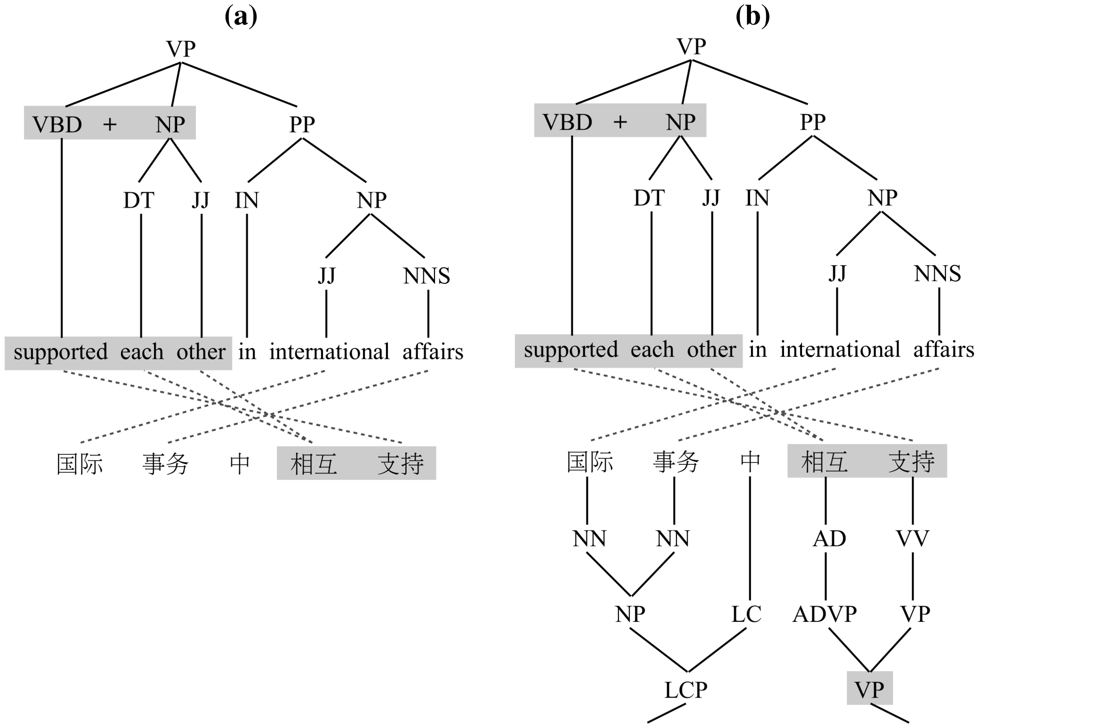

Figure 2: Sample extraction of an SAMT grammar rule: (a) with monolingual syntax and (b) with bilingual syntax.

bilingual SAMT extension used by Chiang (2010) and Zollmann (2011). Both a source- and a target-side parse tree are used to extract rules from a parallel sentence; two SAMT-style labels are worked out independently on each side for each nonterminal instance, then packed into a joint label. It is therefore possible for a nonterminal instance to be labeled s::t, $s_1\backslash s_2$::t, $s_1{+}s_2$::t_1/t_2, or various other combinations depending on what parse nodes the nonterminal spans in each tree.

Such a bilingually labeled rule is extracted in Figure 2(b). The target-side labels from Figure 2(a) are now paired with source-side labels extracted from an added Chinese parse tree. In this case, the abstracted sub-phrase *supported each other* is given the joint label VP::VBD+NP, while the rule's left-hand side becomes LCP+VP::VP.

We implement bilingual SAMT grammar extraction by modifying Thrax (Weese et al., 2011), an open-source, Hadoop-based framework for extracting standard SAMT grammars. By default, Thrax can produce grammars labeled either on the source or target side, but not both. It also outputs rules that are already scored according to a user-specified set of translation model features, meaning that the raw rule counts needed to compute the label conditional probabilities $P(s \mid t)$ and $P(t \mid s)$ are not directly available. We implement a new subclass of grammar extractor with logic for independently labeling both sides of an SAMT rule in order to get the necessary bilingual labels; an adaptation to the existing Thrax "rarity" feature provides the rule counts.

3 Label Collapsing in SAMT Rules

Our method of producing label-collapsed SAMT grammars is shown graphically in Figure 3.

We first obtain an SAMT grammar with bilingual labels, together with the frequency count for each rule, using the modified version of Thrax described in Section 2.3. The rules can be grouped according to the target-side label of their left-hand sides (Figure 3(a)).

The rule counts are then used to compute labeling probabilities $P(s \mid t)$ and $P(t \mid s)$ over left-hand-side usages of each source label s and each target label t. These are simple maximum-likelihood estimates: if $\#(s_i, t_j)$ represents the combined frequency counts of all rules with s_i::t_j on the left-hand

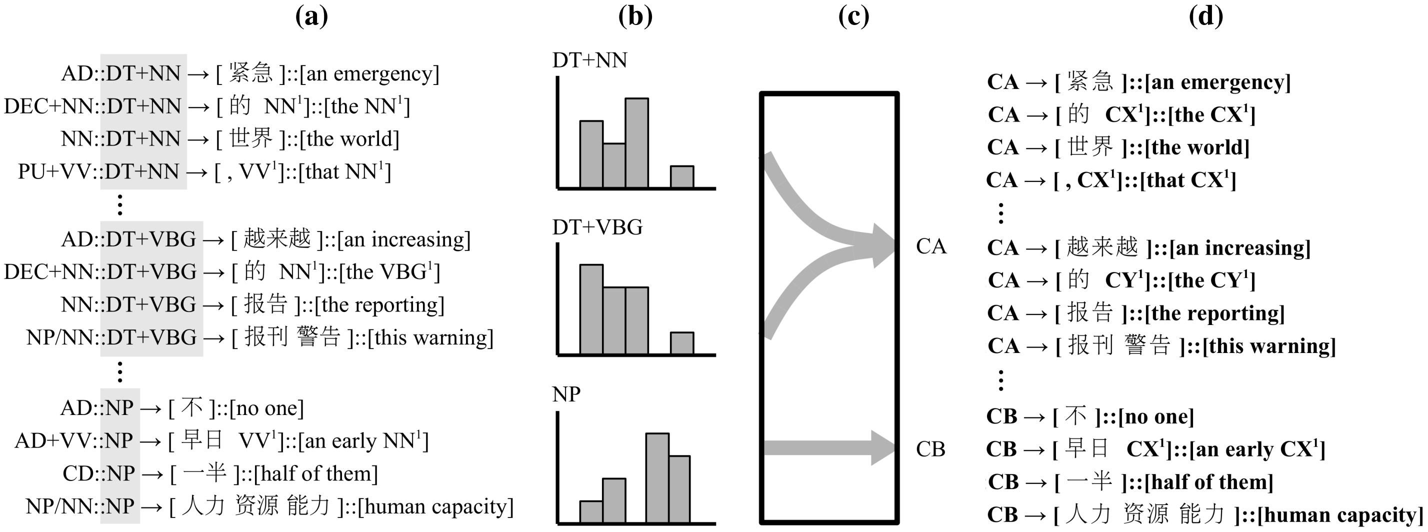
(a)
(b)
(c)
(d)
AD::DT+NN → [紧急]::[an emergency]
DEC+NN::DT+NN → [的 NN^1]::[the NN^1]
NN::DT+NN → [世界]::[the world]
PU+VV::DT+NN → [, VV^1]::[that NN^1]
AD::DT+VBG → [越来越]::[an increasing]
DEC+NN::DT+VBG → [的 NN^1]::[the VBG^1]
NN::DT+VBG → [报告]::[the reporting]
NP/NN::DT+VBG → [报刊 警告]::[this warning]
AD::NP → [不]::[no one]
AD+VV::NP → [早日 VV^1]::[an early NN^1]
CD::NP → [一半]::[half of them]
NP/NN::NP → [人力 资源 能力]::[human capacity]
DT+NN
DT+VBG
NP
CA
CB
CA → [紧急]::[an emergency]
CA → [的 CX^1]::[the CX^1]
CA → [世界]::[the world]
CA → [, CX^1]::[that CX^1]
CA → [越来越]::[an increasing]
CA → [的 CY^1]::[the CY^1]
CA → [报告]::[the reporting]
CA → [报刊 警告]::[this warning]
CB → [不]::[no one]
CB → [早日 CX^1]::[an early CX^1]
CB → [一半]::[half of them]
CB → [人力 资源 能力]::[human capacity]

Figure 3: Stages of preparing label-collapsed rules for SAMT grammars. (a) SAMT rules with bilingual nonterminals are extracted and collected based on their target left-hand sides. (b) Probabiliites $P(t \mid s)$ and $P(s \mid s)$ are computed. (c) Nonterminals are clustered according to the label collapsing algorithm. (d) Source sides of nonterminals are removed to create a standard SAMT grammar.

side, the source-given-target labeling probability is:

$$P(s_i \mid t_j) = \frac{\#(s_i{::}t_j)}{\sum_{t \in T} \#(s_i{::}t)} \tag{5}$$

The computation for target given source is analogous. Each monolingual label can thus be represented as a distribution over the labels it is aligned to in the opposite language (Figure 3(b)).

Such distributions over labels are the input to the label-collapsing algorithm, as described in Section 2.2. As shown in Figure 3(c), the algorithm results in the original target-side labels being combined into different groups, denoted in this case as new labels CA and CB. We run label collapsing for varying numbers of iterations to produce varying degrees of coarsened label sets.

Given a mapping from original target-side labels to collapsed groups, all nonterminals in the original SAMT grammar are overwritten accordingly. The source-side labels are dropped at this point: we use them only for the purpose of label collapsing, but not in assembling or scoring the final grammar. The resulting monolingual SAMT-style grammar with collapsed labels (Figure 3(d)) can now be scored and used for decoding in the usual way.

For constructing a baseline SAMT grammar without label collapsing, we merely extract a bilingual grammar as in the first step of Figure 3, immediately remove the source-side labels from it, and proceed to grammar scoring.

All grammars are scored according to a set of eight features. For an SCFG rule with left-hand-side label t, source right-hand side f, and target right-hand side e, they are:

- Standard maximum-likelihood phrasal translation probabilities $P(f \mid e)$ and $P(e \mid f)$
- Maximum-likelihood labeling probability $P(t \mid f, e)$
- Lexical translation probabilities $P_{lex}(f \mid e)$ and $P_{lex}(e \mid f)$, as calculated by Thrax
- Rarity score $\frac{\exp(\frac{1}{c})-1}{\exp(1)-1}$ for a rule with extracted count c
- Binary indicator features that mark phrase pair (as opposed to hierarchical) rules and glue rules

Scored grammars are filtered down to the sentence level, retaining only those rules whose source-side terminals match an individual tuning or testing sentence. In addition to losslessly filtering grammars in this way, we also carry out two types of lossy pruning in order to reduce overall grammar

System	Labels	Rules	Per Sent.
SAMT	4181	69,401,006	48,444
Collapse 1	913	64,596,618	35,004
Collapse 2	131	60,526,479	24,510
Collapse 3	72	58,483,310	20,445
Hiero	1	36,538,657	7,738

Table 1: Grammar statistics for different degrees of label collapsing: number of target-side labels, unique rules in the whole grammar, and average number of pruned rules after filtering to individual sentences.

size. One pruning pass keeps only the 80 most frequently observed target right-hand sides for each source right-hand side. A second pass globally removes hierarchical rules that were extracted fewer than six times in the training data.

4 Experiments

We conduct experiments on Chinese-to-English MT, using systems trained from the FBIS corpus of approximately 302,000 parallel sentence pairs. We parse both sides of the training data with the Berkeley parsers (Petrov and Klein, 2007) for Chinese and English. The English side is lowercased after parsing; the Chinese side is segmented beforehand. Unidirectional word alignments are obtained with GIZA++ (Och and Ney, 2003) and symmetrized, resulting in a parallel parsed corpus with Viterbi word alignments for each sentence pair. Our modified version of Thrax takes the parsed and aligned corpus as input and returns a list of rules, which can then be label-collapsed and scored as previously described.

In Thrax, we retain most of the default settings for Hiero- and SAMT-style grammars as specified in the extractor's configuration file. Inheriting from Hiero, we require the right-hand side of all rules to contain at least one pair of aligned terminals, no more than two nonterminals, and no more than five terminals and nonterminal elements combined. Nonterminals are not allowed to be adjacent on the source side, and they may not contain unaligned boundary words. Rules themselves are not extracted from any span in the training data longer than 10 tokens.

Our initial bilingual SAMT grammar uses 2699 unique source-side labels and 4181 unique target-side labels, leading to the appearance of 29,088 joint bilingual labels in the rule set. We provide the joint labels (along with their counts) to the label collapsing algorithm, while we strip out the source-side labels to create the baseline SAMT grammar with 4181 unique target-side labels. Table 1 summarizes how the number of target labels, unique extracted rules, and the average number of pruned rules available per sentence change as the initial grammar is label-collapsed to three progressively coarser degrees. Once the collapsing process has occurred exhaustively, the original SAMT grammar becomes a Hiero-format grammar with a single nonterminal.

Each of the five grammars in Table 1 is used to build an MT system. All systems are tuned and decoded with cdec (Dyer et al., 2010), an open-source decoder for SCFG-based MT with arbitrary rule formats and nonterminal labels. We tune the systems on the 1664-sentence NIST Open MT 2006 data set, optimizing towards the BLEU metric. Our test sets are the NIST 2003 data set of 919 sentences and the NIST 2008 data set of 1357 sentences. The tuning set and both test sets all have four English references.

We evaluate systems on BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011), and TER (Snover et al., 2006), as calculated in all three cases by MultEval version 0.5.0.[1] These scores for the MT '03 test set are shown in Table 2, and those for the MT '08 test set in Table 3, combined by MultEval over three optimization runs on the tuning set.

MultEval also implements statistical significance testing between systems based on multiple optimizer runs and approximate randomization. This process (Clark et al., 2011) randomly swaps outputs between systems and estimates the probability that the observed score difference arose by chance. We report these results in the tables as well for three MERT runs and a p-value of 0.05. Systems that were judged statistically different from the SAMT baseline have triangles in the appropriate "Sig. SAMT?" columns; systems judged different from the Hiero baseline have triangles under the "Sig. Hiero?" columns. An up-triangle (▲) indicates that the system was better, while a down-triangle (▽) means that the baseline was better.

[1] https://github.com/jhclark/multeval

	Metric Scores			Sig. SAMT?			Sig. Hiero?		
System	BLEU	MET	TER	B	M	T	B	M	T
SAMT	31.18	30.64	61.02				▽	▽	▽
Collapse 1	31.42	31.31	60.95		▲		▽		▽
Collapse 2	31.90	31.73	60.98	▲	▲		▽	▲	▽
Collapse 3	32.32	31.75	60.54	▲	▲	▲		▲	▽
Hiero	32.30	31.42	60.10	▲	▲	▲			

Table 2: MT ’03 test set results. The first section gives automatic metric scores; the remaining sections indicate whether each system is statistically significantly better (▲) or worse (▽) than the SAMT and Hiero baselines.

	Metric Scores			Sig. SAMT?			Sig. Hiero?		
System	BLEU	MET	TER	B	M	T	B	M	T
SAMT	22.10	24.94	63.78				▽	▽	▽
Collapse 1	23.01	26.03	63.35	▲	▲	▲		▲	
Collapse 2	23.53	26.50	63.29	▲	▲	▲	▲	▲	
Collapse 3	23.61	26.37	63.07	▲	▲	▲	▲	▲	▲
Hiero	23.01	25.72	63.53	▲	▲	▲			

Table 3: MT ’08 test set results. The first section gives automatic metric scores; the remaining sections indicate whether each system is statistically significantly better (▲) or worse (▽) than the SAMT and Hiero baselines.

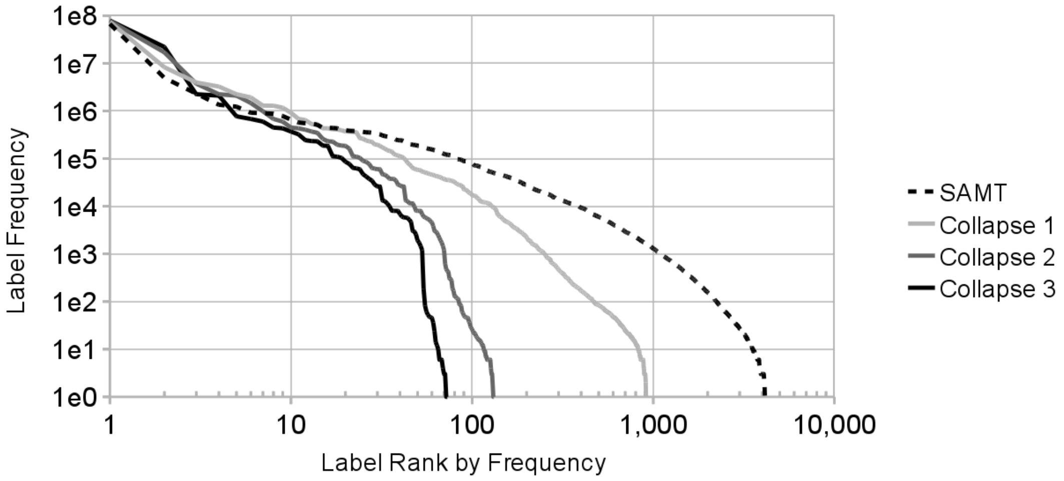

Figure 4: Extracted frequency of each target-side label, with labels arranged in order of decreasing frequency count. Note the log–log scale of the plot.

5 Analysis

Tables 2 and 3 show that the coarsened grammars significantly improve translation performance over the SAMT baseline. This is especially true for the "Collapse 3" setting of 72 labels, which scores 1.14 BLEU higher on MT '03 and 1.51 BLEU higher on MT '08 than the uncollapsed system.

On the easier MT '03 set, label-collapsed systems do not generally outperform Hiero, although Collapse 3 achieves a statistical tie according to BLEU (+0.02) and a statistical improvement over Hiero according to METEOR (+0.33). MT '08 appears as a significantly harder test set: metric scores for all systems are drastically lower, and we find approximately 7% to 8% fewer phrase pair matches per sentence. In this case the label-collapsed systems perform better, with all three of them achieving statistical significance over Hiero in at least one metric and statistical ties in the other. The coarsened systems' comparatively better performance on the harder test set suggests that the linguistic information encoded in multiple-nonterminal grammars helps the systems more accurately parse new types of input.

Table 1 already showed at a global scale the strong effect of label collapsing on reducing derivational ambiguity, as labeled variants of the same basic structural rule were progressively combined. Since category coarsening is purely a relabeling operation, any reordering pattern implemented in the original SAMT grammar still exists in the collapsed versions; therefore, any reduction in the size of the grammar is a reduction in variant labelings. Figure 4 shows this process in more detail for the baseline SAMT grammar and the three collapsed grammars. For each grammar, labels are arranged in decreasing order of extracted frequency, and the frequency count of each label is plotted. The long tail of rare categories in the SAMT grammar (1950 labels seen fewer than 100 times each) is combined into a progressively sharper distribution at each step. Not only are there fewer rare labels, but these hard-to-model categories consume a proportionally smaller fraction of the total label set: from 47% in the baseline grammar down to 26% in Collapse 3.

We find that label collapsing disproportionately affects frequently extracted and hierarchical rules over rarer rules and phrase pairs. The 15.7% reduction in total grammar size between the SAMT baseline and the Collapse 3 system affects 18.0% of the hierarchical rules, but only 1.6% of the phrase pairs. If rules are counted separately each time they match another source sentence, the average reduction in size of a sentence-filtered grammar is 57.8%.

Intuitively, hierarchical rules are more affected by label collapsing because phrase pairs do not have many variant left-hand-side labels to begin with, while the same hierarchical rule pattern may be instantiated in the grammar by a large number of variant labelings. We can see this situation in more detail by counting variants of a particular set of rules. Labeled forms of the Hiero-style rule

$$X \rightarrow [X^1 \text{的} X^2] :: [\text{the } X^2 \text{ of } X^1] \qquad (6)$$

are among the most frequently used rules in all five of our systems. The way they are treated by label collapsing thus has a strong impact on the results of runtime decoding.

In the SAMT baseline, Rule (6) appears in the grammar with 221 different labels in the X^1 nonterminal slot, 53 labels for the X^2 slot, and 90 choices of left-hand side — a total of 1330 different labelings all together. More than three-fourths of these variants were extracted three times or fewer from the training data; even if they can be used in a test sentence, statistical features for such low-count rules are poorly estimated. During label collapsing, the number of labeled variations of Rule (6) drops from 1330 to 325, to 96, and finally to 63 in the Collapse 3 grammar. There, the pattern is instantiated with 14 possible X^1 labels, five X^2 labels, and three different left-hand sides.

It is difficult to measure rule sparsity directly (i.e. to count the number of rules that are missing during decoding), but a *reduction* in rule sparsity between systems should be manifested as an increased number of hierarchical rule applications. Figure 5 shows the average number of hierarchical rules applied per sentence, distinguishing syntactic rules from glue rules, on both test sets. The collapsed grammars allow for approximately one additional syntactic rule application per sentence compared to the SAMT baseline, or three additional applications compared to Hiero. This shows an implicit reduction in missing syntactic rules in the collapsed grammars. In the

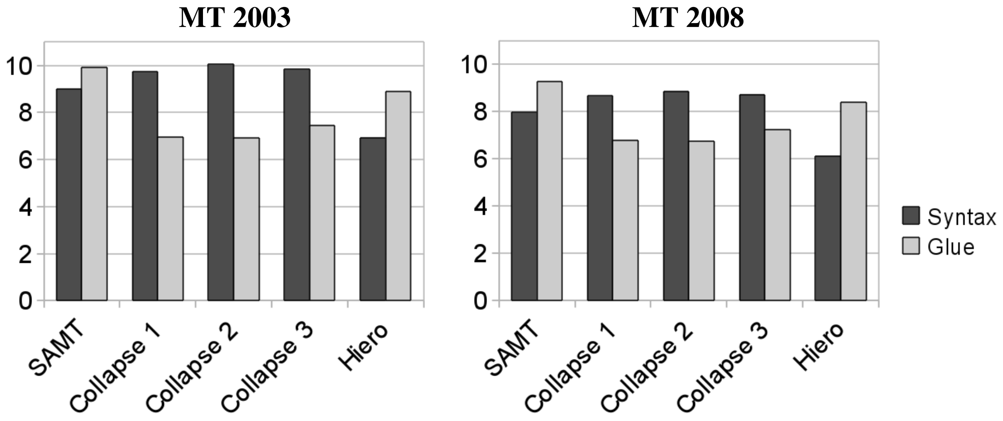

Figure 5: Average number of hierarchical rules (both syntactic and glue rules) applied per sentence on each test set.

glue rule columns, we note that label collapsing also promotes a shift away from generic glue rules, possibly via the creation of more permissive — but still meaningfully labeled — syntactic rules.

6 Conclusion

We demonstrated a viable technique for reducing the label set size in SAMT grammars by temporarily inducing bilingual syntax and using it in an existing tree-to-tree category coarsening algorithm. In collapsing SAMT category labels, we were able to significantly improve translation quality while using a grammar less than half the size of the original. We believe it is also more robust to test-set or domain variation than a single-nonterminal Hiero grammar. Collapsed grammars confer practical benefits during both model estimation and runtime decoding. We showed that, in particular, they suffer less from rule sparsity and derivational ambiguity problems that are common to larger label sets.

We can highlight two areas for potential improvements in future work. In our current implementation of label collapsing, we indiscriminately allow either source labels or target labels to be collapsed at each iteration of the algorithm (see Equation 3). This is an intuitively sensible setting when collapsing bilingual labels, but it is perhaps less obviously so for a monolingually labeled system such as SAMT. An alternative would be to collapse target-side labels only, leaving the source-side labels alone since they do not appear in the final grammar anyway. In this case, the target labels would be represented and clustered as distributions over a static set of latent categories.

A larger area of future concern is the stopping point of the collapsing algorithm. In our previous work (Hanneman and Lavie, 2011), we manually identified iterations in our run of the algorithm where the L_1 distance between the most recently collapsed label pair was markedly lower than the L_1 difference of the pair in the previous iteration. Such an approach is more feasible in our previous runs of 120 iterations than in ours here of nearly 2100, where it is not likely that three manually chosen stopping points represent the optimal collapsing results. In future work, we plan to work towards the development of an automatic stopping criterion, a more principled test for whether each successive iteration of label collapsing provides some useful benefit to the underlying grammar.

Acknowledgments

This research work was supported in part by computing resources provided by the NSF-sponsored XSEDE program under grant TG-CCR110017. Thanks to Chris Dyer for providing the word-aligned and preprocessed corpus we used in our experiments. We also thank the anonymous reviewers for helpful comments and suggestions for analysis.

References

Vamshi Ambati, Alon Lavie, and Jaime Carbonell. 2009. Extraction of syntactic translation models from parallel data using syntax from source and target languages.

In *Proceedings of the 12th Machine Translation Summit*, pages 190–197, Ottawa, Canada, August.

Kathryn Baker, Michael Bloodgood, Bonnie J. Dorr, Chris Callison-Burch, Nathaniel W. Filardo, Christine Piatko, Lori Levin, and Scott Miller. 2012. Use of modality and negation in semantically-informed syntactic MT. *Computational Linguistics*, 38(2):411–438.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

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

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Crontrolling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 176–181, Portland, OR, June.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, United Kingdom, July.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12, Uppsala, Sweden, July.

Michel Galley, 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.

Greg Hanneman and Alon Lavie. 2011. Automatic category label coarsening for syntax-based machine translation. In *Proceedings of SSST-5: Fifth Workshop on Syntax, Semantics, and Structure in Statistical Translation*, pages 98–106, Portland, OR, June.

Bryant Huang 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.

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

Alon Lavie, 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.

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

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation 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.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, 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.

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

Matthew Snover, 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.

Ashish Venugopal, 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.

Jonathan Weese, Juri Ganitkevitch, Chris Callison-Burch, Matt Post, and Adam Lopez. 2011. Joshua 3.0: Syntax-based machine translation with the Thrax grammar extractor. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 478–484, Edinburgh, United Kingdom, July.

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

Andreas Zollmann. 2011. *Learning Multiple-Nonterminal Synchronous Grammars for Machine Translation*. Ph.D. thesis, Carnegie Mellon University.