Automatic Category Label Coarsening for Syntax-Based Machine Translation

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

Abstract

We consider SCFG-based MT systems that get syntactic category labels from parsing both the source and target sides of parallel training data. The resulting joint nonterminals often lead to needlessly large label sets that are not optimized for an MT scenario. This paper presents a method of iteratively coarsening a label set for a particular language pair and training corpus. We apply this label collapsing on Chinese–English and French–English grammars, obtaining test-set improvements of up to 2.8 BLEU, 5.2 TER, and 0.9 METEOR on Chinese–English translation. An analysis of label collapsing's effect on the grammar and the decoding process is also given.

1 Introduction

A common modeling choice among syntax-based statistical machine translation systems is the use of synchronous context-free grammar (SCFG), where a source-language string and a target-language string are produced simultaneously by applying a series of re-write rules. Given a parallel corpus that has been statistically word-aligned and annotated with constituency structure on one or both sides, SCFG models for MT can be learned via a variety of methods. Parsing may be applied on the source side (Liu et al., 2006), on the target side (Galley et al., 2004), or on both sides of the parallel corpus (Lavie et al., 2008; Zhechev and Way, 2008).

In any of these cases, using the raw label set from source- and/or target-side parsers can be undesirable. Label sets used in statistical parsers are usually inherited directly from monolingual treebank projects, where the inventory of category labels was designed by independent teams of human linguists. These labels sets are not necessarily ideal for statistical parsing, let alone for bilingual syntax-based translation models. Further, the side(s) on which syntax is represented defines the nonterminal label space used by the resulting SCFG. A pair of aligned adjectives, for example, may be labeled ADJ if only source-side syntax is used, JJ if only target-side syntax is used, or ADJ::JJ if syntax from both sides is used in the grammar. Beyond such differences, however, most existing SCFG-based MT systems do not further modify the nonterminal label set in use. Those that do require either specialized decoders or complicated parameter tuning, or the label set may be unsatisfactory from a computational point of view (Section 2).

We believe that representing both source-side and target-side syntax is important. Even assuming two monolingually perfect label sets for the source and target languages, using label information from only one side ignores any meaningful constraints expressed in the labels of the other. On the other hand, using the default node labels from both sides generates a joint nonterminal set of thousands of unique labels, not all of which may be useful. Our real preference is to use a joint nonterminal set adapted to our particular language pair or translation task.

In this paper, we present the first step towards a tailored label set: collapsing syntactic categories to remove the most redundant labels and shrink the overall source–target nonterminal set.[1] There are

[1] The complementary operation, splitting existing labels, is beyond the scope of this paper and is left for future work.

Proceedings of SSST-5, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 98–106,
ACL HLT 2011, Portland, Oregon, USA, June 2011. ©2011 Association for Computational Linguistics

two problems with an overly large label set:

First, it encourages labeling ambiguity among rules, a well-known practical problem in SCFG-based MT. Most simply, the same right-hand side may be observed in rule extraction with a variety of left-hand-side labels, each leading to a unique rule in the grammar. The grammar may further contain many rules with the same structure and reordering pattern that differ only with respect to the actual labels in use. Together, these properties can cause an SCFG-based MT system to process a large number of alternative syntactic derivations that use different rules but produce identical output strings. Limiting the possible number of variant labelings cuts down on ambiguous derivations.

Second, a large label set leads to rule sparsity. A rule whose right-hand side can only apply on a very tightly specified set of labels is unlikely to be estimated reliably from a parallel corpus or to apply in all needed cases at test time. However, a coarser version of its application constraints may be more frequently observed in training data and more likely to apply on test data.

We therefore introduce a method for automatically clustering and collapsing category labels, on either one or both sides of SCFG rules, for any language pair and choice of statistical parsers (Section 3). Turning to alignments between source and target parse nodes as an additional source of information, we calculate a distance metric between any two labels in one language based on the difference in alignment probabilities to labels in the other language. We then apply a greedy label collapsing algorithm that repeatedly merges the two labels with the closest distance until some stopping criterion is reached. The resulting coarsened labels are used in the SCFG rules of a syntactic machine translation system in place of the original labels.

In experiments on Chinese–English translation (Section 4), we find significantly improved performance of up to 2.8 BLEU points, 5.2 TER points, and 0.9 METEOR points by applying varying degrees of label collapsing to a baseline syntax-based MT system (Section 5). In our analysis of the results (Section 6), we find that the largest immediate effect of coarsening the label set is to reduce the number of fully abstract hierarchical SCFG rules present in the grammar. These rules' increased permissiveness, in turn, directs the decoder's search into a largely disjoint realm from the search space explored by the baseline system. A full summary and ideas for future work are given in Section 7.

2 Related Work

One example of modifying the SCFG nonterminal set is seen in the Syntax-Augmented MT (SAMT) system of Zollmann and Venugopal (2006). In SAMT rule extraction, rules whose left-hand sides correspond exactly to a target-side parse node t retain that label in the grammar. Additional nonterminal labels of the form $t_1 + t_2$ are created for rules spanning two adjacent parse nodes, while categorial grammar–style nonterminals t_1/t_2 and $t_1\backslash t_2$ are used for rules spanning a partial t_1 node that is missing a t_2 node to its right or left.

These compound nonterminals in practice lead to a very large label set. Probability estimates for rules with the same structure up to labeling can be combined with the use of a preference grammar (Venugopal et al., 2009), which replaces the variant labelings with a single SCFG rule using generic "X" labels. The generic rule's "preference" over possible labelings is stored as a probability distribution inside the rule for use at decoding time. Preference grammars thus reduce the label set size to one for the purposes of some feature calculations — which avoids the fragmentation of rule scores due to labeling ambiguity — but the original labels persist for specifying which rules may combine with which others.

Chiang (2010) extended SAMT-style labels to both source- and target-side parses, also introducing a mechanism by which SCFG rules may apply at run time even if their labels do not match. Under Chiang's soft matching constraint, a rule headed by a label A::Z may still plug into a substitution site labeled B::Y by paying additional model costs $subst_{B\to A}$ and $subst_{Y\to Z}$. This is an on-the-fly method of coarsening the effective label set on a case-by-case basis. Unfortunately, it also requires tuning a separate decoder feature for each pair of source-side and each pair of target-side labels. This tuning can become prohibitively complex when working with standard parser label sets, which typically contain between 30 and 70 labels on each side.

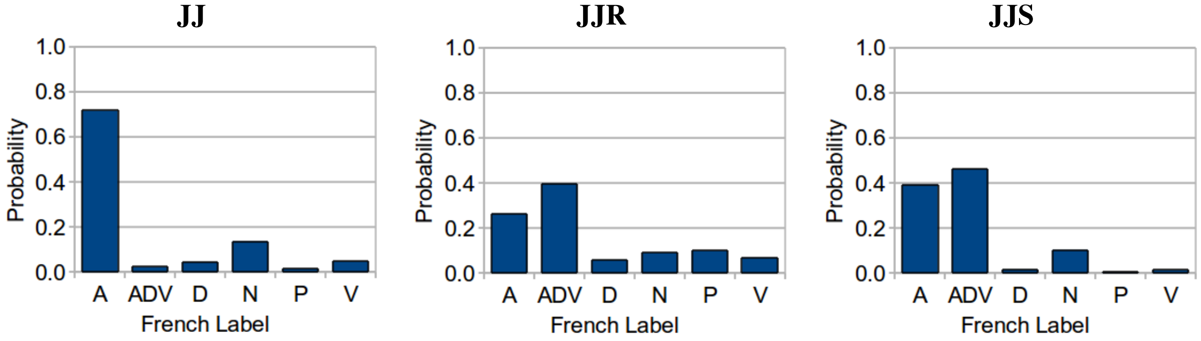


Figure 1: Alignment distributions over French labels for the English adjective labels JJ, JJR, and JJS.

3 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 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 distribution $P(s \,|\, t)$ by simple counting and normalizing:

$$P(s \,|\, t) = \frac{\#(s :: t)}{\#(t)} \qquad (1)$$

We use an analogous equation to calculate $P(t \,|\, s)$. For two target-language labels t_1 and t_2, we have an equally simple metric of alignment distribution difference d: the total of the absolute differences in likelihood for each aligned source-language label.

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

Again, the calculation for $d(s_1, s_2)$ is analogous.

If t_1 and t_2 are plotted as points in $|S|$-dimensional space such that each point's position in dimension s is equal to $P(s \,|\, t)$, then this metric is equivalent to the L_1 distance between t_1 and t_2.

Sample alignment distributions into French for three English adjective labels are shown in Figure 1. Bars in the chart represent alignment probabilities between French and English according to Equation 1, with the various French labels as s and JJ, JJR, or JJS as t. To compute an L_1 alignment distribution difference between a pair of English adjective tags, we sum the absolute differences in bar heights for each column of two graphs, as in Equation 2. It is already visually clear from Figure 1 that all three English labels are somewhat related in terms of distribution, but it appears that JJR and JJS are more closely related to each other than either is to JJ. This is reflected in the actual L_1 distances: $d(\text{JJ}, \text{JJR}) = 0.9941$ and $d(\text{JJ}, \text{JJS}) = 0.8730$, but $d(\text{JJR}, \text{JJS}) = 0.3996$.

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 the L_1 distance between all pairs of labels, then collapse the pair with the smallest distance into a single label. Then L_1 distances are recomputed over the new, smaller label set, and again the label pair with the smallest distance is collapsed. This process continues until some stopping criterion is reached. Label pairs being considered for collapsing may be only source-side labels, only target-side labels, or both. In general, we choose to allow label collapsing to apply on either side during each iteration of our algorithm.

In the limit, label collapsing can be applied iteratively until all syntactic categories on both the source and target sides have been collapsed into a single label. In Section 5, we explore several earlier and more meaningful stopping points.

4 Experimental Setup

Experiments are conducted on Chinese-to-English translation using approximately 300,000 sentence pairs from the FBIS corpus. To obtain parse trees over both sides of each parallel corpus, we used the English and Chinese grammars of the Berkeley parser (Petrov and Klein, 2007).

Given a parsed and word-aligned parallel sentence, we extract SCFG rules from it following the procedure of Lavie et al. (2008). The method first identifies node alignments between the two parse trees according to support from the word alignments. A node in the source parse tree will be aligned to a node in the target parse tree if all the words in the yield of the source node are either all aligned to words within the yield of the target node or have no alignments at all. 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 rule, each decomposition point also produces a phrase pair rule with the node pair's yields as the right-hand side, as long as the length of the yield is less than a specified threshold.

Following grammar extraction, labels are optionally clustered and collapsed according to the algorithm in Section 3. The grammar is re-written with the modified nonterminals, then scored as usual according to our translation model features. Feature weights themselves are learned via minimum error rate training as implemented in Z-MERT (Zaidan, 2009) with the BLEU metric (Papineni et al., 2002). Decoding is carried out with Joshua (Li et al., 2009), an open-source platform for SCFG-based MT.

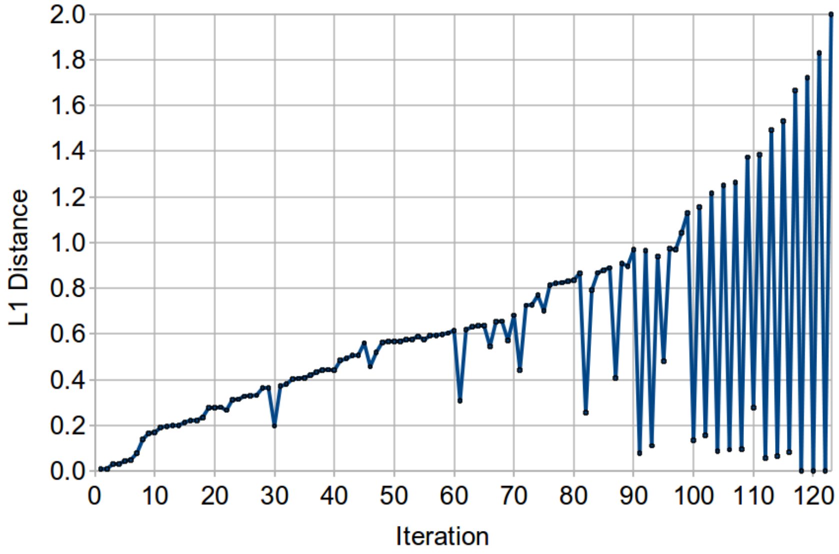
Due to engineering limitations in decoding with a large grammar, we apply three additional error-correction and filtering steps to every system. First, we observed that the syntactic parsers were most likely to make labeling errors for cardinal numbers in English and punctuation marks in all languages. We thus post-process the parses of our training data to tag all English cardinal numbers as CD and to overwrite the labels of various punctuation marks with the correct labels as defined by each language's label set. Second, after rule extraction, we compute the distribution of left-hand-side labels for each unique labeled right-hand side in the grammar, and we remove the labels in the least frequent 10% of the distribution. This puts a general-purpose limit on labeling ambiguity. Third, we filter and prune the final scored grammar to each individual development and test set before decoding: all matching phrase pairs are retained, along with the most frequent 10,000 hierarchical grammar rules.

5 Experiments and Results

In our first set of experiments, we sought to explore the effect of increasing degrees of label collapsing on a baseline system and to determine a reasonable stopping point. Starting with the baseline grammar, we ran the label collapsing algorithm of Section 3 until all the constituent labels on each side had been collapsed into a single category. We next examined the L_1 distances between the label pairs that had been merged in each iteration of the algorithm. This data is shown in Figure 2 as a plot of L_1 distance versus iteration number. The distances between the successive labels merged in the first 29 iterations of the algorithm are nearly monotonically increasing, followed by a much larger discontinuity at iteration 30. Similar patterns emerge for iterations 30 to 45 and for iterations 46 to 60. The next regions of the graph, from iterations 61 to 81 and from iterations 82 to 99, show an increasing prevalence of discontinuities. Finally, from iterations 100 to 123, the successive L_1 distances entirely alternate between very high and very low values.

Discontinuities 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. Looking back to Figure 1, for example, we could bring the distributions for JJ and JJS much closer together by merging A and ADV on the French side. 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 that categories previously very different have become much less distinguishable. On the other hand, further reduction of the label set may have a variety of pratical benefits.

We tested this trade-off empirically by building five Chinese–English MT systems, each exhibiting an increasing degree of label collapsing compared to the original label set, which serves as our baseline. The degree of label collapsing in each of the five systems corresponds to one of the major discontinuity features highlighted in the right-hand side Figure 2. The systems were tuned on the NIST MT 2006 data set, and we evaluated performance on the NIST MT 2003 and 2008 sets. (All data sets have four



Iter.	L_1 Dist.
29	0.3646
45	0.5607
60	0.6155
81	0.8665
99	1.1303

Figure 2: Observed L_1 distance values for the labels merged in each iteration of our algorithm on a Chinese–English SCFG. We divide the graph into six distinct regions using the cutoffs at right.

Chinese–English	**MT 2003 Test Set**			**MT 2008 Test Set**		
System	**METEOR**	**BLEU**	**TER**	**METEOR**	**BLEU**	**TER**
Baseline	54.35	24.39	68.01	45.68	18.27	69.18
Collapsed, 29 iterations	**55.24**	27.03	63.77	46.25	19.78	65.88
Collapsed, 45 iterations	54.65	26.69	**62.76**	46.02	19.60	**64.88**
Collapsed, 60 iterations	55.11	**27.23**	63.06	**46.30**	20.19	65.18
Collapsed, 81 iterations	54.87	26.87	64.92	45.70	**20.48**	66.75
Collapsed, 99 iterations	54.86	26.16	64.17	45.87	19.52	65.61

Table 1: Results of applying increasing degrees of label collapsing on our Chinese–English baseline system. Bold figures indicate the best score in each column.

references.) Table 1 reports automatic metric results for version 1.0 of METEOR (Lavie and Denkowski, 2009) using the default settings, uncased IBM-style BLEU (Papineni et al., 2002), and uncased TER version 0.7 (Snover et al., 2006).

No matter the degree of label collapsing, we find significant improvements in BLEU and TER scores on both test sets. On the MT 2003 set, label-collapsed systems score 1.77 to 2.84 BLEU points and 3.09 to 5.25 TER points better than the baseline. On MT 2008, improvements range from 1.25 to 2.21 points on BLEU and from 2.43 to 4.30 points on TER. Improvements on both sets according to METEOR, though smaller, are still noticable (up to 0.89 points). In the case of BLEU, we verified the significance of the improvements by conducting paired bootstrap resampling (Koehn, 2004) on the MT 2003 output. With $n = 1000$ and $p < 0.05$, all five label-collapsed systems were statistically significant improvements over the baseline, and all other collapsed systems were significant improvements over the 99-iteration system.

Thus, though the system that provides the highest score changes across metrics and test sets, the overall pattern of scores suggests that over-collapsing labels may start to weaken results. A more moderate stopping point is thus preferable, but beyond that we suspect the best result is determined more by the test set, automatic metric choice, and MERT instability than systematic changes in the label set.

6 Analysis

Table 1 showed a strong practical benefit to running the label collapsing algorithm. In this section, we seek to further understand where this benefit comes from, tracing the effects of label collapsing via its modification of labels themselves, the differences in the resulting grammars, and collapsing's effect on decoding and output.

6.1 Labels Selected for Collapsing

Our first concern is for the size of the grammar's overall nonterminal set. The baseline system uses a total of 55 labels on the Chinese side and 71 on the English side, leading to an observed joint nonterminal set of 1556 unique labels. After 29 iterations of label collapsing, this is reduced to 46 Chinese, 51 English, and 1035 joint labels — a reduction of 33%. In the grammar of our most collapsed grammar variant (99 iterations), the nonterminal set is reduced to 14 English and 14 Chinese labels, for a total of 106 joint labels and a reduction of 93% from the baseline grammar. This demonstrates one facet of our introductory claim from Section 1: since we have improved translation results by removing the vast majority of our grammar nonterminals, most of the initial joint Chinese–English syntactic categories were not necessary for Chinese–English translation.

We identify three broad trends in the sets of labels that are collapsed:

- **Full Subtype Collapsing.** The Chinese-side parses include six phrase-level tags for various types of verb compounds. As label collapsing progresses, these labels are all combined with each other at relatively low L_1 distances.

- **Partial Subtype Collapsing.** In English, three of the four noun labels (NN, NNS, and NNPS) form a cohesive cluster early on in Chinese–English collapsing. However, the fourth tag (NNP, for singular proper nouns) remains separate, then later joins a cluster for more adjective-like labels.

- **Combination by Syntactic Function.** In French–English label collapsing (see below), we find the creation of a combined label in English for reduced relative clauses (RRC), adjective phrases headed by a *wh*-adjective (WHADJP), and interjections (INTJ). Even though these tags are unrelated in surface form, at some level they all represent parenthetical insertions or explanatory phrases.

The formulation of the L_1 distance metric in Section 3 means that our label collapsing algorithm will naturally produce different label clusters for different input grammars — any change in the Viterbi word alignments, underlying parallel corpus, initial label set, or choice of automatic parser will necessarily change the label alignment distributions on which the collapsing algorithm is based. In particular, the label clusters formed in one language are likely to be markedly different depending on which other language it is paired with. We examine these differences in more detail for the case of English when paired with either Chinese or with French. Our 29-iteration run of label collapsing for Chinese–English merged labels on the English side 19 times. For an exact comparison, we run iterations of label collapsing on a large-scale French–English grammar, extracted in the same way as the Chinese–English grammar, until the same number of English-side merges have been carried out, then examine the results.

Table 2 shows the English label clusters created from the Chinese–English and French–English grammars, arranged by broad syntactic categories. The differences in English label clusters hint at differences in the source-side label sets, as well as structural divergences relevant for translating Chinese versus French into English.

For example, Table 2 shows partial subtype collapsing of the English verb tags when paired with French. The French Berkeley parser has a single tag, V, to represent all verbs, and most English verb tags as well as the tag for modals very consistently align to it. The exception is VBG, for present-progressive or gerundive verb forms, which is more easily conflatable in French–English translation with a noun or an adjective. In translation from Chinese, however, it is VBG that is combined early on with a smaller selection of English verb labels that correspond most strongly to a basic Chinese verb. Other English verb tags are more likely to align to Chinese copulas, existential verbs, and nouns; they are not combined with the group for more "typical" verbs until iteration 67. The adverb series presents another example of translational divergence between language pairs.

Cluster	Chinese–English	French–English
Nouns	NN NNS NNPS #	NN NNS $
Verbs	VB VBG VBN	VB VBD VBN VBP VBZ MD
Adverbs	RB RBR	RBR RBS
Punctuation	LRB RRB “ ” , .	“ ”
Prepositions		IN TO SYM
Determiners		DT PRP$
Noun phrases	NP NX QP UCP NAC	NP WHNP NX WHADVP NAC
Adjective phrases	ADJP WHADJP	
Adverb phrases	ADVP WHADVP	
Prepositional phrases		PP WHPP
Sentences	S SINV SBARQ FRAG	S SQ SBARQ

Table 2: English-side label clusters created after partial label collapsing of a Chinese–English and a French–English grammar. In each case, the algorithm has been run until merges have occurred 19 times on the English side.

6.2 Effect on the Grammar

With a smaller label set, we also expect a reduction in the overall size of our various label-collapsed grammars as labeling ambiguity is removed. In the aggregate, however, even 99 iterations of Chinese–English label collapsing has a minimal effect on the total number of unique rules in the resulting SCFG. A clearer picture emerges when we separate rules according to their form. Figure 3 partitions the grammar into three parts: one for phrase pairs, where the rules’ right-hand sides are made up entirely of terminals (“P-type” rules); one for hierarchical rules whose right-hand sides are made up entirely of nonterminals (abstract or “A-type” rules); and one for hierarchical rules whose right-hand sides include a mix of terminals and nonterminals (remaining grammar or “G-type” rules).

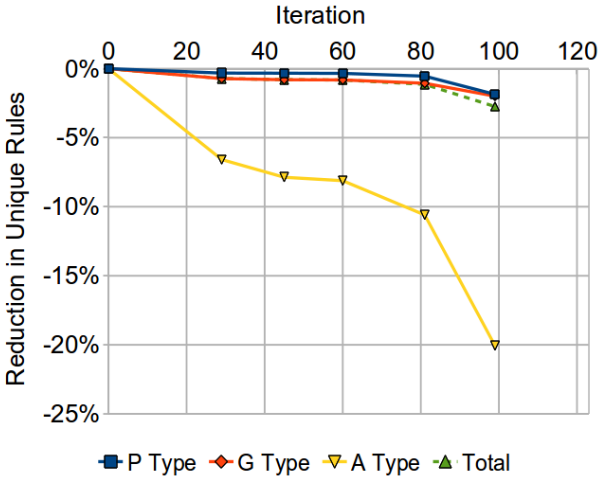


Figure 3: The effect of label collapsing on the number of unique phrase pairs, partially lexicalized grammar rules, and fully abstract grammar rules.

This separation reveals two interesting facts. First, although the size of the label set continues to shrink considerably between iterations 29 and 81, the number of unique rules in the grammar remains relatively unchanged. Second, the reduction in the size of the grammar is largely due to a reduction in the number of fully abstract grammar rules, rather than phrase pairs or partially lexicalized grammar rules. From these observations, we infer that the major practical benefit of label collapsing is a reduction in rule sparsity rather than a reduction in left-hand-side labeling ambiguity. Many highly ambiguous rules have had their possible left-hand-side labels effectively pruned down by the pre-processing steps we described in Section 4, which in preliminary experiments had a larger effect on the overall size of the grammar than label collapsing. As a more complementary technique, increasing the applicability of the fully abstract rules via label collapsing is important for performance. Such rules make up 49% to 59% of the hierarchical rules retained at decoding time, and they account for 76% to 87% of the rule application instances on the MT 2003 test set.

6.3 Effect on Decoding and Output

Interestingly, the label collapsing algorithm does not owe its success at decoding time to a significant increase in the number of rule applications. Among our systems, both the 45-iteration and the

60-iteration collapsed versions scored highly according to automatic metrics. Nevertheless, the 45-iteration system used 32% and 38% more rule applications than the baseline on the MT 2003 and MT 2008 test sets, respectively, while the 60-iteration system used 15% and 11% fewer. The number of unique rule types and the number of reordering rules applied on a test set may also go up or down.

Instead, the practical effect of making the grammar more permissive seems to be a significant change in the search space explored during decoding. This can be seen superficially via an examination of output n-best lists. On both test sets combined (2276 sentences), the 60-iteration label-collapsed system's top-best output appears in the baseline's 100-best list in only 81 sentences. When it does appear in the baseline, the improved system's translation is ranked fairly highly — always 30th place or higher. Conversely, the baseline's top-best output tends to be ranked lower in the improved system's n-best list: among the 114 times it appears, it is placed as low as 87th.

We ran a small follow-up analysis on the translation fragments explored during decoding. Using a modified version of the Joshua decoder, we dumped lists of hypergraph entries that were explored by cube pruning during Joshua's lazy generation of a 100-best list. These entries represent the decoder's approximative search through the larger space of translations licenced by the grammar for each test sentence. We then compared the hypergraph entries, excluding glue rules, produced on the first 100 sentences of the MT 2003 test set by both the baseline and the 60-iteration label-collapsed system.

A full 90% of the entries produced by the label-collapsed system had no analogue in the baseline system. The average length of the entries that do match is 2.3 source words, compared with an average of 6.2 words for the non-matched entries. We believe that the increased permissiveness of the hierarchical grammar rules is again the root cause of these results. Low-level constituents are more likely to be matched in both the baseline and the label-collapsed system, but different applications of the grammar rules, perhaps combined with retuned feature weights, leads the search for larger translation fragments into new areas.

7 Conclusions and Future Work

This paper has presented a language-specific method for automatically coarsening the label set used in an SCFG-based MT system. Our motivation for collapsing labels comes from the intuition that the full cross-product of joint source–target labels, as produced by statistical parsers, is too large and not specifically created for bilingual MT modeling. The greedy collapsing algorithm we developed is based on iterative merging of the two single-language labels whose alignment distributions are most similar according to a simple L_1 distance metric.

In applying varying degrees of label collapsing to a baseline MT system, we found significantly improved automatic metric results even when the size of the joint label set had been reduced by 93%. The best results, however, were obtained with more moderate coarsening. The coarser labels that our method produces are syntactically meaningful and represent specific cross-language behaviors of the language pair involved. At the grammar level, label collapsing primarily caused a reduction in the number of rules whose right-hand sides are made up entirely of nonterminals. The coarser labels made the grammar more permissive, cutting down on the problem of rule sparsity. Labeling ambiguity, on the other hand, was more effectively addressed by pre-processing we applied to the grammar beforehand. At run time, the more permissive collapsed grammar allowed the decoder to search a markedly different region of the allowable translation space than in the baseline system, generally leading to improved output.

One shortcoming of our current algorithm is that it is based entirely on label alignment distribution without regard to the different contexts in which labels occur. It thus cannot distinguish between two labels that align similarly but appear in very different rules. For example, singular common nouns (NN) and plural proper nouns (NNPS) in English both most frequently align to French nouns (N) and are thus strong candidates for label collapsing under our algorithm. However, when building noun phrases, an N::NNPS will more likely require a rule to delete a French-side determiner, while an N::NN will typically require a determiner in both French and English. Thus, collapsing NN and NNPS may lead to additional ambiguity or incorrect choices when applying larger rules.

Another dimension to be explored is the trade-off between greedy collapsing and other methods that cluster all labels at once. K-means clustering could be a reasonable contrast in this respect; its downside would be that all labels in one language must be assigned to clusters without knowledge of what clusters are being formed in the other language.

Finally, label collapsing is only the first step in a broader exploration of SCFG labeling for MT. We also plan to investigate methods for refining existing category labels in order to find finer-grained subtypes that are useful for translating a particular language pair. By running label collapsing and refining together, our end goal is to be able to adapt standard parser labels to individual translation scenarios.

Acknowledgments

This research was supported in part by U.S. National Science Foundation grants IIS-0713402 and IIS-0915327 and by the DARPA GALE program. Thanks to Chris Dyer for providing the word-aligned and preprocessed FBIS corpus we used in our Chinese–English experiments, and to Jon Clark for suggesting and setting up the hypergraph comparison analysis. We also thank Yahoo! for the use of the M45 research computing cluster, where we ran many steps of our experimental pipeline.

References

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.

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.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.

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

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.

Zhifei Li, 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.

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.

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.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Ventsislav Zhechev 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.

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.