Improved Features and Grammar Selection for Syntax-Based MT

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

Abstract

We present the Carnegie Mellon University Stat-XFER group submission to the WMT 2010 shared translation task. Updates to our syntax-based SMT system mainly fell in the areas of new feature formulations in the translation model and improved filtering of SCFG rules. Compared to our WMT 2009 submission, we report a gain of 1.73 BLEU by using the new features and decoding environment, and a gain of up to 0.52 BLEU from improved grammar selection.

1 Introduction

From its earlier focus on linguistically rich machine translation for resource-poor languages, the statistical transfer MT group at Carnegie Mellon University has expanded in recent years to the increasingly successful domain of syntax-based statistical MT in large-data scenarios. Our submission to the 2010 Workshop on Machine Translation is a syntax-based SMT system with a synchonous context-free grammar (SCFG), where the SCFG rules are derived from full constituency parse trees on both the source and target sides of parallel training sentences. We participated in the French-to-English shared translation task.

This year, we focused our efforts on making more and better use of syntactic grammar. Much of the work went into formulating a more expansive feature set in the translation model and a new method of assigning scores to phrase pairs and grammar rules. Following a change of decoder that allowed us to experiment with systems using much larger syntactic grammars than previously, we also adapted a technique to more intelligently pre-filter grammar rules to those most likely to be useful.

2 System Overview

We built our system on a partial selection of the provided French–English training data, using the Europarl, News Commentary, and UN sets, but ignoring the Giga-FrEn data. After tokenization and some pruning of our training data, this left us with a corpus of approximately 8.6 million sentence pairs. We word-aligned the corpus with MGIZA++ (Gao and Vogel, 2008), a multi-threaded implementation of the standard word alignment tool GIZA++ (Och and Ney, 2003). Word alignments were symmetrized with the "grow-diag-final-and" heuristic. We automatically parsed the French side of the corpus with the Berkeley parser (Petrov and Klein, 2007), while we used the fast vanilla PCFG model of the Stanford parser (Klein and Manning, 2003) for the English side. These steps resulted in a parallel parsed corpus from which to extract phrase pairs and grammar rules.

Phrase extraction involves three distinct steps. In the first, we perform standard (non-syntactic) phrase extraction according to the heuristics of phrase-based SMT (Koehn et al., 2003). In the second, we obtain syntactic phrase pairs using the tree-to-tree matching method of Lavie et al. (2008). Briefly, this method aligns nodes in parallel parse trees by projecting up from the word alignments. A source-tree node s will be aligned to a target-tree node t if the word alignments in the yield of s all land within the yield of t, and vice versa. This node alignment is similar in spirit to the subtree alignment method of Zhechev and Way (2008), except our method is based on the specific Viterbi word alignment links found for each

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 82–87,
Uppsala, Sweden, 15-16 July 2010.

sentence rather than on the general word translation probabilities computed for the corpus as a whole. This enables us to use efficient dynamic programming to infer node alignments, rather than resorting to a greedy search or the enumeration of all possible alignments. Finally, in the third step, we use the node alignments from syntactic phrase pair extraction to extract grammar rules. Each aligned node in a tree pair specifies a decomposition point for breaking the parallel trees into a series of SCFG rules. Like Galley et al. (2006), we allow "composed" (non-minimal) rules when they build entirely on lexical items. However, to control the size of the grammar, we do not produce composed rules that build on other non-terminals, nor do we produce multiple possible rules when we encounter unaligned words. Another difference is that we discard internal structure of composed lexical rules so that we produce SCFG rules rather than synchronous tree substitution grammar rules.

The extracted phrase pairs and grammar rules are collected together and scored according to a variety of features (Section 3). Instead of decoding with the very large complete set of extracted grammar rules, we select only a small number of rules meeting certain criteria (Section 4).

In contrast to previous years, when we used the Stat-XFER decoder, this year we switched to the the Joshua decoder (Li et al., 2009) to take advantage of its more efficient architecture and implementation of modern decoding techniques, such as cube pruning and multi-threading. We also managed system-building workflows with LoonyBin (Clark and Lavie, 2010), a toolkit for managing multi-step experiments across different servers or computing clusters. Section 5 details our experimental results.

3 Translation Model Construction

One major improvement in our system this year is the feature scores we applied to our grammar and phrase pairs. Inspired largely by the Syntax-Augmented MT system (Zollmann and Venugopal, 2006), our translation model contains 22 features in addition to the language model. In contrast to earlier formulations of our features (Hanneman and Lavie, 2009), our maximum-likelihood features are now based on a strict separation between counts drawn from non-syntactic phrase extraction heuristics and our syntactic rule extractor; no feature is estimated from counts in both spaces.

We define an aggregate rule instance as a 5-tuple $r = (L, S, T, C_{phr}, C_{syn})$ that contains a left-hand-side label L, a sequence of terminals and non-terminals for the source (S) and target (T) right-hand sides, and aggregated counts from phrase-based SMT extraction heuristics C_{phr} and the syntactic rule extractor C_{syn}.

In preparation for feature scoring, we:

1. Run phrase instance extraction using standard phrase-based SMT heuristics to obtain tuples $(\mathsf{PHR}, S, T, C_{phr}, \emptyset)$ where S and T never contain non-terminals
2. Run syntactic rule instance extraction as described in Section 2 above to obtain tuples $(L, S, T, \emptyset, C_{syn})$
3. Share non-syntactic counts such that, for any two tuples $r_1 = (\mathsf{PHR}, S, T, C_{phr}, \emptyset)$ and $r_2 = (L_2, S, T, \emptyset, C_{syn})$ with equivalent S and T values, we produce $r_2 = (L_2, S, T, C_{phr}, C_{syn})$

Note that there is no longer any need to retain PHR rules (PHR, S, T) that have syntactic equivalents $(L \neq \mathsf{PHR}, S, T)$ since they have the same features In addition, we assume there will be no tuples where S and T contain non-terminals while $C_{phr} = 0$ and $C_{syn} > 0$. That is, the syntactic phrases are a subset of non-syntactic phrases.

3.1 Maximum-Likelihood Features

Our most traditional features are $P_{phr}(T \mid S)$ and $P_{phr}(S \mid T)$, estimated using only counts C_{phr}. These features apply only to rules not containing any non-terminals. They are equivalent to the phrase $P(T \mid S)$ and $P(S \mid T)$ features from the Moses decoder, even when $L \neq$ PHR. In contrast, we used $P_{syn \cup phr}(L, S \mid T)$ and $P_{syn \cup phr}(L, T \mid S)$ last year, which applied to all rules. The new features are no longer subject to increased sparsity as the number of non-terminals in the grammar increases.

We also have grammar rule probabilities $P_{syn}(T \mid S)$, $P_{syn}(S \mid T)$, $P_{syn}(L \mid S)$, $P_{syn}(L \mid T)$, and $P_{syn}(L \mid S, T)$ estimated using C_{syn}; these apply only to rules where S and T contain non-terminals. By no longer including counts from phrase-based SMT extraction heuristics in these features, we encourage rules where $L \neq \mathsf{PHR}$ since the smaller counts from the rule learner would have otherwise been overshadowed by the much larger counts from the phrase-based SMT heuristics.

Finally, we estimate "not labelable" (NL) features $P_{syn}(\text{NL} \mid S)$ and $P_{syn}(\text{NL} \mid T)$. With R denoting the set of all extracted rules,

$$P_{syn}(\text{NL} \mid S) = \frac{C_{syn}}{\sum_{r' \in R \text{ s.t. } S'=S} C'_{syn}} \quad (1)$$

$$P_{syn}(\text{NL} \mid T) = \frac{C_{syn}}{\sum_{r' \in R \text{ s.t. } T'=T} C'_{syn}} \quad (2)$$

We use additive smoothing (with $n = 1$ for our experiments) to avoid a probability of 0 when there is no syntactic label for an (S, T) pair. These features can encourage syntactic rules when syntax is likely given a particular string since probability mass is often distributed among several different syntactic labels.

3.2 Instance Features

We add several features that use sufficient statistics local to each rule. First, we add three binary low-count features that take on the value 1 when the frequency of the rule is exactly 1, 2, or 3. There are also two indicator features related to syntax: one each that fires when $L = \text{PHR}$ and when $L \neq \text{PHR}$. Other indicator features analyze the abstractness of grammar rules: $A_S = 1$ when the source side contains only non-terminals, $A_T = 1$ when the target side contains only non-terminals, TGTINSERTION $= 1$ when $A_S = 1, A_T = 0$, SRCDELETION $= 1$ when $A_S = 0, A_T = 1$, and INTERLEAVED $= 1$ when $A_S = 0, A_T = 0$.

Bidirectional lexical probabilities for each rule are calculated from a unigram lexicon MLE-estimated over aligned word pairs in the training corpus, as is the default in Moses.

Finally, we include a glue rule indicator feature that fires whenever a glue rule is applied during decoding. In the Joshua decoder, these monotonic rules stitch syntactic parse fragments together at no model cost.

4 Grammar Selection

With extracted grammars typically reaching tens of millions of unique rules — not to mention phrase pairs — our systems clearly face an engineering challenge when attempting to include the full grammar at decoding time. Iglesias et al. (2009) classified SCFG rules according to the pattern of terminals and non-terminals on the rules' right-hand sides, and found that certain patterns could be entirely left out of the grammar without loss of MT quality. In particular, large classes of monotonic rules could be removed without a loss in automatic metric scores, while small classes of reordering rules contributed much more to the success of the system. Inspired by that approach, we passed our full set of extracted grammar rule instances through a filter after scoring. Using the rule notation from Section 3, the filter retained only those rules that matched one of the following patterns:

$$\begin{aligned} S = X^1\, w, &\quad T = w\, X^1 \\ S = w\, X^1, &\quad T = X^1\, w \\ S = X^1\, X^2, &\quad T = X^2\, X^1 \\ S = X^1\, X^2, &\quad T = X^1\, X^2 \end{aligned}$$

where X represents any non-terminal and w represents any span of one or more terminals. The choice of the specific reordering patterns above captures our intuition that binary swaps are a fundamental ordering divergence between languages, while the inclusion of the abstract monotonic pattern $(X^1\, X^2, X^1\, X^2)$ ensures that the decoder is not disproportionately biased towards applying reordering rules without supporting lexical evidence merely because in-order rules are left out.

Orthogonally to the pattern-based pruning, we also selected grammars by sorting grammar rules in decreasing order of frequency count and using the top n in the decoder. We experimented with $n = 0$, 100, 1000, and 10,000. In all cases of grammar selection, we disallowed rules that inserted unaligned target-side terminals unless the inserted terminals were among the top 100 most frequent unigrams in the target-side vocabulary.

5 Results and Analysis

5.1 Comparison with WMT 2009 Results

We performed our initial development work on an updated version of our previous WMT submission (Hanneman et al., 2009) so that the effects of our changes could be directly compared. Our 2009 system was trained from the full Europarl and News Commentary data available that year, plus the pre-release version of the Giga-FrEn data, for a total of 9.4 million sentence pairs. We used the news-dev2009a set for minimum error-rate training and tested system performance on news-dev2009b. To maintain continuity with our previously reported scores, we report new scores here using the same training, tuning, and testing sets, using the uncased versions of IBM-style

System Configuration	METEOR	BLEU
1. WMT ’09 submission	0.5263	0.2073
2. Joshua decoder	0.5231	0.2158
3. New TM features	0.5348	0.2246

Table 1: Dev test results (on news-dev2009b) from our WMT 2009 system when updating decoding environment and feature formulations.

System Configuration	METEOR	BLEU
1. n = 100	0.5314	0.2200
2. n = 100, filtered	0.5341	0.2242
3. n = 1000	0.5324	0.2206
4. n = 1000, filtered	0.5330	0.2233
5. n = 10,000	0.5332	0.2198
6. n = 10,000, filtered	0.5350	0.2250

Table 2: Dev test results (on news-dev2009b) from our WMT 2009 system with and without pattern-based grammar selection.

BLEU 1.04 (Papineni et al., 2002) and METEOR 0.6 (Lavie and Agarwal, 2007).

Table 1 shows the effect of our new scoring and decoding environment. Line 2 uses the same extracted phrase pairs and grammar rules as line 1, but the system is tuned and tested with the Joshua decoder instead of Stat-XFER. For line 3, we rescored the extracted phrase pairs from lines 1 and 2 using the updated features discussed in Section 3.[1] The difference in automatic metric scores shows a significant benefit from both the new decoder and the updated feature formulations: 0.8 BLEU points from the change in decoder, and 0.9 BLEU points from the expanded set of 22 translation model features.

Our next test was to examine the usefulness of the pattern-based grammar selection described in Section 4. For various numbers of rules n, Table 2 shows the scores obtained with and without filtering the grammar before the n most frequent rules are skimmed off for use. We observe a small but consistent gain in scores from the grammar selection process, up to half a BLEU point in the largest-grammar systems (lines 5 and 6).

Source	Target
un rôle AP^1	ADJP^1 roles
l’ instabilité AP^1	ADJP^1 instability
l’ argent PP^1	NP^1 money
une pression AP^1	ADJP^1 pressure
la gouvernance AP^1	ADJP^1 governance
la concurrence AP^1	ADJP^1 competition
des preuves AP^1	ADJP^1 evidence
les outils AP^1	ADJP^1 tools
des changements AP^1	ADJP^1 changes

Table 3: Rules fitting the pattern $(S = w\ X^1, T = X^1\ w)$ that applied on the news-test2010 test set.

5.2 WMT 2010 Results and Analysis

We built the WMT 2010 version of our system from the training data described in Section 2. (The system falls under the strictly constrained track: we used neither the Giga-FrEn data for training nor the LDC Gigaword corpora for language modeling.) We used the provided news-test2008 set for system tuning, while news-test2009 served as our 2010 dev test set. Based on the results in Table 2, our official submission to this year’s shared task was constructed as in line 6, with 10,000 syntactic grammar rules chosen after a pattern-based grammar selection step. On the news-test2010 test set, this system scored 0.2327 on case-insensitive IBM-style BLEU 1.04, 0.5614 on METEOR 0.6, and 0.5519 on METEOR 1.0 (Lavie and Denkowski, 2009).

The actual application of grammar rules in the system is quite surprising. Despite having a grammar of 10,000 rules at its disposal, the decoder chose to only apply a total of 20 unique rules in 392 application instances in the 2489-sentence news-test2010 set. On a per-sentence basis, this is actually *fewer* rule applications than our system performed last year with a 26-rule handpicked grammar! The most frequently applied rules are fully abstract, monotonic structure-building rules, such as for stitching together compound noun phrases with adverbial phrases or prepositional phrases. Nine of the 20 rules, listed in Table 3, demonstrate the effect of our pattern-based grammar selection. These partially lexicalized rules fit the pattern $(S = w\ X^1, T = X^1\ w)$ and handle cases of lexicalized binary reordering between French and English. Though the overall impact of these rules on automatic metric scores is presum-

[1] In line 2, we did not control for difference in formulation of the translation length feature: Stat-XFER uses a length ratio, while Joshua uses a target word count. Line 3 does not include 26 manually selected grammar rules present in lines 1 and 2; this is because our new feature scoring requires information from the grammar rules that was not present in our 2009 extracted resources.

ably quite small, we believe that the key to effective syntactic grammars in our MT approach lies in retaining precise rules of this type for common linguistically motivated reordering patterns.

The above pattern of rule applications is also observed in our dev test set, news-test2009, where 16 distinct rules apply a total of 352 times. Seven of the fully abstract rules and three of the lexicalized rules that applied on news-test2009 also applied on news-test2010, while a further two abstract and four lexicalized rules applied on news-test2009 alone. We thus have a general trend of a set of general rules applying with higher frequency across test sets, while the set of lexicalized rules used varies according to the particular set.

Since, overall, we still do not see as much grammar application in our systems as we would like, we plan to concentrate future work on further improving this aspect. This includes a more detailed study of grammar filtering or refinement to select the most useful rules. We would also like to explore the effect of the features of Section 3 individually, on different language pairs, and using different grammar types.

Acknowledgments

This research was supported in part by NSF grant IIS-0534217 (LETRAS) and the DARPA GALE program. We thank Yahoo! for the use of the M45 research computing cluster, where we ran many steps of our experimental pipeline.

References

Jonathan Clark and Alon Lavie. 2010. LoonyBin: Keeping language technologists sane through automated management of experimental (hyper)workflows. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC '10)*, Valletta, Malta, May.

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

Qin Gao 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.

Greg Hanneman 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 Translations*, pages 1–9, Boulder, CO, June.

Greg Hanneman, Vamshi Ambati, Jonathan H. Clark, Alok Parlikar, and Alon Lavie. 2009. An improved statistical transfer systems for French–English machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 140–144, Athens, Greece, March.

Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009. Rule filtering by pattern for efficient hierarchical translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 380–388, Athens, Greece, March–April.

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

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

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June.

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.

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.

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.