Syntax Augmented Machine Translation via Chart Parsing

Andreas Zollmann and **Ashish Venugopal**
School of Computer Science
Carnegie Mellon University
{zollmann,ashishv}@cs.cmu.edu

Abstract

We present translation results on the shared task "Exploiting Parallel Texts for Statistical Machine Translation" generated by a chart parsing decoder operating on phrase tables augmented and generalized with target language syntactic categories. We use a target language parser to generate parse trees for each sentence on the target side of the bilingual training corpus, matching them with phrase table lattices built for the corresponding source sentence. Considering phrases that correspond to syntactic categories in the parse trees we develop techniques to augment (declare a syntactically motivated category for a phrase pair) and generalize (form mixed terminal and nonterminal phrases) the phrase table into a synchronous bilingual grammar. We present results on the French-to-English task for this workshop, representing significant improvements over the workshop's baseline system. Our translation system is available open-source under the GNU General Public License.

1 Introduction

Recent work in machine translation has evolved from the traditional word (Brown et al., 1993) and phrase based (Koehn et al., 2003a) models to include hierarchical phrase models (Chiang, 2005) and bilingual synchronous grammars (Melamed, 2004). These advances are motivated by the desire to integrate richer knowledge sources within the translation process with the explicit goal of producing more fluent translations in the target language. The hierarchical translation operations introduced in these methods call for extensions to the traditional beam decoder (Koehn et al., 2003a). In this work we introduce techniques to generate syntactically motivated generalized phrases and discuss issues in chart parser based decoding in the statistical machine translation environment.

(Chiang, 2005) generates synchronous context-free grammar (SynCFG) rules from an existing phrase translation table. These rules can be viewed as phrase pairs with mixed lexical and non-terminal entries, where non-terminal entries (occurring as pairs in the source and target side) represent placeholders for inserting additional phrases pairs (which again may contain nonterminals) at decoding time. While (Chiang, 2005) uses only two nonterminal symbols in his grammar, we introduce multiple syntactic categories, taking advantage of a target language parser for this information. While (Yamada and Knight, 2002) represent syntactical information in the decoding process through a series of transformation operations, we operate directly at the phrase level. In addition to the benefits that come from a more structured hierarchical rule set, we believe that these restrictions serve as a syntax driven language model that can guide the decoding process, as n-gram context based language models do in traditional decoding. In the following sections, we describe our phrase annotation and generalization process followed by the design and pruning decisions in our chart parser. We give results on the French-English Europarl data and conclude with prospects for future work.

2 Rule Generation

We start with phrase translations on the parallel training data using the techniques and implementation described in (Koehn et al., 2003a). This phrase table provides the purely lexical entries in the final hierarchical rule set that will be used in decoding. We then use Charniak's parser (Charniak, 2000) to generate the most likely parse tree for each English target sentence in the training corpus. Next, we determine all phrase pairs in the phrase table whose source and target side occur in each respective source and target sentence pair defining the scope of the initial rules in our SynCFG.

Annotation If the target side of any of these initial rules correspond to a syntactic category C of the target side parse tree, we label the phrase pair with that syntactic category. This label corresponds to the left-hand side of our synchronous grammar. Phrase pairs that do not correspond to a span in the parse tree are given a default category "X", and can still play a role in the decoding process. In work done after submission to the 2006 data track, we assign such phrases an extended category of the form $C_1 + C_2$, C_1/C_2, or $C_2\backslash C_1$, indicating that the phrase pair's target side spans two adjacent syntactic categories (e.g., *she went: NP+V*), a partial syntactic category C_1 missing a C_2 to the right (e.g., *the great: NP/NN*), or a partial C_1 missing a C_2 to the left (e.g., *great wall: DT\NP*), respectively.

Generalization In order to mitigate the effects of sparse data when working with phrase and n-gram models we would like to generate generalized phrases, which include non-terminal symbols that can be filled with other phrases. Therefore, after annotating the initial rules from the current training sentence pair, we adhere to (Chiang, 2005) to recursively generalize each existing rule; however, we abstract on a per-sentence basis. The grammar extracted from this evaluation's training data contains 75 nonterminals in our standard system, and 4000 nonterminals in the extended-category system. Figure 1 illustrates the annotation and generalization process.

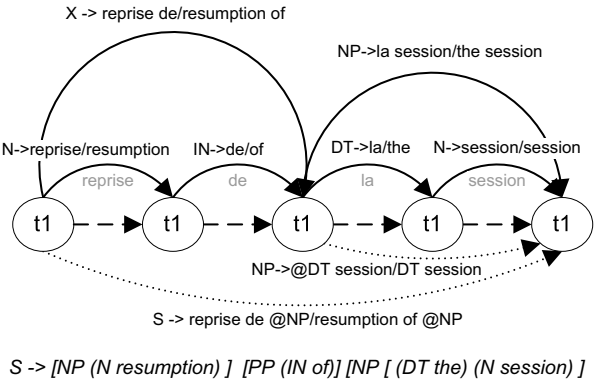

Figure 1: *Selected annotated and generalized (dotted arc) rules for the first sentence of Europarl.*

3 Scoring

We employ a log-linear model to assign costs to the SynCFG. Given a source sentence f, the preferred translation output is determined by computing the lowest-cost derivation (combination of hierarchical and glue rules) yielding f as its source side, where the cost of a derivation $R_1 \circ \cdots \circ R_n$ with respective feature vectors $v^1, \ldots, v^n \in \mathbb{R}^m$ is given by

$$\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n} (v^j)_i \ .$$

Here, $\lambda_1, \ldots, \lambda_m$ are the parameters of the log-linear model, which we optimize on a held-out portion of the training set (2005 development data) using minimum-error-rate training (Och, 2003). We use the following features for our rules:

- source- and target-conditioned neg-log lexical weights as described in (Koehn et al., 2003b)
- neg-log relative frequencies: left-hand-side-conditioned, target-phrase-conditioned, source-phrase-conditioned
- Counters: n.o. rule applications, n.o. target words
- Flags: IsPurelyLexical (i.e., contains only terminals), IsPurelyAbstract (i.e., contains only nonterminals), IsXRule (i.e., non-syntactical span), IsGlueRule

- Penalties: rareness penalty $\exp(1 - \textit{RuleFrequency})$; unbalancedness penalty $|\textit{MeanTargetSourceRatio} * \textit{`n.o. source words'} - \textit{`n.o. target words'}|$

4 Parsing

Our SynCFG rules are equivalent to a probabilistic context-free grammar and decoding is therefore an application of chart parsing. Instead of the common method of converting the CFG grammar into Chomsky Normal Form and applying a CKY algorithm to produce the most likely parse for a given source sentence, we avoided the explosion of the rule set caused by the introduction of new non-terminals in the conversion process and implemented a variant of the CKY+ algorithm as described in (J.Earley, 1970).

Each cell of the parsing process in (J.Earley, 1970) contains a set of hypergraph nodes (Huang and Chiang, 2005). A hypergraph node is an equivalence class of complete hypotheses (derivations) with identical production results (left-hand sides of the corresponding applied rules). Complete hypotheses point directly to nodes in their backwards star, and the cost of the complete hypothesis is calculated with respect to each back pointer node's best cost.

This structure affords efficient parsing with minimal pruning (we use a single parameter to restrict the number of hierarchical rules applied), but sacrifices effective management of unique language model states contributing to significant search errors during parsing. At initial submission time we simply re-scored a K-Best list extracted after first best parsing using the lazy retrieval process in (Huang and Chiang, 2005).

Post-submission After our workshop submission, we modified the K-Best list extraction process to integrate an n-gram language model during K-Best extraction. Instead of expanding each derivation (complete hypothesis) in a breadth-first fashion, we expand only a single back pointer, and score this new derivation with its translation model scores and a language model cost estimate, consisting of an accurate component, based on the words translated so far, and an estimate based on each remaining (not expanded) back pointer's top scoring hypothesis.

To improve the diversity of the final K-Best list, we keep track of partially expanded hypotheses that have generated identical target words and refer to the same hypergraph nodes. Any arising twin hypothesis is immediately removed from the K-Best extraction beam during the expansion process.

5 Results

We present results that compare our system against the baseline Pharaoh implementation (Koehn et al., 2003a) and MER training scripts provided for this workshop. Our results represent work done before the submission due date as well as after with the following generalized phrase systems.

- Baseline - Pharaoh with phrases extracted from IBM Model 4 training with maximum phrase length 7 and extraction method 'diag-growth-final' (Koehn et al., 2003a)
- Lex - Phrase-decoder simulation: using only the initial lexical rules from the phrase table, all with LHS X, the Glue rule, and a binary reordering rule with its own reordering-feature
- XCat - All nonterminals merged into a single X nonterminal: simulation of the system Hiero (Chiang, 2005).
- Syn - Syntactic extraction using the Penn Treebank parse categories as nonterminals; rules containing up to 4 nonterminal abstraction sites.
- SynExt - Syntactic extraction using the extended-category scheme, but with rules only containing up to 2 nonterminal abstraction sites.

We also explored the impact of longer initial phrases by training another phrase table with phrases up to length 12. Our results are presented in Table 1. While our submission time system (Syn using LM for rescoring only) shows no improvement over the baseline, we clearly see the impact of integrating the language model into the K-Best list extraction process. Our final system shows at statistically significant improvement over the baseline (0.78 BLEU points is the 95 confidence level). We also see a trend towards improving translation quality as we

System	Dev: w/o LM	Dev: LM-rescoring	Test: LM-r.	Dev: integrated LM	Test: int. LM
Baseline - max. phr. length 7	–	–	–	31.11	30.61
Lex - max. phrase length 7	27.94	29.39	29.95	28.96	29.12
XCat - max. phrase length 7	27.56	30.27	29.81	30.89	31.01
Syn - max. phrase length 7	29.20	**30.95**	**30.58**	31.52	31.31
SynExt - max. phrase length 7	–	–	–	31.73	31.41
Baseline - max. phr. length 12	–	–	–	31.16	30.90
Lex - max. phr. length 12	–	–	–	29.30	29.51
XCat - max. phr. length 12	–	–	–	30.79	30.59
SynExt - max. phr. length 12	–	–	–	31.07	31.76

Table 1: *Translation results (IBM BLEU) for each system on the Fr-En '06 Shared Task 'Development Set' (used for MER parameter tuning) and '06 'Development Test Set' (identical to last year's Shared Task's test set). The system submitted for evaluation is highlighted in bold.*

employ richer extraction techniques. The relatively poor performance of Lex with LM in K-Best compared to the baseline shows that we are still making search errors during parsing despite tighter integration of the language model.

We also ran an experiment with CMU's phrase-based decoder (Vogel et al., 2003) using the length-7 phrase table. While its development-set score was only 31.01, the decoder achieved 31.42 on the test set, placing it at the same level as our extended-category system for that phrase table.

6 Conclusions

In this work we applied syntax based resources (the target language parser) to annotate and generalize phrase translation tables extracted via existing phrase extraction techniques. Our work reaffirms the feasibility of parsing approaches to machine translation in a large data setting, and illustrates the impact of adding syntactic categories to drive and constrain the structured search space. While no improvements were available at submission time, our subsequent performance highlights the importance of tight integration of n-gram language modeling within the syntax driven parsing environment. Our translation system is available open-source under the GNU General Public License at: `www.cs.cmu.edu/~zollmann/samt`

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Eugene Charniak. 2000. A maximum entropy-inspired parser. In *Proceedings of the North American Association for Computational Linguistics (HLT/NAACL)*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of the Association for Computational Linguistics*.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies*.

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

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003a. Pharaoh: A beam search decoder for phrase-base statistical machine translation models. In *Proceedings of the Sixth Confernence of the Association for Machine Translation in the Americas*, Edomonton, Canada, May 27-June 1.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003b. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *ACL*, pages 653–660.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical translation system. In *Proceedings of MT Summit IX*, New Orleans, LA, September.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical mt. In *Proc. of the Association for Computational Linguistics*.