The Syntax Augmented MT (SAMT) System for the Shared Task in the 2007 ACL Workshop on Statistical Machine Translation

Andreas Zollmann and **Ashish Venugopal** and **Matthias Paulik** and **Stephan Vogel**
School of Computer Science, Carnegie Mellon University, Pittsburgh
interACT Lab, University of Karlsruhe
{ashishv,zollmann,paulik,vogel+}@cs.cmu.edu

Abstract

We describe the CMU-UKA Syntax Augmented Machine Translation system 'SAMT' used for the shared task "Machine Translation for European Languages" at the ACL 2007 Workshop on Statistical Machine Translation. Following an overview of syntax augmented machine translation, we describe parameters for components in our open-source SAMT toolkit that were used to generate translation results for the Spanish to English in-domain track of the shared task and discuss relative performance against our phrase-based submission.

1 Introduction

As Chiang (2005) and Koehn et al. (2003) note, purely lexical "phrase-based" translation models suffer from sparse data effects when translating conceptual elements that span or skip across several source language words. Phrase-based models also rely on distance and lexical distortion models to represent the reordering effects across language pairs. However, such models are typically applied over limited source sentence ranges to prevent errors introduced by these models and to maintain efficient decoding (Och and Ney, 2004).

To address these concerns, hierarchically structured models as in Chiang (2005) define weighted transduction **rules**, interpretable as components of a probabilistic synchronous grammar (Aho and Ullman, 1969) that represent translation and reordering operations. In this work, we describe results from the open-source Syntax Augmented Machine Translation (SAMT) toolkit (Zollmann and Venugopal, 2006) applied to the Spanish-to-English in-domain translation task of the ACL'07 workshop on statistical machine translation.

We begin by describing the probabilistic model of translation applied by the SAMT toolkit. We then present settings for the pipeline of SAMT tools that we used in our shared task submission. Finally, we compare our translation results to the CMU-UKA phrase-based SMT system and discuss relative performance.

2 Synchronous Grammars for SMT

Probabilistic synchronous context-free grammars (PSCFGs) are defined by a source terminal set (source vocabulary) $\mathcal{T}_S$, a target terminal set (target vocabulary) $\mathcal{T}_T$, a shared nonterminal set $\mathcal{N}$ and production rules of the form

$$X \rightarrow \langle \gamma, \alpha, \sim, w \rangle$$

where following (Chiang, 2005)

- $X \in \mathcal{N}$ is a nonterminal
- $\gamma \in (\mathcal{N} \cup \mathcal{T}_S)^*$: sequence of source nonterminals and terminals
- $\alpha \in (\mathcal{N} \cup \mathcal{T}_T)^*$: sequence of target nonterminals and terminals
- the count $\#\mathrm{NT}(\gamma)$ of nonterminal tokens in γ is equal to the count $\#\mathrm{NT}(\alpha)$ of nonterminal tokens in α,
- $\sim$: $\{1, \ldots, \#\mathrm{NT}(\gamma)\} \rightarrow \{1, \ldots, \#\mathrm{NT}(\alpha)\}$ one-to-one mapping from nonterminal tokens in γ to nonterminal tokens in α
- $w \in [0, \infty)$: nonnegative real-valued weight

Chiang (2005) uses a single nonterminal category, Galley et al. (2004) use syntactic constituents for the PSCFG nonterminal set, and Zollmann and Venugopal (2006) take advantage of CCG (Combinatorial Categorical Grammar) (Steedman, 1999) inspired "slash" and "plus" categories, focusing on target (rather than source side) categories to generate well formed translations.

We now describe the identification and estimation of PSCFG rules from parallel sentence aligned corpora under the framework proposed by Zollmann and Venugopal (2006).

Proceedings of the Second Workshop on Statistical Machine Translation, pages 216–219,
Prague, June 2007.

2.1 Grammar Induction

Zollmann and Venugopal (2006) describe a process to generate a PSCFG given parallel sentence pairs $\langle f, e \rangle$, a parse tree π for each e, the maximum *a posteriori* word alignment a over $\langle f, e \rangle$, and phrase pairs $Phrases(a)$ identified by any alignment-driven phrase induction technique such as e.g. (Och and Ney, 2004).

Each phrase in $Phrases(a)$ (phrases identifiable from a) is first annotated with a syntactic category to produce initial **rules**. If the target span of the phrase does not match a constituent in π, heuristics are used to assign categories that correspond to partial rewriting of the tree. These heuristics first consider concatenation operations, forming categories like “NP+VP”, and then resort to CCG style “slash” categories like “NP/NN” giving preference to categories found closer to the leaves of the tree.

To illustrate this process, consider the following French-English sentence pair and selected phrase pairs obtained by phrase induction on an automatically produced alignment a, and matching target spans with π.

$$\begin{array}{rcl} f & = & \text{il ne va pas} \\ e & = & \text{he does not go} \\ \text{PRP} & \rightarrow & \text{il, he} \\ \text{VB} & \rightarrow & \text{va, go} \\ \text{RB+VB} & \rightarrow & \text{ne va pas, not go} \\ \text{S} & \rightarrow & \text{il ne va pas, he does not go} \end{array}$$

The alignment a with the associated target side parse tree is shown in Fig. 1 in the alignment visualization style defined by Galley et al. (2004).

Following the Data-Oriented Parsing inspired rule generalization technique proposed by Chiang (2005), one can now generalize each **identified** rule (initial or already partially generalized) $N \rightarrow f_1 \ldots f_m / e_1 \ldots e_n$ for which there is an **initial** rule $M \rightarrow f_i \ldots f_u / e_j \ldots e_v$ where $1 \leq i < u \leq m$ and $1 \leq j < v \leq n$, to obtain a new rule

$$N \rightarrow f_1 \ldots f_{i-1} M_k f_{u+1} \ldots f_m / e_1 \ldots e_{j-1} M_k e_{v+1} \ldots e_n$$

where k is an index for the nonterminal M that indicates the one-to-one correspondence between the new M tokens on the two sides (it is not in the space of word indices like i, j, u, v, m, n). The initial rules listed above can be generalized to additionally extract the following rules from f, e.

$$\begin{array}{rcl} \text{S} & \rightarrow & \text{PRP}_1 \text{ ne va pas , PRP}_1 \text{ does not go} \\ \text{S} & \rightarrow & \text{il ne VB}_1 \text{ pas , he does not VB}_1 \\ \text{S} & \rightarrow & \text{il RB+VB}_1\text{, he does RB+VB}_1 \\ \text{S} & \rightarrow & \text{PRP}_1 \text{ RB+VB}_2\text{, PRP}_1 \text{ does RB+VB}_2 \\ \text{RB+VB} & \rightarrow & \text{ne VB}_1 \text{ pas , not VB}_1 \end{array}$$

Fig. 2 uses regions to identify the labeled, source and target side span for all initial rules extracted on our example sentence pair and parse. Under this representation, generalization can be viewed as a process that selects a region, and proceeds to subtract out any sub-region to form a generalized rule.

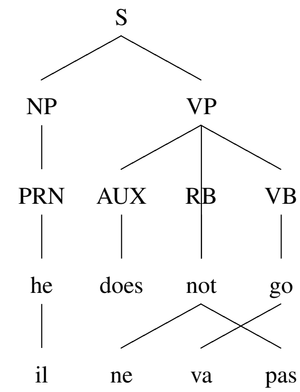

Figure 1: Alignment graph (word alignment and target parse tree) for a French-English sentence pair.

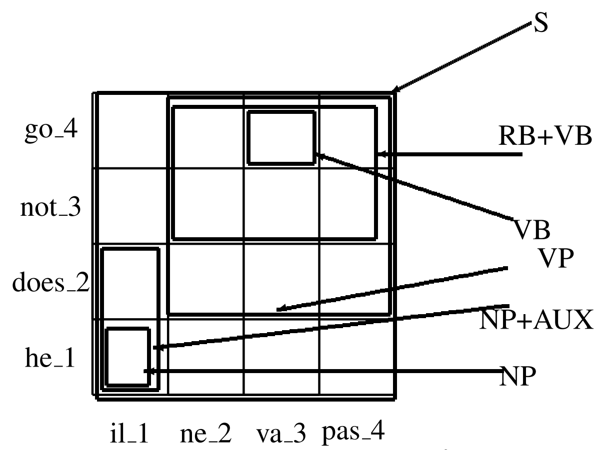

Figure 2: Spans of initial lexical phrases w.r.t. f, e. Each phrase is labeled with a category derived from the tree in Fig. 1.

2.2 Decoding

Given a source sentence f, the translation task under a PSCFG grammar can be expressed analogously to monolingual parsing with a CFG. We find the most likely derivation D with source-side f and read off the English translation from this derivation:

$$\hat{e} = \text{tgt}\left(\underset{D:\text{src}(D)=f}{\arg\max}\; p(D) \right) \tag{1}$$

where $\text{tgt}(D)$ refers to the target terminals and $\text{src}(D)$ to the source terminals generated by derivation D.

Our distribution p over derivations is defined by a log-linear model. The probability of a derivation D

is defined in terms of the rules r that are used in D:

$$p(D) = \frac{p_{LM}(\mathrm{tgt}(D))^{\lambda_{\mathrm{LM}}} \prod_{r \in D} \prod_i \phi_i(r)^{\lambda_i}}{Z(\lambda)} \quad (2)$$

where ϕ_i refers to features defined on each rule, p_{LM} is a language model (LM) probability applied to the target terminal symbols generated by the derivation D, and $Z(\lambda)$ is a normalization constant chosen such that the probabilities sum up to one. The computational challenges of this search task (compounded by the integration of the LM) are addressed in (Chiang, 2007; Venugopal et al., 2007). The feature weights λ_i are trained in concert with the LM weight via minimum error rate (MER) training (Och, 2003).

We now describe the parameters for the SAMT implementation of the model described above.

3 SAMT Components

SAMT provides tools to perform grammar induction (“extractrules”, “filterrules”), from bilingual phrase pairs and target language parse trees, as well as translation (“FastTranslateChart”) of source sentences given an induced grammar.

3.1 extractrules

extractrules is the first step of the grammar induction pipeline, where rules are identified based on the process described in section 2.1. This tool works on a per sentence basis, considering phrases extracted for the training sentence pair $\langle s_i, t_i \rangle$ and the corresponding target parse tree π_i. **extractrules** outputs identified rules for each input sentence pair, along with associated statistics that play a role in the estimation of the rule features ϕ. These statistics take the form of real-valued feature vectors for each rule as well as summary information collected over the corpus, such as the frequency of each nonterminal symbol, or unique rule source sides encountered.

For the shared task evaluation, we ran **extractrules** with the following extraction parameter settings to limit the scope and number of rules extracted. These settings produce the same initial phrase table as the CMU-UKA phrase based system. We limit the source-side length of the phrase pairs considered as initial rules to 8 (parameter `MaxSourceLength`). Further we set the maximum number of source and target terminals per rule (`MaxSource`/`MaxTargetWordCount`) to 5 and 8 respectively with 2 of nonterminal pairs (i.e., substitution sites) per rule (`MaxSubstititionCount`). We limit the total number of symbols in each rule to 8 (`MaxSource`/`TargetSymbolCount`) and require all rules to contain at least one source-side terminal symbol (`noAllowAbstractRules`, `noAllowRulesWithOnlyTargetTerminals`) since this reduces decoding time considerably. Additionally, we discard all rules that contain source word sequences that do not exist in the development and test sets provided for the shared task (parameter `-r`).

3.2 filterrules

This tool takes as input the rules identified by **extractrules**, and associates each rule with a feature vector ϕ, representing multiple criteria by which the decoding process can judge the quality of each rule and, by extension, each derivation. **filterrules** is also in charge of pruning the resulting PSCFG to ensure tractable decoding.

ϕ contains both real and Boolean valued features for each rule. The following probabilistic features are generated by **filterrules**:

- $\hat{p}(r|\,\mathrm{lhs}(X))$: Probability of a rule given its left-hand-side (“result”) nonterminal
- $\hat{p}(r|\,\mathrm{src}(r))$: Prob. of a rule given its source side
- $\hat{p}(\mathrm{ul}(\mathrm{src}(r)), \mathrm{ul}(\mathrm{tgt}(r))|\,\mathrm{ul}(\mathrm{src}(r))$: Probability of the unlabeled source and target side of the rule given its unlabeled source side.

Here, the function ul removes all syntactic labels from its arguments, but retains ordering notation, producing relative frequencies similar to those used in purely hierarchical systems. As in phrase-based translation model estimation, ϕ also contains two lexical weights (Koehn et al., 2003), counters for number of target terminals generated. ϕ also boolean features that describe rule types (i.e. purely terminal vs purely nonterminal).

For the shared task submission, we pruned away rules that share the same source side based on $\hat{p}(r|\,\mathrm{src}(r))$ (the source conditioned relative frequency). We prune away a rule if this value is less that 0.5 times the one of the best performing rule (parameters `BeamFactorLexicalRules`, `BeamFactorNonlexicalRules`).

3.3 FastTranslateChart

The **FastTranslateChart** decoder is a chart parser based on the CYK+(Chappelier and Rajman, 1998) algorithm. Translation experiments in this paper are performed with a 4-gram SRI language model trained on the target side of the corpus. **FastTranslateChart** implements both methods of handling the LM intersection described in (Venugopal et al., 2007). For this submission, we use the Cube-Pruning (Chiang, 2007) approach (the default setting). LM and rule feature parameters λ are trained with the included MER training tool. Our pruning settings allow up to 200 chart items per cell with left-hand side nonterminal '_S' (the reserved sentence spanning nonterminal), and 100 items per cell for each other nonterminal. Beam pruning based on an (LM-scaled) additive beam of neglob probability 5 is used to prune the search further. These pruning settings correspond to setting `'PruningMap=0-100-5-@_S-200-5'`.

4 Empirical Results

We trained our system on the Spanish-English in-domain training data provided for the workshop. Initial data processing and normalizing is described in the workshop paper for the CMU-UKA ISL phrase-based system. NIST-BLEU scores are reported on the 2K sentence development 'dev06' and test 'test06' corpora as per the workshop guidelines (case sensitive, de-tokenized). We compare our scores against the CMU-UKA ISL phrase-based submission, a state-of-the art phrase-based SMT system with part-of-speech (POS) based word reordering (Paulik et al., 2007).

4.1 Translation Results

The SAMT system achieves a BLEU score of 32.48% on the 'dev06' development corpus and 32.15% on the unseen 'test06' corpus. This is slightly better than the score of the CMU-UKA phrase-based system, which achieves 32.20% and 31.85% when trained and tuned under the same in-domain conditions. [1]

To understand why the syntax augmented approach has limited additional impact on the Spanish-to-English task, we consider the impact of reordering within our phrase-based system. Table 1 shows the impact of increasing reordering window length (Koehn et al., 2003) on translation quality for the 'dev06' data.[2] Increasing the reordering window past 2 has minimal impact on translation quality, implying that most of the reordering effects across Spanish and English are well modeled at the local or phrase level. The benefit of syntax-based systems to capture long-distance reordering phenomena based on syntactic structure seems to be of limited value for the Spanish to English translation task.

ReOrder	1	2	3	4	POS	**SAMT**
BLEU	31.98	32.24	32.30	32.26	32.20	**32.48**

Table 1: Impact of phrase based reordering model settings compared to SAMT on the 'dev06' corpus measured by NIST-BLEU

5 Conclusions

In this work, we briefly summarized the Syntax-augmented MT model, described how we trained and ran our implementation of that model on the MT'07 Spanish-to-English translation task. We compared SAMT translation results to a strong phrase-based system trained under the same conditions. Our system is available open-source under the GNU General Public License (GPL) and can be downloaded at `www.cs.cmu.edu/~zollmann/samt`

[1] The CMU-UKA phrase-based workshop submission was tuned on out-of-domain data as well.

[2] Variant of the CMU-UKA ISL phrase-based system without POS based reordering. With POS-based reordering turned on, additional window-based reordering even for window length 1 had no improvement in NIST-BLEU.

References

Alfred Aho and Jeffrey Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*.

Jean-Cedric. Chappelier and Martin Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proc. of Tabulation in Parsing and Deduction (TAPD'98)*, Paris, France.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*.

David Chiang. 2007. Hierarchical phrase based translation. *Computational Linguistics*. To appear.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. of HLT/NAACL*, Boston, Massachusetts.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT/NAACL*, Edmonton,Canada.

Franz Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguistics*.

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

Matthias Paulik, Kay Rottmann, Jan Niehues, Silja Hildebrand, and Stephan Vogel. 2007. The ISL phrase-based MT system for the 2007 ACL workshop on statistical MT. In *Proc. of the Association of Computational Linguistics Workshop on Statistical Machine Translation*.

Mark Steedman. 1999. Alternative quantifier scope in CCG. In *Proc. of ACL*, College Park, Maryland.

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. 2007. An efficient two-pass approach to synchronous CFG driven MT. In *Proc. of HLT/NAACL*, Rochester, NY.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proc. of the Workshop on Statistical Machine Translation, HLT/NAACL*, New York, June.