INTERSPEECH 2009 BRIGHTON

Using Syntax in Large-Scale Audio Document Translation

Jing Zheng[1] *Necip Fazil Ayan*[1] *Wen Wang*[1] *David Burkett*[2*]

[1]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA 94025, USA
[2]EECS Department, University of California, Berkeley, Berkeley, CA 94720, USA
{zj,nfa,wwang}@speech.sri.com dburkett@cs.berkeley.edu

Abstract

Recently, the use of syntax has very effectively improved machine translation (MT) quality in many text translation tasks. However, using syntax in speech translation poses additional challenges because of disfluencies and other spoken language phenomena, and of errors introduced by automatic speech recognition (ASR). In this paper, we investigate the effect of using syntax in a large-scale audio document translation task targeting broadcast news and broadcast conversations. We do so by comparing the performance of three synchronous context-free grammar based translation approaches: 1) hierarchical phrase-based translation, 2) syntax-augmented MT, and 3) string-to-dependency MT. The results show a positive effect of explicitly using syntax when translating broadcast news, but no benefit when translating broadcast conversations. The results indicate that improving the robustness of syntactic systems against conversational language style is important to their success and requires future effort.

Index Terms: syntax, machine translation, audio document

10.21437/Interspeech.2009-158

1. Introduction

Recently, research has increasingly focused on using syntax in large-scale machine translation (MT) tasks, and has obtained very promising results. The top three performing Chinese-English systems in the *constrained training track* of NIST open MT evaluation in 2008 all employed syntactic information. Syntax is particularly helpful for translating language pairs like Chinese-to-English that require careful consideration of the long-distance reordering issue.

Most published results were drawn from text translation, especially from newswire translation. However, in the DARPA GALE program, the audio document translation task includes the domains of broadcast news (BN) and broadcast conversations (BC). Compared to text translation, audio-document translation poses the following additional challenges:

- Noisy input. Before translation, audio documents are first transcribed by using automatic speech recognition (ASR) technology, which is presently far from error-free. In fact, the state-of-the-art Chinese ASR system still has more than a 15% character error rate (CER) on BC transcription, meaning that most sentences will have at least one error.
- Missing information. The audio document has no punctuation, and the plain ASR output is typically just a stream of text. Although machine-learning techniques can help obtain sentence boundary and type information at a reasonable accuracy, within-sentence punctuation—such as commas and quotation marks, which play important roles in text translation and syntactic parsing—cannot be reliably obtained.
- Spoken language phenomena. BC documents contain mostly spontaneous conversational speech, which has many disfluencies, such as filled pauses, self-repairs, hesitations, etc. Also, in a dialogue, speakers tend to omit syntactic components, such as subjects and objects, without causing inter-speaker confusion. All these phenomena make the source language ungrammatical.

Given these problems, we only explore syntactic MT approaches based on target-side syntax, as such approaches are not directly influenced by noise in the source-side data. Our baseline system is based on the hierarchical phrase-based translation model [1], and we compare it to two syntactic systems trained on the same dataset: one is based on syntax-augmented MT [2], and the other on string-to-dependency MT [3]. Both approaches can be viewed as extensions of the standard hierarchical phrase-based translation approach incorporating information from target-side syntax.

The rest of this paper is organized as follows: Section 2 describes the systems used in this study. Section 3 describes the experimental setup and reports results. Section 4 discusses these results further. And finally, Section 5 summarizes the research findings and suggests future work.

2. Translation Approaches

2.1. Baseline: Hierarchical Phrase-Based Translation

The hierarchical phrase based translation (HPBT) model can be formally described as a weighted synchronous context-free grammar (SCFG) [4]. Using notation from [1], an SCFG rule is written as:

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

where X is a nonterminal, γ and α are both strings of terminals and nonterminals, and ~ is a one-to-one index correspondence between nonterminal occurrences in γ and α. In a standard HPBT model, all rules use a single nonterminal "*X*" except for the two glue rules, which are used to combine a sequence of "*X*"s to form "*S*," the start of all SCFG derivations.

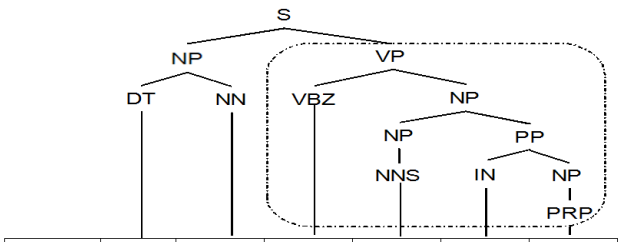
All SCFG rules are associated with a set of features that are used to compute derivation probabilities under a log-linear model [5]. The features used in this work include:

- Relative frequency in two directions
- Lexical weights in two directions
- Phrase penalty
- Hierarchical rule penalty
- Glue rule penalty

In addition to these rule-related features, we also use target language model score and target sentence length. The scaling

[*] Work done during internship at SRI

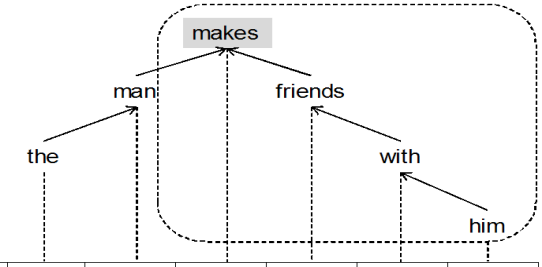
Copyright © 2009 ISCA



	the	man	makes	[friends]	with	[him]
那人	X	X				
和					X	
[他]						X
交			X			
[朋友]				X		

HPBT: X→< '和' X_1 '交' X_2 : 'makes' X_2 'with' X_1 >
SAMT: VP→< '和' PRP_1 '交' NNS_2 : 'makes' NNS_2 'with' PRP_1>

(a) HPBT vs. SAMT



	the	man	makes	[friends]	with	[him]
那人	X	X				
和					X	
[他]						X
交			X			
[朋友]				X		

HPBT: X→< '和' X_1 '交' X_2 : 'makes' X_2 'with' X_1 >
String-to-Dep: X_{fixed}→< '和' X_1 '交' X_2 :
'makes'#h=0 X_2#h=1 'with'#h=2 X_1#h=3>

(b) HPBT vs. String-to-Dependency MT. Attribute "#h=x" means the headword position within the rule (1 for the leftmost position) is x, and x = 0 means the headword is not within the span of the rule.

Figure 1: Illustration of rule extraction

factors for all features are optimized by minimum error rate training (MERT) to maximize BLEU [6].

Given an input sentence in the source language, translation into the target language is cast as a search problem, where the goal is to find the highest-probability derivation that generates the source-side sentence, using the rules in our SCFG. The source-side derivation corresponds to a synchronous target-side derivation and the terminal yield of this target-side derivation is the output of the system. In this work, we use an SRI-developed, CKY-style decoder to solve the search problem.

Although the translation model is formally syntactic, the HPBT rules are extracted without using any syntactic information in the linguistic sense. Instead, the needed data are the parallel text and word alignments. SCFG rules are extracted based on a phrase-containment relationship: initial phrases consistent with word alignments are first extracted as in phrase-based translation [7]. Replacing a smaller phrase embedded in a larger phrase with a generic, nonterminal symbol generates a hierarchical rule. All such rules are extracted, subject to certain constraints that are necessary to limit both model and decoding complexity.

2.2. Syntax Augmented MT

Syntax augmented MT (SAMT) is an extension to hierarchical phrase-based translation, adding linguistic syntactic information. Instead of using a single, generic nonterminal, SAMT uses syntactically categorized nonterminals derived from target-side parse trees over every sentence in the training corpus. The rule-extraction procedure is similar to that of HPBT, except that a phrase is assigned to a category based on how its target span matches with the constituent spans in the parse tree. If the target span exactly matches a constituent span, the label of the constituent is assigned to the left-hand-side nonterminal; otherwise, some heuristics are used to assign categories based on a partial tree structure. Figure 1(a) shows an example of SAMT rule extraction.

In this work, we use different heuristics for non-constituent phrases than in [2]. If a phrase's target span does not match any constituent span in the parse tree, we find the smallest constituent that covers the span. We then transform the label of the constituent to indicate which side of the constituent is incomplete. For example, if the constituent's label is X and the phrase's left boundary of the target span does not match the constituent's left boundary, we prepend a hyphen to X (-X); if the right boundary does not match, we append a hyphen to X (X-). As a result, our grammar operates on nonterminals such as NP, -NP, -NP-, NP-. We apply these simplified heuristics in order to generate a smaller set of nonterminals than in [2], thereby reducing the size of the resulting rule table and alleviating the data-sparseness problem.

We enhance our HPBT decoder to support multiple nonterminals for SAMT decoding. Even with the much reduced nonterminal set, the SAMT rule table is still many times larger than that of HPBT and demands high memory size and processing power. To mitigate this problem, we use a two-pass coarse-to-fine decoding strategy to improve search efficiency [15]. In the first pass, we treat all non-terminals as identical and use an HPBT decoder to find a high-probability hypergraph in the decoding forest. In the second pass, we constrain the SAMT search to the edges of the hypergraph, thereby substantially reducing the search space and decoding time.

In GALE's text-translation task, an SAMT system outperforms an equally trained HPBT system by 0.7-0.8% BLEU absolute in both the newswire and web text genres, as shown in Table 1.

2.3. String-to-Dependency MT

String-to-dependency MT is a recently developed approach that extends the HPBT model with target-side syntactic dependency structures. Both HPBT and SAMT techniques translate a source language string into a target language string, whereas string-to-dependency MT translates a source language string into a target language dependency structure, incorporating a dependency language model into the model. A dependency tree depicts modifier-headword relationship within a sentence, and can be obtained by applying some headword percolation rules to a syntactic parse tree.

Some words that are distant in the surface sentence can be adjacent in a dependency tree, allowing the use of a standard n-gram modeling technique to model their relationships in a dependency language model. This dependency LM has the capacity to model language's syntactic structure, providing

information that is complementary to the standard language model.

String-to-dependency MT rules are created from HPBT rules by replacing target side phrases with the dependency substructures that contain these phrases in the training data, as shown in Figure 1(b). However, only dependency structures that satisfy certain well-formedness criteria can be guaranteed to be composable at decoding time [3]. Therefore, phrases that do not map onto well-formed dependency structures in the training data are discarded, resulting in a loss of phrase pair coverage relative to HPBT or SAMT. Nevertheless, the loss of grammar coverage is usually offset by the power of dependency language model.

We extend our HPBT decoder to support string-to-dependency MT decoding by 1) augmenting the search state with dependency structure information, 2) supporting dependency structure operations, and 3) computing dependency LM probabilities. To address the coverage problem that results from limiting our search space to well-formed target-side dependency structures, we allow ill-formed structures to be created at an additional cost in model score. The SRILM toolkit [14] is used to estimate the dependency language model from statistics collected on a corpus of automatically annotated dependency trees.

In GALE's text-translation task, a string-to-dependency MT system outperforms the HPBT system trained on the same corpus by 1.1% absolute BLEU, as shown in Table 2. As the systems used a different word segmentation and preprocessing pipeline, the Table 1 data cannot be directly compared with Table 2.

Table 1. Comparing HPBT and SAMT on text genres. Results in 4-reference case-insensitive BLEU(%)

	NW	WT
HPBT	31.7	26.0
SAMT	32.5	26.7

Table 2. Comparing HPBT and String-to-dependency MT on text genres. Results in 4-reference case-insensitive BLEU(%).

	NW	WT
HPBT	32.1	25.3
String-to-dep.	33.2	26.4

Table 3. Tuning and testing data size summary.

	# Sentences		# English tokens	
	BN	BC	BN	BC
Dev08 (tune)	529	1134	13k	17k
Test08 (test)	483	937	12k	14k

Table 4. Comparing performance on human transcripts of Test08 data. Results in 4-reference case-insensitive BLEU (%).

	BN	BC
HPBT	30.9	30.0
SAMT	32.2	30.0
String-to-dep	32.0	29.6

Table 5. Comparing performance on ASR output of Test08 data. Results in 4-reference case-insensitive BLEU (%).

	BN	BC
CER (%)	6.7	16.9
HPBT	30.2	26.0
SAMT	31.1	26.0
String-to-dep	31.1	26.0

Table 6. System combination results on Test08.

	BN	BC
BLEU improvement over single best (%)	1.2	0.6
TER reduction over single best (%)	1.0	1.2

3. Experiments

To measure the effectiveness of the explicit use of syntactic information in a speech-based translation pipeline, we tested the three translation approaches on the DARPA GALE Chinese-to-English audio-translation task, made up of broadcast news and broadcast conversation genres. We trained all three translation systems on a parallel text corpus containing 2.4 million sentences, with about 59 million running words in the English side. The vast majority of the data came from the text domain, especially newswire. In order to boost performance on speech translation, we assigned a weight of 5.0 to the limited amount of BN and BC data made available by LDC to the GALE community. GIZA++ was used to generate word alignments with "grow-diag-final-and" symmetrization heuristics [7]. A 4-gram language model trained from about 5 billion words was used in decoding for all three systems.

We trained genre-adapted English parsers by applying co-training [16] to the discriminatively reranked Charniak parser [17] and a combination of the Berkeley parser [18] and a RankBoost-based reranker [19]. To do this, we first assembled an annotated corpus from the Penn Wall Street Journal treebank, the Brown treebank, the Switchboard treebank, and a small seed BN and BC treebank created by the GALE OntoNotes team. We trained the initial models for our two parsers on this corpus, and then bootstrapped on a large amount of unlabeled data from BN and BC transcripts and web text. During each iteration of co-training, a small subset of the unlabeled data was randomly selected. The sentences in this set were then annotated by generating 50-best lists from each parser and reranking with each parser's corresponding discriminative reranker. We then used some example selection heuristics to select a subset of the data that had been annotated by the Charniak parser and added it to the training pool of the Berkeley Parser. We likewise augmented the training data for the Charniak parser with output from the Berkeley parser. Both parsers were then retrained based on the updated training pool, and the entire procedure was repeated until all the unlabeled data had been parsed [8]. By combining co-training and discriminative reranking in this way, we improved parsing accuracy significantly on the three low-resource genres: WT, BN, and BC.

To train our syntactic MT systems, we parsed every English sentence in the bilingual training corpus to generate parse trees, which were then used to extract SAMT rules. Applying Magerman's percolation rules [9] to the parse trees, we obtained the dependency trees that we used to train the string-to-dependency MT model. We also used these dependency trees, along with additional parsed English text to train the dependency language model.

We used two GALE test sets for the experiment: Dev08 and Test08. Characteristics of the two test sets are shown in Table 3. We used Dev08 to optimize the log-linear model scaling factors, and evaluated on Test08. We used case-insensitive IBM BLEU [10] with four references as our target metric for minimum error rate training and to report results.

First, we wanted to measure the effect of syntax without factoring in ASR errors, so we tested the systems on the human transcripts of the test sets. To simulate real ASR output, we stripped out sentence-internal punctuation from these transcripts and removed punctuation from the source sides of all translation rules, using the approach described in

[11]. As Table 4 shows, on BN data both the SAMT and the string-to-dependency system performed better than the HPBT system, with improvements comparable to those in text genres, indicating that linguistic syntactic information is indeed helpful for translation. However, for BC, this situation is different, as neither the SAMT nor the string-to-dependency system outperformed the HPBT system.

We then evaluated on real ASR output. For Test08, the ASR character error rate is 6.7% for BN and 16.9% for BC. As reflected in Table 5, for BN, the systems using linguistic syntax still performed better than the formal syntax system, although the gain is slightly smaller; for BC, once again, no difference in performance was shown.

Finally, we ran a system combination experiment to check if the three approaches are complementary to each other. The actual systems used in this experiment were augmented with additional language models applied in n-best rescoring steps, as in [20], with a word-based system combination algorithm as described in [21]. As Table 6 shows, system combination improved performance on both BN and BC data, as measured by both BLEU and TER [12], indicating there is indeed complementariness among the systems.

4. Discussion

The results indicate that explicitly using syntactic information helps when translating broadcast news, but not when translating broadcast conversations. The reason seems more related to genre than ASR error rate. On BN, both SAMT and string-to-dependency MT have better performance than HPBT, for either human transcripts or real ASR output. On BC, however, neither SAMT nor string-to-dependency MT has any advantage, even when translating error-free human transcriptions. Why doesn't syntax help with BC?

By manually inspecting BN and BC data, we observe that the style of BC sentences is far more conversational than that of BN. In BC, disfluencies are very common, making sentences frequently ungrammatical. Checking the string-to-dependency decoder output, we find that the decoder often chose to take the penalty instead of generating a complete dependency tree. This indicates that the translation rules and the dependency LM are not well adapted to conversational language. Making the use of syntax more effective for BC translation requires addressing disfluencies and other conversational language phenomena.

We also observed very significant degradation from translating human transcripts to ASR output on BC, ranging from 3.6 to 4.0 BLEU points. Therefore, improving ASR accuracy is also crucial to raising BC translation quality.

5. Conclusions and Future Work

We compared performance of three SCFG-based translation approaches—HPBT, SAMT, and string-to-dependency MT—on a GALE test set. SAMT and string-to-dependency MT outperform HPBT on BN by a margin of 0.9-1.1 BLEU points, indicating that syntactic information is indeed helpful for BN translation. On BC data, all three approaches performed similarly, although combining them generated improvement.

To improve the effectiveness of using syntax for BC translation, we need to address the issue of disfluencies. One possible solution is to adapt the translation model and dependency LM to the BC genre. A second possible solution is to automatically detect and remove disfluencies from the source language, as in [13]. We hope to investigate both these approaches in future work.

6. Acknowledgements

We thank Dr. Andreas Stolcke for his help in solving some issues of using SRILM toolkit in this work. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. References

[1] Chiang, D., "Hierarchical Phrase-based Translation", Computational Linguistics, 33(2):573-605, 2007.

[2] Zollmann, A. and Venugopal A., "Syntax Augmented Machine Translation via Chart Parsing", in NAACL 2006 - Workshop on statistical machine translation, New York, 2006.

[3] Shen, L., Xu, J. and Weischedel R., "A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model", in ACL 2008, Columbus, OH, 2008.

[4] Lewis, P. M., and Stearns. R. E., "Syntax-directed transduction", Journal of the ACM, 15:465–488. 1968.

[5] Och, F. and Ney, H., "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation", in ACL 2002, Philadelphia, PA, 2002.

[6] Och, F., "Minimum Error Rate Training for Statistical Machine Translation", in ACL 2003, Sapporo, Japan, 2003

[7] Koehn, P., Och, F. and Marcu, D., "Statistical Phrase-Based Translation". In HLT-NAACL 2003, Edmonton, Canada, 2003.

[8] Wang, W., "Combining Discriminative Re-ranking and Co-training for Parsing Mandarin Speech Transcripts", in ICASSP 2009, Taiwan, 2009.

[9] Magerman, D., "Statistical Decision-Tree Models for Parsing", in ACL-1995, Cambridge, MA, 1995.

[10] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J., "BLEU: a method for automatic evaluation of machine translation", in ACL-2002, Philadelphia, PA, 2002.

[11] Zheng, J., Wang, W., and Ayan, N., "Development of SRI's translation systems for broadcast news and broadcast conversations", in Interspeech'2008, Brisbane, Australia, September 2008.

[12] Snover, M., Dorr, B., Schwartz R., Micciulla L., and Makhoul J., "A study of translation edit rate with targeted human annotation", in AMTA-2006, Cambridge, MA, 2006.

[13] Liu, Y., Stolcke, A., Shriberg, E., and Harper, M., "Using Conditional Random Fields For Sentence Boundary Detection in Speech", in ACL-2005, Ann Arbor, MI, 2005.

[14] Stocke, A., "SRILM – an extensible language modeling toolkit", in. ICSLP-2002, Denver, CO, 2002.

[15] Zhang, H. and Gildea, D., "Efficient Multi-pass Decoding for Synchronous Context Free Grammars", in ACL-08:HLT, Columbus, OH, 2008.

[16] Blum, A. and Mitchell, T., "Combining labeled and unlabeled data with co-training", in Proceedings of COLT, 1998.

[17] E. Charniak and M. Johnson, "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking", in ACL, 2005.

[18] Petrov, S., Barrett, L., Thibaux, R., and Klein, D., "Learning Accurate, Compact, and Interpretable Tree Annotation", in ACL, Sydney, Australia, 433--440, 2006.

[19] Collins, M. and Koo. T., "Discriminative reranking for natural language parsing", Computational Linguistics, 31(1):25-70, 2005

[20] Wang, W., Stolcke, A., and Zheng, J., "Reranking Machine Translation Hypotheses With Structured and Web-based Language Models", in Proc. IEEE Automatic Speech Recognition and Understanding Workshop, Kyoto, 2007.

[21] Ayan, N., Zheng, J., and Wang W., "Improving Alignments for Better Confusion Networks for Combinating Machine Translation Systems," In Proc. 22nd Int'l Conf. Computational Linguistics (COLING'08), Manchester, UK, 2008.