ISI's 2005 Statistical Machine Translation Entries

Steve DeNeefe, Kevin Knight

Information Sciences Institute and Department of Computer Science
The Viterbi School of Engineering, University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
sdeneefe@isi.edu, knight@isi.edu

Abstract

ISI entered two statistical machine translation systems in the IWSLT evaluation this year: one was phrase-based and the other syntax-based. The syntax-based system represents the results of a current research effort, while the phrase-based system is representative of the current techniques in state-of-the-art machine translation. This paper primarily describes the syntax-based system and its comparison to the phrase-based system. We will give a brief overview of the components of the systems and discuss the performance on the IWSLT development data, the evaluation results, and some post-evaluation results.

1. Introduction

Statistical phrase-based machine translation is currently the state-of-the-art in many translation tasks, achieving results often surpassing other methods [1]. Systems that strive to improve on the current results of phrase-based machine translation often incorporate a higher-level notion of the structure in language, sometimes just the hierarchical structure [2] and other times a fully syntactic model. For the past several years, the ISI/USC machine translation group has been investigating how to use syntactic information to improve translation quality beyond the capability of our existing phrase-based translation system and in the process created a new syntax-based translation system. Both systems have similarities: they are both statistical and trained on bilingual parallel data, both combine their translation model with several other knowledge sources in a log-linear manner, and both require parameter tuning to determine the weights of the individual components. The syntax-based system is different in two main respects: the translation model incorporates syntactic structure on the target language side (in our case, English), and the decoder uses a parser-like method to create syntactic trees as output hypotheses.

1.1. Syntax-based Translation Model

Simply put, our syntax model translates phrases in the source language into syntactic chunks in the target language. For example, when translating from Chinese into English, our system learns simple rules that translate words or phrases, such as

$$\texttt{NPB(PRP(I))} \leftrightarrow 我$$
$$\texttt{NN(hotel)} \leftrightarrow 酒店$$
$$\texttt{NP-C(NPB(DT(this) NN(address)))} \leftrightarrow 这个\ 地址$$

It also learns phrases with "holes" in the source language (represented here by the variable x_0), as long as they conform to a syntactic structure in the target language

$$\texttt{NP-C(NPB(PRP\$(my)}\ x_0\texttt{:NN))} \leftrightarrow 我\ 的\ x_0$$
$$\texttt{NP-C(NPB(PRP\$(my)}\ x_0\texttt{:NN))} \leftrightarrow 我\ x_0$$
$$\texttt{PP(TO(to) NP-C(NPB(}x_0\texttt{:NNP NNP(park))))} \leftrightarrow 去\ x_0\ 公园$$

Other rules bring together already translated phrases, such as the following rules which take a translated verb next to a translated noun phrase and combine them together into a verb phrase:

$$\texttt{VP(}x_0\texttt{:VBZ}\ x_1\texttt{:NP-C)} \leftrightarrow x_0\ x_1$$
$$\texttt{VP(}x_0\texttt{:VBZ}\ x_1\texttt{:NP-C)} \leftrightarrow x_1\ x_0$$

The first rule combines the pair in order. The second takes a noun phrase located before a verb, switches the order, then builds the final verb phrase.

To learn these rules automatically, we first word-aligned a bilingual parallel corpus using GIZA++ [3]. We then parsed the target side[1] using our own implementation of Collins Model 2 [5], [6]. This resulted in a large set of tree-string pairs, aligned at the word level. From this set, a list of translation rules were extracted, in the manner described by [7]. Probabilities were applied according to a relative frequency model conditioned on the root non-terminals of the left-hand sides of the rules.

[1] Actually, we bootstrapped our parser by first training it on the Penn Treebank [4], then used the resulting parsing model to parse the English half of the supplied training data. We then re-trained a second-generation parser on this data, which was then used to parse the same data a second time.

Language	Phrase-based			Syntax-based		
	Pre-eval blind test	Evaluation	**Post-eval** (correctly trained)	Pre-eval blind test	Evaluation	**Post-eval** (trigram model)
Arabic	53.79	37.39	**50.16**	43.84	39.62	**44.47**
Chinese	32.1	33.23	**41.16**	25.73	37.64	**40.08**
Japanese	44.07	28.31	**33.82**	36.66	27.41	**29.98**
Korean	35.48	23.74	**30.02**	26.2	25.22	**27.65**

Table 1: BLEU scores on Syntax and Phrase systems on both evaluation data and blind test sets. The post-evaluation phrase system is using the correctly trained phrase tables. The post-evaluation syntax system is using a trigram language model integrated into the decoder search.

Language	Phrase-based			Syntax-based		
	Pre-eval blind test	Evaluation	**Post-eval** (correctly trained)	Pre-eval blind test	Evaluation	**Post-eval** (trigram model)
Arabic	0.9544	0.7528	**0.9591**	0.8444	0.8157	**0.8989**
Chinese	0.9562	0.8897	**0.9750**	0.9312	0.9742	**0.9757**
Japanese	0.9704	0.8715	**0.9529**	0.8885	0.7421	**0.9042**
Korean	0.9734	0.9231	**0.9997**	0.8344	0.8365	**0.9466**

Table 2: Brevity Penalties on Syntax and Phrase systems on both evaluation data and blind test sets for the same runs as shown in Table 1.

1.2. Language Model

For the evaluation run we integrated a smoothed bigram model into our decoder search, and generated lists of 25,000 hypotheses for each sentence, then re-ranked these results using a smooth trigram model.[2] We used the SRI Language Modeling Toolkit to train both language models, and trained on the English half of the supplied parallel training corpus, which contained 192,362 words (7,803 unique) after preprocessing.

1.3. Model Weight Training

To train the individual model weights of the log-linear model, we split the provided development data into two parts, nearly equal in size. For Chinese, Arabic, and Japanese, devset 1 was used as blind test data, while devset 2 was reserved for development training of the weights. Since only one devset was supplied for Korean, we split this devset in two, and used the first 253 lines for testing, while the second 253 were reserved for training.

The syntax system does not yet have a reliable automatic parameter tuning method. Instead, we used a much slower exhaustive method to train our model weights. We ran our decoder on the development set using hundreds of parameter settings, each time recording the BLEU score. The settings that resulted in the highest BLEU score were then run on the blind test corpus, along with our baseline settings to ensure that we had made some improvement. This method was very time consuming, so we only had time to tune values for the Chinese development set. We used these same parameters for translating the other three languages.

[2] Due to the search space complexity of combining our translation model with a language model, we were at the time unable to integrate a trigram language model into the search process. In our post-evaluation runs of the syntax system, we did use an integrated trigram model, and did no re-ranking.

1.4. Syntax-based Decoder

Our syntax decoder implements a probabilistic CKY-style parsing algorithm with beams. It applies the translation rules to the Chinese sentence and builds its way, step-by-step, to the top of an English parse structure, as discussed in [8]. This results in an English syntax tree corresponding to the Chinese sentence, which guarantees the output to have some kind of globally coherent syntactic structure.

1.5. The Contrastive System: Phrase-Based MT

The phrase-based machine translation system we entered in the evaluation is the same as last year [9], [10], except that it was trained solely on the supplied data. It used the smoothed trigram language model in an integrated fashion, and the model weights were trained using the minimum error rate training method described in [11]. For training this system, we used the same training/testing split of the development data described above.

2. Results

In Table 1, we report three sets of BLEU scores for both the syntax and phrase systems: one for our blind test set (mea-

sured on devset 1 after training on devset 2),[3] one for the final evaluation results, and one for a post-evaluation run. Note that for the syntax system, the evaluation and test scores are relatively comparable, while for the phrase system, the evaluation scores are much lower than the test scores. This was an error on our part while running the phrase-based system on the evaluation data: we did not correctly re-collect the phrase tables with respect to the evaluation source data, so our syntax system did not have all the relevant phrase-pairs while decoding. After this problem was discovered, we fixed this problem in our phrase tables, and re-ran the same system. The results are shown in Table 1 in the phrase-based post-evaluation column, and are more consistent with our expectations for this system.

After the evaluation, we were also able to run the syntax system with an integrated trigram model. Those results, again tuned only on the Chinese development data, are shown in Table 1's syntax-based post-evaluation column.

Table 2 shows the brevity penalties for the same runs as Table 1. Again, note the severe penalties given the phrase-based system on the evaluation runs (second column), as compared to the post-evaluation run (third column) and blind test results (first column). The syntax system, on the other hand, produces short sentences consistently for all languages except Chinese, an indication that tuning in each language might be advantageous.

3. Discussion

We were quite surprised at the poor evaluation scores of our phrase-based system. As our post-evaluation results (and the results of other teams) demonstrate, these scores were certainly not indicative of the caliber of the phrase-based approach. Even a good system can be thwarted by user error.

On the other hand, our syntax system's performance was a more pleasant surprise. Especially in our post-evaluation run of the Chinese system, when using the trigram language model integrated into the search, the syntax-based system achieved results close to those of the phrase-based system. This is surprising because the syntax system is currently not able to learn phrase pairs to the same level as a phrase-based system. With such a small training dataset as what was given for this evaluation, our system encountered many unknown words in the test data. Thus the resulting sentences were sometimes short on content words. But apparently the strengths of the syntax-based approach made up for this deficiency in part.

Also worth mentioning is that questions comprised a large percentage of the data in this evaluation. This is an area where a syntax-based method could really shine, or do quite poorly, based on the quality of the parsing and how well the model handles large-scale movements in the tree.[4] Since we trained our parser on text that contains very few questions, it is unlikely that the resulting parse trees for questions were of very high quality. Manual inspection of our translations also shows that questions were not translated well. Better quality parsing of questions is one of the areas we will be investigating.

[3] As mentioned before, scores for Korean were measured only on first half of devset 1.

[4] Our current translation model does allow movements, but perhaps not at the scale necessary.

4. References

[1] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT-NAACL 2003)*, pp 48–54.

[2] David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 263–270.

[3] Franz Josef Och, Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, volume 29, number 1, pp. 19–51.

[4] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, volume 19, number 2, pp. 313–330.

[5] Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, volume 29, number 4, pp. 589–673.

[6] Daniel M. Bikel. 2004. Intricacies of Collins' Parsing Model. In *Computational Linguistics*, volume 30, number 4, pp. 479–511.

[7] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT-NAACL 2004)*, pp. 273–280.

[8] Steve DeNeefe, Kevin Knight, and Hayward H. Chan. 2005. Interactively Exploring a Machine Translation Model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005) Interactive Poster and Demonstration Sessions*, pp. 97–100.

[9] Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, volume 30, number 4, pp. 417–449.

[10] Ignacio Thayer, Emil Ettelaie, Kevin Knight, Daniel Marcu, Dragos Stefan Munteanu, Franz Josef Och, Quamrul Tipu. 2004. The ISI/USC MT system. In

Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2004), pp. 59–60.

[11] Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pp. 160–167.