Improving Phrase-Based Translation with Prototypes of Short Phrases

Frank Liberato[†], **Behrang Mohit**[‡], **Rebecca Hwa**[†‡]
[†]Department of Computer Science [‡]Intelligent Systems Program
University of Pittsburgh
{frank,behrang,hwa@cs.pitt.edu}

Abstract

We investigate methods of generating additional bilingual phrase pairs for a phrase-based decoder by translating short sequences of source text. Because our translation task is more constrained, we can use a model that employs more linguistically rich features than a traditional decoder. We have implemented an example of this approach. Experimental results suggest that the phrase pairs produced by our method are useful to the decoder, and lead to improved sentence translations.

1 Introduction

Recently, there have been a number of successful attempts at improving phrase-based statistical machine translation by exploiting linguistic knowledge such as morphology, part-of-speech tags, and syntax. Many translation models use such knowledge before decoding (Xia and McCord, 2004) and during decoding (Birch et al., 2007; Gimpel and Smith, 2009; Koehn and Hoang, 2007; Chiang et al., 2009), but they are limited to simpler features for practical reasons, often restricted to conditioning left-to-right on the target sentence. Traditionally, n-best rerankers (Shen et al., 2004) have applied expensive analysis after the translation process, on both the source and target side, though they suffer from being limited to whatever is on the n-best list (Hasan et al., 2007).

We argue that it can be desirable to pre-translate parts of the source text before sentence-level decoding begins, using a richer model that would typically be out of reach during sentence-level decoding. In this paper, we describe a particular method of generating additional bilingual phrase pairs for a new source text, using what we call *phrase prototypes*, which are are learned from bilingual training data. Our goal is to generate improved translations of relatively short phrase pairs to provide the SMT decoder with better phrasal choices. We validate the idea through experiments on Arabic-English translation. Our method produces a 1.3 BLEU score increase (3.3% relative) on a test set.

2 Approach

Re-ranking tends to use expensive features of the entire source and target sentences, s and t, and alignments, a, to produce a score for the translation. We will call this scoring function $\phi(s, t, a)$. While $\phi(\cdot)$ might capture quite a bit of linguistic information, it can be problematic to use this function for decoding directly. This is due to both the expense of computing it, and the difficulty in using it to guide the decoder's search. For example, a choice of $\phi(\cdot)$ that relies on a top-down parser is difficult to integrate into a left-to-right decoder (Charniak et al., 2003).

Our idea is to use an expensive scoring function to guide the search for potential translations for *part* of a source sentence, S, even if translating all of it isn't feasible. We can then provide these translations to the decoder, along with their scores, to incorporate them as it builds the complete translation of S. This differs from approaches such as (Och and Ney, 2004) because we generate new phrase pairs in isolation, rather than incorporating everything into the sentence-level decoder. The baseline system is the Moses phrase-based translation system (Koehn

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 301–304,
Los Angeles, California, June 2010.

et al., 2007).

2.1 Description of Our Scoring Function

For this work, we consider a scoring function based on part-of-speech (POS) tags, $\phi_{POS}(\cdot)$. It operates in two steps: it converts the source and target phrases, plus alignments, into what we call a *phrase prototype*, then assigns a score to it based on how common that prototype was during training.

Each phrase pair prototype is a tuple containing the source prototype, target prototype, and alignment prototype, respectively. The source and target prototypes are a mix of surface word forms and POS tags, such as the Arabic string ⟨NN Al JJ⟩, or the English string ⟨NN NN⟩. For example, the source and target prototypes above might be used in the phrase prototype ⟨NN_0 Al JJ_1 , NN_1 NN_0⟩, with the alignment prototype specified implicitly via subscripts for brevity. For simplicity, the alignment prototype is restricted to allow a source or target word/tag to be unaligned, plus 1:1 alignments between them. We do not consider 1:many, many:1, or many:many alignments in this work.

For any input $\langle s, t, a \rangle$, it is possible to construct potentially many phrase prototypes from it by choosing different subsets of the source and target words to represent as POS tags. In the above example, the Arabic determiner Al could be converted into an unaligned POS tag, making the source prototype ⟨NN DT JJ⟩. For this work, we convert all aligned words into POS tags. As a practical concern, we insist that unaligned words are always kept as their surface form.

$\phi_{POS}(s, t, a)$ assign a score based on the probability of the resulting prototypes; more likely prototypes should yield higher scores. We choose:

$$\phi_{POS}(s, t, a) = p(SP, AP|TP) \cdot p(TP, AP|SP)$$

where SP is the source prototype constructed from s, t, a. Similarly, TP and AP are the target and alignment prototypes, respectively.

To compute $\phi_{POS}(\cdot)$, we must build a model for each of $p(SP, AP|TP)$ and $p(TP, AP|SP)$. To do this, we start with a corpus of aligned, POS-tagged bilingual text. We then find phrases that are consistent with (Koehn et al., 2003). As we extract these phrase pairs, we convert each into a phrase prototype by replacing surface forms with POS tags for all aligned words in the prototype.

After we have processed the bilingual training text, we have collected a set of phrase prototypes and a count of how often each was observed.

2.2 Generating New Phrases

To generate phrases, we scan through the source text to be translated, finding any span of source words that matches the source prototype of at least one phrase prototype. For each such phrase, and for each phrase prototype which it matches, we generate all target phrases which also match the target and alignment prototypes.

To do this, we use a word-to-word dictionary to generate all target phrases which honor the alignments required by the alignment prototype. For each source word which is aligned to a POS tag in the target prototype, we substitute all single-word translations in our dictionary[1].

For each target phrase that we generate, we must ensure that it matches the target prototype. We give each phrase to a POS tagger, and check the resulting tags against any tags in the target prototype. If there are no mismatches, then the phrase pair is retained for the phrase table, else it is discarded. In the latter case, $\phi_{POS}(\cdot)$ would assign this pair a score of zero.

2.3 Computing Phrase Weights

In the Moses phrase table, each entry has four parameters: two lexical weights, and the two conditional phrase probabilities $p(s|t)$ and $p(t|s)$. While the lexical weights can be computed using the standard method (Koehn et al., 2003), estimating the conditional phrase probabilities is not straightforward for our approach because they are not observed in bilingual training data. Instead, we estimate the *maximum* conditional phrase probabilities that would be assigned by the sentence-level decoder for this phrase pair, as if it had generated the target string from the source string using the baseline phrase table[2]. To do this efficiently, we use some

[1] Since we required that all unaligned target words are kept as surface forms in the target prototype, this is sufficient. If we did not insist this, then we might be faced with the unenviable task of choosing a target languange noun, without further guidance from the source text.

[2] If we use these probabilities for our generated phrase pair's probability estimates, then the sentence-level decoder would see

simplifying assumptions: we do not restrict how often a source word is used during the translation, and we ignore distortion / reordering costs. These admit a simple dynamic programming solution.

We must also include the score from $\phi_{POS}(\cdot)$, to give the decoder some idea of our confidence in the generated phrase pair. We include the phrase pair's score as an additional weight in the phrase table.

3 Experimental Setup

The Linguistic Data Consortium Arabic-English corpus2[3] is used to train the baseline MT system (34K sentences, about one million words), and to learn phrase prototypes. The LDC multi-translation Arabic-English corpus (NIST2003)[4] is used for tuning and testing; the tuning set consists of the first 500 sentences, and the test set consists of the next 500 sentences. The language model is a 4-gram model built from the English side of the parallel corpus, plus the English side of the wmt07 German-English and French-English news commentary data. The baseline translation system is Moses (Koehn et al., 2007), with the `msd-bidirectional-fe` reordering model. Evaluation is done using the BLEU (Papineni et al., 2001) metric with four references. All text is lowercased before evaluation; recasing is not used. We use the Stanford Arabic POS Tagging system, based on (Toutanova et al., 2003)[5]. The word-to-word dictionary that is used in the phrase generation step of our method is extracted from the highest-scoring translations for each source word in the baseline phrase table. For some closed-class words, we use a small, manually constructed dictionary to reduce the noise in the phrase table that exists for very common words. We use this in place of a stand-alone dictionary to reduce the need for additional resources.

4 Experiments

To see the effect on the BLEU score of the resulting sentence-level translation, we vary the amount of bilingual data used to build the phrase prototypes.

(approximately) no difference between building the generated phrase using the baseline phrase table, or using our generated phrase pair directly.

[3]Catalogue numbers LDC2004T17 and LDC2004T18

[4]Catalogue number: LDC2003T18

[5]It is available at http://nlp.stanford.edu/software/tagger.shtml

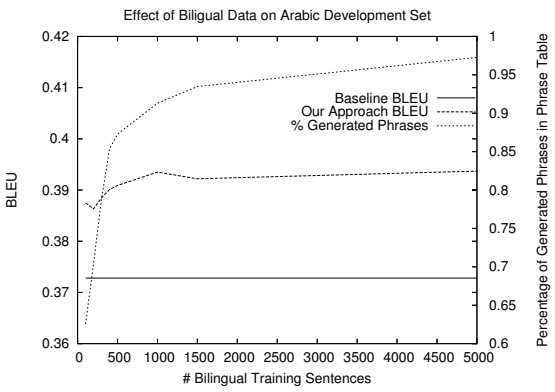

Figure 1: Bilingual training size vs. BLEU score (middle line, left axis) and phrase table composition (top line, right axis) on Arabic Development Set. The baseline BLEU score (bottom line) is included for comparison.

As we increase the amount of training data, we expect that the phrase prototype extraction algorithm will observe more phrase prototypes. This will cause it to generate more phrase pairs, introducing both more noise and more good phrases into the phrase table. Because quite a few phrase prototypes are built in any case, we require that each is seen at least three times before we use it to generate phrases. Phrase prototypes seen fewer times than this are discarded before phrase generation begins. Varying this minimum support parameter does not affect the results noticeably.

The results on the tuning set are seen in Figure 1. The BLEU score on the tuning set generally improves as the amount of bilingual training data is increased, even as the percentage of generated phrases approaches 100%. Manual inspection of the phrase pairs reveals that many are badly formed; this suggests that the language model is doing its job in filtering out disfluent phrases.

Using the first 5,000 bilingual training sentences to train our model, we compare our method to the baseline moses system. Each system was tuned via MERT (Och, 2003) before running it on the test set. The tuned baseline system scores 38.45. Including our generated phrases improves this by 1.3 points to 39.75. This is a slightly smaller gain than exists in the tuning set experiment, due in part that we did not

run MERT for experiment shown in Figure 1.

5 Discussion

As one might expect, generated phrases both help and hurt individual translations. A sentence that can be translated starting with the phrase "`korea added that the syrian prime minister`" is translated by the baseline system as "`korean | foreign minister | added | that | the syrian`". While "`the syrian foreign minister`" is an unambiguous source phrase, the baseline phrase table does not include it; the language and reordering models must stitch the translation together. Ours method generates "`the syrian foreign minister`" directly.

Generated phrases are not always correct. For example, a generated phrase causes our system to choose "`europe role`", while the baseline system picks "`the role of | europe`". While the same prototype is used (correctly) for reordering Arabic "$\texttt{NN}_0$ $\texttt{JJ}_1$" constructs into English as "$\texttt{NN}_1$ $\texttt{NN}_0$" in many instances, it fails in this case. The language model shares the blame, since it does not prefer the correct phrase over the shorter one. In contrast, a 5-gram language model based on the LDC Web IT 5-gram counts[6] prefers the correct phrase.

6 Conclusion

We have shown that translating short spans of source text, and providing the results to a phrase-based SMT decoder can improve sentence-level machine translation. Further, it permits us to use linguistically informed features to guide the generation of new phrase pairs.

Acknowledgements

This work is supported by U.S. National Science Foundation Grant IIS-0745914. We thank the anonymous reviewers for their suggestions.

References

A. Birch, M. Osborne, and P. Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proc. of the Second Workshop on SMT*.

E. Charniak, K. Knight, and K. Yamada. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX*.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Assoc. for Computational Linguistics*.

K. Gimpel and N.A. Smith. 2009. Feature-rich translation by quasi-synchronous lattice parsing. In *Proc. of EMNLP*.

S. Hasan, R. Zens, and H. Ney. 2007. Are very large n-best lists useful for SMT? *Proc. NAACL, Short paper*, pages 57–60.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, page 54.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual meeting-Association for Computational Linguistics*, volume 45, page 2.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41st Annual Meeting on Assoc. for Computational Linguistics*.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of Association for Computational Linguistics*.

L. Shen, A. Sarkar, and F.J. Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Joint HLT and NAACL Conference (HLT 04)*, pages 177–184.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

F. Xia and M. McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*.

[6] Catalogue number LDC2006T13.