Assignment 2: Phrase-based Machine Translation on Small Data — Efforts, Failures, and Lessons Learned

Pengcheng Yin
pcyin@cs.cmu.edu

Abstract

We implemented a simple phrase-based machine translation system based on noisy-channel models. The implementation consists of a cascaded pipeline of tri-gram language modeling, word alignment, phrase extraction, and FST-based decoding. We tried several extensions: (1) word alignment using IBM model 2, (2) Bi-directional word alignment with `grow-diag-final-and` merging heuristics, and (3) heuristic pruning of phrase table. Our best model achieved 18.79 BLEU points, with a fast decoding speed of only 12.94 seconds on the whole test set. This was obtained by using uni-directional (German↦English) word alignment with IBM Model 2 and heuristic pruning of phrase table. We posit the reason why more advanced methods (e.g., bi-directional word alignment) fail in our case is because of the lack of enough training data and the nature of the language pair we use.

1 Introduction

Statistical phrase-based Machine Translation models (a.k.a PBMT) (Koehn et al., 2003) have enjoyed much success over the last 13 years, which was recently surpassed by neural network-based machine translation (NMT) models (Bahdanau et al., 2014). However, a PBMT system still has several advantages. First, it is able to use a phrase table to memorize rare high-order phrase pairs. Second, a PBMT can be easily formulated as a noisy channel model $p(E|F) \propto p(F|E) \cdot p(E)$, which makes it convenient to utilize unlabeled data to model $p(E)$.

In this assignment, we built a simple PBMT. It uses IBM Models 1&2 for learning word alignment, and open FST to facilitate fast search for the best hypothesis. We tried the following extensions:

- We observe that a simple pruning strategy of the phrase extraction algorithm that does not considered bordering non-aligned words in the source side could significantly reduce the size of phrase table, which leads to 164x speeds up in decoding time and better BLEU score.

- We played with both IBM model 1 and 2 for learning word alignments. Our system using IBM model 2 achieved the best BLEU score of 18.79.

- Moreover, we hoped bi-directional word alignments with advanced merging heuristics like `grow-diag-final-and` would work better. However, our empirical results are counter-intuitive. This is probably because of the lack of training data and the bad performance (in terms of the end-to-end BLEU score) of IBM models in English↦German direction.

2 Extensions

2.1 Pruning Phrase Table

We prune the phrase table by ignoring bordering non-aligned words in the source side. This means that we do not implement lines 11 - 18 in Algorithm 6 in the course material. This simple and crude pruning method yields surprisingly good performance: it significantly reduces the size of phrase table, leading

	w/o pruning	w/ pruning
BLEU score	17.97	**18.29**
Num. phrase pairs	1197K	261K
Decoding time (test set)	2116s	**12.94s**

Table 1: Performance with and without phrase table pruning

Settings	Dev. BLEU	Test BLEU
IBM Model 1 (English↦German)	14.58	14.61
IBM Model 1 (German↦English)	**19.36**	18.29
IBM Model 2 (English↦German)	13.47	14.70
IBM Model 2 (German↦English)	19.30	**18.79**
IBM Model 2 (`grow-diag-final-and`)	18.28	18.17

Table 2: Performance with different word alignment models

to 164 times faster decoding speed. Meanwhile, the BLEU score got better.

2.2 Word Alignment with IBM Model 2

We tried IBM Model 2 for learning word alignment. Compared with the simpler IBM Model 1 which assumes uniform alignment probability, IBM Model 1 learns the alignment probability $p(a_j = i||E|, |F|)$ using EM algorithm. In our implementation, we initialize IBM Model 2 using the learned lexical translation probability from IBM Model 1.

2.3 Bi-directional Word Alignment

Hoping that we can further improve the quality of the phrase table and therefore the end-to-end translation performance by learning a bi-directional word alignment model, we trained two word alignment models (IBM Model 2) for both German↦English and English↦German directions. The alignment results are then merged using the `grow-diag-final-and` heuristic[1]. Training IBM Model 2 in two directions is slow, so we use multi-threading to train in both directions simultaneously.

Unfortunately, this bidirectional model fails to outperform our unidirectional model. We present more analysis in Section 3.2.

2.4 Others

We also played with other extensions like allowing for null alignment in the target side. We do not include these in the report since we do not observe any improvements.

3 Empirical Results

3.1 Effectiveness of Pruning Phrase Table

Table 1 lists the performance of our basic implementation with and without pruning the phrase table. We use an IBM Model 1 in German↦English direction to learn word alignments. With such a simple pruning strategy, our baseline implementation achieves a BLEU score of 18.29, and is able to decode the whole test set of 1565 sentences in only 12 seconds.

Nevertheless, we remark that this pruning strategy of ignoring all bordering non-aligned words in the source side is rather crude. We posit that the reason it works well is probably because that the data set we use is relatively small.

3.2 Comparison of IBM Model 1 and 2

Table 2 lists the performance of our system when using IBM Models 1 and 2 for learning word alignment. We tested both directions of German↦English and English↦German.

Interestingly, we found that the performance of German↦English is consistently better than the opposite direction. This is probably because of the effect of compound words in German: a single Ger-

[1] `http://www.statmt.org/moses/?n=FactoredTraining.AlignWords`

man compound word could map to multiple English words. Therefore, learning the lexical translation probability $p(e|f)$(German↦English) would be more meaningful.

Our next experiment combines the alignment results from IBM Model 2 in both directions using the `grow-diag-final-and` heuristic. Unfortunately, we got worse results — the BLEU score on the test set is only 18.17. While the actual reason remains unclear, we posit that this is because of two factors. First, the lack of enough training data makes estimating phrasal translation probabilities difficult when the number of noisy phrase pairs grows larger. Second, the bad performance of English↦German word alignments (c.f. Table 2) becomes a bottleneck of the bi-directional alignment model.

References

[Bahdanau et al.2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel 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*, pages 48–54. Association for Computational Linguistics.