Homework 1: Alignment

Roger Que

2014–02–18

This aligner is a Python implementation of IBM Model 2 [1]. It independently trains two models, one in the "forward" direction from source to target, and the other in the "reverse" direction from target to source, through five iterations of EM. It then uses both models to align each sentence pair, according to the most probable decoding of each individual word, and returns only those alignments that appear in both decodings. This intersection process reduces the probability that rare source words will become "garbage collectors" that align to too many words in the target language because there is not enough evidence to minimize the probability of superfluous alignments [4, 3]. Testing shows that intersection trades recall for a significant increase in precision, improving overall AER.

I also implemented a version of [2], which reparameterizes Model 2 by adding a diagonal alignment prior. This method has the advantage of assigning better position probabilities than Model 1, which assumes a uniform distribution, while not requiring EM over another set of parameters as Model 2 does. Due to time constraints, instead of learning the precision parameter λ by gradient descent as outlined in the original paper, this implementation used a constant λ throughout EM. The value was selected by training on the first 1000 sentences of the data set with $\lambda \in \{1, 2, 3, 4, 5\}$, and choosing the value that resulted in the lowest AER. In the case of "forward"-only alignment, this method gave an optimal λ value of 2; with intersection, 1. In both cases, while my implementation showed an improvement in AER over Model 1, against Model 2 it fared somewhat worse. As a result, the submitted aligner uses Model 2.

Model	AER	**Pre**	**Rec**
Model 1	0.37	0.57	0.74
[2], $\lambda = 2$	0.33	0.63	0.77
[2], $\lambda = 1$, intersection	0.29	0.84	0.57
Model 2	0.28	0.67	**0.81**
Model 2, intersection	**0.18**	**0.89**	0.74

Table 1: Alignment performance, as measured by AER, precision, and recall, for the algorithms implemented.

References

[1] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[2] Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of NAACL-HLT*, pages 644–648, 2013.

[3] Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics, 2006.

[4] Robert C. Moore. Improving IBM word-alignment model 1. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 518. Association for Computational Linguistics, 2004.