CIS 526 Homework 1: Word Alignment

James Yang

January 29, 2015

1 Introduction

Natural language translation is an open and rather difficult problem to solve. While direct lexical translation is somewhat straightforward, ordering the translated words to make the translation seem more natural for a fluent speaker is a far more difficult task. In this homework, we were tasked with post-translation word alignment of a sample Canadian Hansard corpus. We were given the English transcript and an aligned French translation from which we were to train a model for word alignment. For this assignment, the IBM Model 1 alignment algorithm was implemented for this particular task. The IBM Model 2 alignment algorithm was also implemented, whose results were compared to Model 1.

2 IBM Model 1

The IBM Model 1 word alignment algorithm is an expectation maximization (EM) algorithm that attempts to maximize the lexical translation probability assuming a uniform prior for alignment reordering. More formally, it attempts to maximize the translation probability of a source sentence $\mathbf{e}$ with a specific alignment a conditioned on a foreign sentence $\mathbf{f}$, $p(\mathbf{e}, a|\mathbf{f})$ assuming a uniform prior on alignment. Hence, we must compute $p(a|\mathbf{e}, \mathbf{f})$ by expanding it using the chain rule:

$$p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f}}{p(\mathbf{e}|\mathbf{f})}$$

where

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e}, a|\mathbf{f}) \\ &= \sum_{a(1)=0} \cdots \sum_{a(l_e)=0} p(\mathbf{e}, a|\mathbf{f}) \\ &= \sum_{a(1)=0} \cdots \sum_{a(l_e)=0} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \\ &= \frac{\epsilon}{(l_f+1)^{l_e}} \sum_{a(1)=0} \cdots \sum_{a(l_e)=0} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \\ &= \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) \end{aligned}$$

Using this definition of $p(\mathbf{e}|\mathbf{f})$, we find with substitution that

$$p(a|\mathbf{e}, \mathbf{f}) = \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_e} t(e_j|f_i)}$$

We use this formula as the expectation step, or **E step**, of the EM algorithm.

For the maximization step, or **M step**, of the EM algorithm, we define a count function that collects evidence from a sentence pair $(\mathbf{e}, \mathbf{f})$ that an input word f translates into its corresponding output word e:

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta_{e,e_j} \delta_{f,f_{a(j)}}$$

where $\delta_{i,j}$ is the Kronecker delta function.

From the counts we estimate a new translation probability:

$$t(e|f;\mathbf{e},\mathbf{f}) = \frac{\sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f})}{\sum_e \sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f})}$$

Using this model, an AER (Alignment Error Rate) of 0.30 was achieved. Convergence was determined as having maximized our expectation with six iterations.

3 IBM Model 2

IBM Model 2 is effectively IBM Model 1 with the notion that all possible re-orderings of a translation are not uniform. Upon maximizing lexical translation probability, Model 2 adds an explicit model for alignment $a(i|j,l_e,l_f)$ where an input sentence pair $(\mathbf{e},\mathbf{f})$ has sentence lengths l_e and l_f respectively. Using a Bayesian approach to a maximum *a-priori* implementation, we now have a new alignment probability:

$$p(\mathbf{e},a|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)})a(a(j)|j,l_e,l_f)$$

With this equation, we remodel our translational probability:

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e},a|\mathbf{f}) \\ &= \epsilon \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=1}^{l_f} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})a(a(j)|j,l_e,l_f) \\ &= \epsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_{a(j)})a(a(j)|j,l_e,l_f) \end{aligned}$$

Our new count function becomes

$$\begin{aligned} c(e|f;\mathbf{e},\mathbf{f}) &= \sum_{j=1}^{l_e} \sum_{i=0}^{l_f} \frac{t(e|f)a(a(j)|j,l_e,l_f)\delta_{(e,e_j)}\delta_{(f,f_i)}}{\sum_{i'=0}^{l_f} t(e|f_{i'})a(i'|j,l_e,l_f))} \\ &= \frac{t(e_j|f_i)a(a(j)|j,l_e,l_f)}{\sum_{i'=0}^{l_f} t(e_j|f_{i'})a(i'|j,l_e,l_f))} \end{aligned}$$

Using this model, an AER (Alignment Error Rate) of 0.2853, a slight improvement on Model 1. Convergence was determined as having maximized our expectation with ten iterations.

4 Conclusion

It is clear that with increased model complexity, there was an overall improvement in AER translation score. It is generally assumed that with increased model complexity, translation would also improve. The improvement between Model 1 and Model 2, while meager, makes sense as it does not make sense to assume a uniform distribution between alignments. For instance, the French word "le" typically has a bijective mapping to the English word "the," whereas a word like "voler" in French can mean two different actions in English. Thus, the chances that the word is one or the other depends purely on context and the frequency of that context.

Further work on word alignment would involve implementing higher order IBM models with different priors as well as adding features such as POS, alternate dictionaries, more aligned data, and other contextual language cues.