IBM Model 1, IBM Model 2, HMM Alignment Models for Statistical Machine Translation

Akshay Srivatsan, Ian Mukherjee, Nathan Smith

February 25, 2016

1 Introduction

Our group incrementally implemented three translation alignment algorithms beginning with the IBM Model 1 described in the assignment guidelines. The pseudocode for Model 1 presented in Professor Koehn's textbook was used as a template papers on advanced alignment algorithms. It was only fitting, then, to begin researching our own model by reconstructing the code for the IBM Model 2. A detailed mathematical and theoretical discussion of each translation alignment algorithm, as well as their accuracies at varying numbers of iterations and sentences, are as follows:

2 IBM Model 1:

The IBM Model 1 is a simple generative model where each output word e produced from an input word f is determined by the translation probability $p(e \mid f)$. We initialize our model by setting $p(e \mid f)$ to be uniformly distributed, with the probability of each being equal to $\frac{1}{len(english\ vocab)}$. We then iterate through each sentence pair (e, f), and for each english word e we iterate through all french words in the sentence pair, summating the value $s\_total[e]$. Next, we iterate through all english and french words, and summate $count(e \mid f)$, collecting evidence for each sentence pair (e, f) that a particular input word f translates into the output word e. Then, at the end of each iteration we re-estimate the probabilities of French words f translating to English words e, where the probability $t(e \mid f)$ is the $count(e \mid f)$ divided by the total $t(e \mid f)$ for that french word f translating to any english word e.

IBM Model 1 results with 10 iterations on 10,000 lines:

Precision	0.555219
Recall	0.730769
AER	0.38585

3 IBM Model 2:

The IBM Model 2 takes the translational basis of IBM Model 1 and adds a model for alignment. While in Model 1 the probabilities for any french and english sentences sharing the same respective words are equal, Model 2's explicit alignment model specifies alignment probabilities depending on the relative positions of the words. Thus, a new alignment probability distribution $a(i, j, len(fs), len(es))$, where i represents the position of the French word, j the position of the English word, and the lengths of the French and English sentences, respectively. This probability distribution is initialized uniformly to be $\frac{1}{len(es)+1}$. When iterating through all sentence pairs, we now accumulate $s\_total[e]$, the jth English word, by multiplying our previous tranlsation probability $t(e \mid f)$ by the alignment probability $a(i \mid j, len(fs), len(es))$. The program then iterates through all word positions in the English and French sentences, with counts summating $\frac{\left(t(e|f) \times a(i,j,len(fs),len(es))\right)}{s\_total[e]}$, again identical to the count in IBM Model 1 being multiplied by the alignment probability of the two word positions. The t and a functions are then reset for all words and word positions by dividing count by total, and the program enters the next iteration, continuing this process until convergence.

IBM Model 2 results with 10 iterations on 10,000 lines:

Precision	0.647504
Recall	0.789941
AER	0.304304

IBM Model 2 results with 10 iterations on the full dataset:

Precision	0.732224
Recall	0.816568
AER	0.239239

4 HMM Model:

We enhanced IBM Model 2 using an HMM approach, based on the "HMM-Based Word Alignment in Statistical Translation" article by Stephan Vogel, Hermann Ney and Christoph Tillman. Their hypotheses stem from a similar speech recognition problem in time alignment, in which the alignment of the current word is weighted by a dependence on the previous word in the sentence, denoted $p(a_j \mid a_{j-1}, L)$, where L is the length of the English sentence. By extending the preceding notion into translation as opposed to recognition, the authors show that the model can accurately predict the alignment of a given English word to a foreign word. We built off of these ideas to create our own variation on the model.

We train the HMM model using Expectation Maximization, just like the IBM models. Much like the previous model, we initialize the translation probabilities to $\log \frac{1}{len(english\ vocab)}$ as well as each words' alignment probabilities trivially to $\log \frac{1}{(len(fs)}$. In the expectation step, instead of computing the alignment counts with respect to word location in the two sentences, we pool counts for any alignments with the same distance between the location of the foreign word and the location of the foreign word aligned to the previous english word. In other words, the alignment probabilities are solely dependent on the "jump width," or the distance between the current and previous words. This means that during the maximation step, we update the alignment probabilities as follows:

$$p(i \mid i', L) = \frac{s(i - i')}{\sum_{l=1}^{L} s(L - i')}$$

Under this model, computing the most probable alignment for any given word is computationally expensive. We optimize this by using the following dynamic programming approach:

$$Q(i, j) = p(f_j \mid e_i) \times \max_{i'=1,\ldots,t} [p(i \mid i') \times Q(i', j - 1)]$$

Once normalized, this will give us the probability of alignment for any English word to any word in the sentence, taking into account the alignment probabilities of all previous words in the english sentence.

On a side note, as our group tested the HMM, we found that some of our alignment probabilities converged to 0. To solve this, we stored probabilities internally as log-probabilities, which prevents underflow.

HMM results for 30 iterations on 10,000 lines:

Precision	0.641452
Recall	0.751479
AER	0.321321

5 Bibliography:

References

[1] Stephan Vogel, Hermann Ney, Christoph Tillmann HMM-Based Word Alignment in Statistical Translation

[2] Chi-Ho Li, Xiaodong He, Yupeng Liu, Ning Xi Incremental HMM Alignment for MT System Combination