CS224N Problem Assignment 1

Enhao Gong, Yuming Kuang

October 8, 2014

1. Word Alignment

1.1 Result

The AER result of PMI/Model 1/Model 2 on different languages for both *dev* set and *test* set with training size 10k sentences is summarized in Table 1 and Table 2.

	French-English	Hindi-English	Chinese-English
PMI	0.7327	0.8546	0.8361
Model 1	0.3674	0.5847	0.5903
Model 2	0.3042	0.5851	0.5718

Table 1: AER results on *dev* set, training sentences 10k

	French-English	Hindi-English	Chinese-English
PMI	0.7129	0.8102	0.8273
Model 1	0.3546	0.5857	0.5917
Model 2	0.2946	0.5860	0.5603

Table 2: AER results on *test* set, training sentences 10k

1.2 Implementing and Developing Details

- **Initialization:** In IBM Model 1, we initialized $t(e_i|f_j)$ uniformly. This is simply setting $t(e_i|f_j)$'s all to 1 in the 1st iteration, since they are essentially $\frac{1}{C}$ while C is a constant and will disppear in normalization. Besides, since in the M step we have $\hat{a_i} = \frac{q(a_i|i,n,m)t(e_i|f_{a_i})}{\sum_j q(j|i,n,m)t(e_i|f_j)} = \frac{t(e_i|f_{a_i})}{\sum_j t(e_i|f_j)}$ which is independent of the q's, so we don't have to do anything for q's. In IBM Model 2, we initialized $t(e_i|f_j)$ using the estimate of IBM Model 1. As for $q(j|i,n,m)$'s, we initialized them randomly. In detail, before the EM iterations, we screening all the sentence pairs in the training set, and for all positions (j,i,n,m) encountered we set $q(j|i,n,m)$ a random real number from $Uniform(0,1)$. Then a normalization is performed.

- **Convergence Criterion:** We define $D_t = mean\{|t^{old}(e|f) - t^{new}(e|f)|, t^{old}(e|f) \neq 0 \text{ or } t^{new}(e|f) \neq 0\}$ and $D_q = mean\{|q^{old}(j|i,n,m) - q^{new}(j|i,n,m)|, q^{old}(j|i,n,m) \neq 0 \text{ or } q^{new}(j|i,n,m) \neq 0\}$, which is the mean change of non-zero elements for $t(e|f)$'s and $q(j|i,n,m)$'s. So our convergence criterion is for IBM Model 1, $D_t \leq \tau$, and for IBM Model 2, $D_t \leq \tau$ and $D_q \leq \tau$. Besides we also set a maximal iteration number 1000. In our implementation we choose $\tau = 10^{-3}$. Table 3 gives the results of different τ choice on French-English *dev* set. We can see that if τ is smaller, AER result becomes better but more iterations thus longer training time are introduced.

	Model 1			Model 2		
τ	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}
Iterations	2	6	16	3	8	24
AER	0.4034	0.3674	0.3554	0.3139	0.3042	0.2879

Table 3: AER result for different τ's on French-English *dev* set for IBM Model 1/2

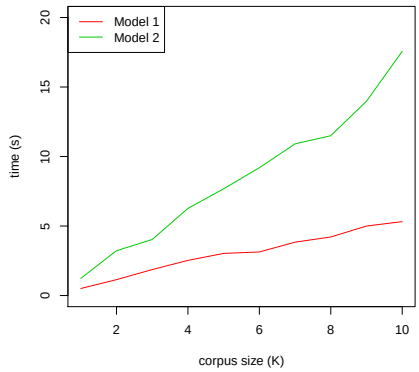

Figure 1: Runtime v.s training corpus size for IBM Model 1/2 on French training set

- **Runtime:** On our settings, the runtime for training IBM Model 1 on 10K (3441 for Hindi) sentences is: 31.13s (French), 12.67s (Hindi), 100.03s (Chinese). The runtime for training IBM Model 2 on 10K sentences is: 147.99s (French), 41.96s (Hindi), 462.03s (Chinese). We also explored the runtime as a function of corpus size. The Figure 1 shows the runtime per iteration on French-English alignment versus training corpus size. We can clearly see that there's a linear relationship, approx. per iteration 0.5215s/1K corpus for Model 1 and 1.667s/1K corpus for Model 2.

1.3 Error Analysis

We are native speakers of Chinese so we choose Chinese-English alignment result for error analysis. Figure 2 shows a few examples we want to use to illustrate our analysis.

- **Word order difference:** From the AER result, we clearly see that IBM models perform better in French-English alignment than Hindi and Chinese, and IBM Model 2 delivers large improvement in French, but little in Hindi and Chinese. That is because French and English share similar grammar structure, so that alignments concentrate on near diagonal places. Thus model 2 can learn to give $q(j|i,n,m)$'s near diagonal larger weights, and reach better result. However for Hindi and Chinese, it's very possible that the alignments are not near diagonal places and not context-free because of different grammar structure. So uniform $q(j|i,n,m)$ is already a good choice. Take Chinese-English alignment result as example, success alignment in different word order often appears when two words share the same meaning, like in Example 1 ('earthquake forecasting cause', '地震预报事业'), ('unique and effective', '独特有效'), in Example 2 ('answer', '答案'), ('i', '我'). while failure occurs in longer structure like clause. For example, in Example 4, 'it develops' is an attributive clause that will be put behind in Enlish, but its counterpart '其建立的' in Chinese will be put ahead, and IBM models are not able to figure it out.

- **Garbage collection:** For Chinese-English alignments, function words are easier to be aligned correctly in shorter sentences, like in Example 3 'the', 'a' and 'of' are all aligned correctly, because there are not much

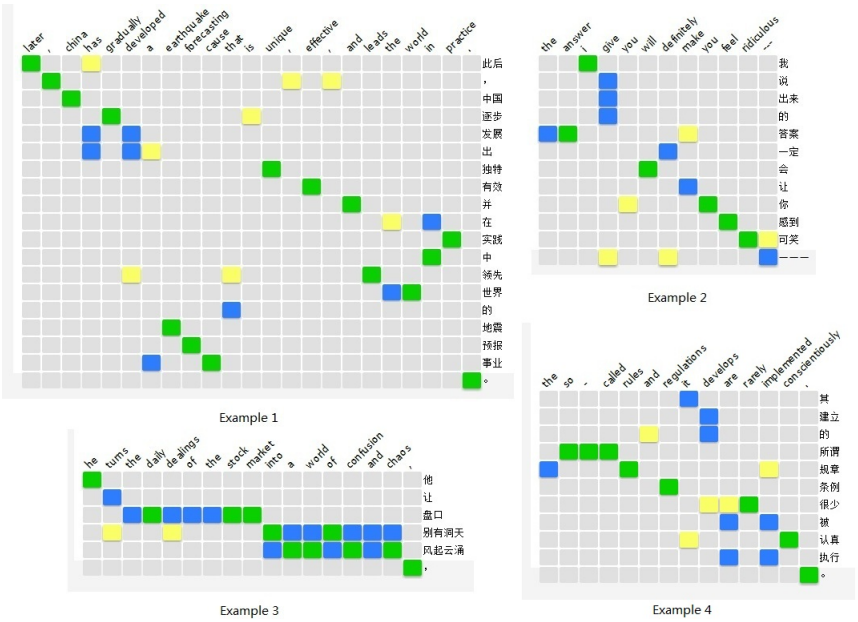

Figure 2: Examples of Chinese-English alignments. (Blue: gold alignment, Yellow: our alignment, Green: correct alignment)

ambiguity. However, in longer sentences like in Example 1, function words are harder to align due to more possible alignment and word difference introduced by complex structure.

- **Common errors:** The most common errors are from phrase alignment. Our model only consider word-to-word alignment, although it allows many-to-one mapping, it can't deal with one-to-many and many-to-many cases. For example, the model can figure out many-to-one case ('the daily dealings of the stock market', '盘口') in Example 3. However for many-to-many case ('into a world of confusions and chaos', '别有洞天风起云涌'), our model can only figure out half the alignment. Even worse, for one-to-many case like in Example 2 ('give', '说出来的') our model totally fails in this example.