Word Alignment

Yating Jing

February 19, 2015

1 Abstract

The word alignment task is conducted on a bilingual dataset (French-English). Three popular methods are implemented: IBM Model 1, IBM Model 2, and the Fast Align Model (log-linear reparameterization of IBM Model 2).

All three models are trained using **EM algorithm**. First make assumptions on the unknown probability distributions and train the model parameters, the use the model to collect counts and update the probabilities, and iterate for several rounds. The best performance comes from **a combination of IBM Model 1 and the Fast Align Model**.

2 IBM Model 1

2.1 Motivation

IBM Model 1 uses only lexical translation probabilities without considering the locations and relations of the words. This method breaks up the given sentence, model the smaller steps with lexical translation probability distributions. And it makes a strong assumption that the words are not correlated, which results in a relatively low performance compared with other advanced models.

2.2 Model Description

The main components of this model include **word translation probability** $t(e|f)$ and **alignment function** a, where e denotes an English word and f denotes a French word. Function a projects each English word at position j to a French word at position i:

$$a : j \to i$$

EM Algorithm First initialize the alignment probabilities uniformly, then collect counts of $< e, f >$ pairs to estimate the model parameters, then use the model to reestimate the probabilities, iterate till convergence.

The translation probability for a French sentence $\mathbf{f}$ of length l_f and an English sentence $\mathbf{e}$ of length l_e with any alignment function $a : j \to i$ is defined as:

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

The probability of an alignment a given sentence $\mathbf{f}$ and $\mathbf{e}$ is then:

$$\begin{aligned} p(a \mid \mathbf{e}, \mathbf{f}) &= \frac{p(\mathbf{e}, a \mid \mathbf{f})}{p(\mathbf{e} \mid \mathbf{f})} \\ &= \prod_{j=1}^{l_e} \frac{t(e_j \mid f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j \mid f_i)} \end{aligned} \tag{2}$$

Our goal is to find the assignment of $\hat{\mathbf{a}}$ with the highest probability:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{arg\,max}\ p(\mathbf{a} \mid \mathbf{e}, \mathbf{f}) \tag{3}$$

For this model, we need to go through all the possible assignments of each word considering the lexical translation probabilities:

$$\hat{a_j} = \underset{a_j}{arg\,max}\ t(e_j \mid f_{a(j)}) \tag{4}$$

Lexical translation probabilities can be obtained by fractional count collection:

$$\begin{aligned} t(e|f; \mathbf{e}, \mathbf{f}) &= \frac{count(< e, f >)}{\sum_e count(< e, f >)} \\ &= \frac{\sum_{<\mathbf{e},\mathbf{f}>} c(e \mid f; \mathbf{e}, \mathbf{f})}{\sum_e \sum_{<\mathbf{e},\mathbf{f}>} c(e \mid f; \mathbf{e}, \mathbf{f})} \end{aligned} \tag{5}$$

where

$$c(e \mid f; \mathbf{e}, \mathbf{f}) = \frac{t(e_j \mid f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j \mid f_i)} \sum_{j=1}^{l_e} \delta(e, e_j)\delta(f, f_{a(j)}) \tag{6}$$

The Kronecker delta function $\delta(x, y)$ is 1 if $x = y$ and 0 otherwise.

2.3 Results

Running 5 iterations of EM using IBM Model 1 on the entire dataset gives an AER of 0.414575.

3 IBM Model 2

3.1 Motivation

Besides taking lexical translation probabilities into consideration, IBM Model 2 adds **absolute reordering model**, which explicitly models the distribution of alignment. It is reasonable to consider the positions of the input and output words, and it makes the following assumption: **Words at similar positions in each sentence are more likely to be the translations of each other.** Though this could lead to overparameterization, it does outperform IBM Model 1.

3.2 Model Description

This model is implemented after 5 iterations of EM with IBM Model 1, and it initializes the word translation probabilities $t(e \mid f)$ directly from the results of Model 1. The **alignment probability distribution** is denoted by $a(i \mid j, l_e, l_f)$. Then IBM Model 2 can be represented as:

$$\begin{aligned} p(\mathbf{e} \mid \mathbf{f}) &= \sum_a p(\mathbf{e}, a \mid \mathbf{f}) \\ &= \epsilon \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j \mid f_{a(j)}) a(a(j) \mid j, l_e, l_f) \end{aligned} \tag{7}$$

The calculation of fractional counts of lexical translation for model 2 becomes:

$$c(e \mid f; \mathbf{e}, \mathbf{f}) = \sum_{j=1}^{l_e} \sum_{i=0}^{l_f} \frac{t(e \mid f) a(a(j) \mid j, l_e, l_f) \delta(e, e_j) \delta(f, f_i)}{\sum_{i'=0}^{l_f} t(e \mid f_{i'}) a(i' \mid j, l_e, l_f)} \tag{8}$$

and the formula of the counts for alignment is:

$$c(i \mid j, l_e, l_f; \mathbf{e}, \mathbf{f}) = \frac{t(e_j \mid f_i) a(a(j) \mid j, l_e, l_f)}{\sum_{i'=0}^{l_f} t(e_j \mid f_{i'}) a(i' \mid j, l_e, l_f)} \tag{9}$$

3.3 Results

Running 5 iterations of EM using IBM Model 1 plus 5 iterations of EM using IBM Model 2 on the entire dataset gives an AER of 0.276676.

4 Fast Align - Reparameterization of IBM Model 2

4.1 Motivation

This model is an efficient log-linear reparameterization of IBM Model 2. The model gives higher values to positions close to diagonal, which makes sense since given a French word f, the corresponding English word e is probably in the similar position in the sentence. It overcomes the overparameterization issue of IBM Model 2. However, in this model the probability to align word e to word f is still independent from the choices of other English words.

4.2 Model Description

This model is trained after IBM Model 1. It utilized some intermediate parameters produced by Model 1. First run 3 iterations on IBM Model 1, then pass the translation probabilities $t(e \mid f)$ on to the Fast Align Model as its initial probabilities, and collect the alignment distributions $\delta(a(j) = i \mid j, l_e, l_f)$ and reestimate the model, run for 3 iterations.

The model favors smaller diagonal distances, which are defined by:

$$h(i, j, l_e, l_f) = \left| \frac{i}{l_f} - \frac{j}{l_e} \right| \tag{10}$$

Alignment probability distribution is:

$$\delta(a(j) = i \mid j, l_e, l_f) = \begin{cases} p_0 & \text{if } i = 0 \\ (1 - p_0)\frac{e^{-\lambda h(i,j,l_e,l_f)}}{Z_\lambda(j,m,n)} & \text{if } 0 < i \leq l_e \end{cases} \tag{11}$$

where the normalization term is

$$Z_\lambda(j, m, n) = \sum_{i'=1}^{l_f} e^{-\lambda h(i',j,l_e,l_f)}$$

Model paramters The scaling factor λ controls how strongly the model favors positions close to the diagonal, set $\lambda = 20$, and the null alignment probability p_0 is 0.08. I chose the parameters λ and p_0 by fixing one of the parameters then sampling different values and trying to find an optimum while running the model on 1000 pairs of sentences.

Table 1: Model Parameters

λ	AER	p_0	AER
19	0.375	0.06	0.370
20	0.370	0.08	0.366
21	0.372	0.10	0.367
$p_0 = 0.0001$		$\lambda = 20$	

The count collection of the translation probabilities $t(e \mid f)$ is the same as the models above. The count collection of alignment distributions $\delta(a(j) = i \mid j, l_e, l_f)$ is:

$$c(i \mid j, l_e, l_f; \mathbf{e}, \mathbf{f}) = \frac{t(e_j \mid f_i)\delta(a(j) = i \mid j, l_e, l_f)}{\sum_{i'=0}^{l_f} t(e_j \mid f_{i'})\delta(i' \mid j, l_e, l_f)} \tag{12}$$

4.3 Results

Running 3 iterations of EM using IBM Model 1 plus 3 iterations of EM using Fast Align Model on the 10000 sentences gives an AER of 0.289896.

5 Summary

Due to the time limit, I was not able to finish running the third model on the entire dataset. From the tables below, we can conclude that IBM Model 1 + IBM Model 2 outperforms IBM Model 1, and the IBM Model 1 + Fast Align Model has the best performance.

Table 2: Model Performance on the entire dataset

Model	Precision	Recall	AER
IBM Model 1	0.488	0.843	0.415
IBM Model 1 + IBM Model 2	0.663	0.852	0.277

Table 3: Model Performance on 10000 sentences

Model	Precision	Recall	AER
IBM Model 1	0.465	0.828	0.436
IBM Model 1 + IBM Model 2	0.505	0.852	0.399
IBM Model 1 + Fast Align Model	0.656	0.825	0.289

References

[1] Philipp Koehn. *Statistical machine translation.* 2009. Cambridge University Press.

[2] Graa, Joao V., Kuzman Ganchev, and Ben Taskar. *Expectation maximization and posterior constraints.* 2007.

[3] Dyer, Chris and Chahuneau, Victor and Smith, Noah A. *A Simple, Fast, and Effective Reparameterization of IBM Model 2. HLT-NAACL.* 2013.