CS224N Statistical Machine Translation

Jiayuan Ma
jiayuanm@stanford.edu

Xincheng Zhang
xinchen2@stanford.edu

October 10, 2013

1 Word Alignment

1.1 IBM Model 1 & 2

Since the $q(\cdot)$ parameter in Model 1 has a very simple form

$$q(a_i|i,n,m) = \frac{1}{m+1} \tag{1}$$

we have

$$\begin{aligned} &\text{argmax}_{a_1,\ldots,a_n} p(a_1,\ldots,a_n|f_1,\ldots,f_m,e_1,\ldots,e_n,n) \\ =&\text{argmax}_{a_1,\ldots,a_n} \prod_{i=1}^{n} q(a_i|i,n,m)t(e_i|f_{a_i}) \\ =&\frac{1}{(m+1)^n}\prod_{i=1}^{n}\text{argmax}_{a_i} t(e_i|f_{a_i}) \end{aligned} \tag{2}$$

Therefore, we can have alignment variables $\{a_i\}_{i=1}^n$ totally independent of $q(\cdot)$ parameters. During EM iterations, we should only keep track of $t(\cdot)$ parameters, which are just the normalized counts of different words' cooccurences. The pseudocode of IBM Model 2 is in Algorithm 1, where we use probabilistic counts $\delta(\cdot)$ to estimate $t(\cdot)$ and $q(\cdot)$.

1.2 Implementation Detail

In Model 1, we uniformly initialize the parameters $t(\cdot)$. In Model 2, we initialize the translation parameters $t(\cdot)$ using the results of Model 1, and we have two different initialization strategies for the position parameters $q(\cdot)$, **random** and **diagonal** initialization. Random initialization randomly chooses the initial parameters $q(\dot{)}$, and normalize it appropriately to make sure that $q(\cdot)$ is a valid conditional probability. Diagonal initialization is inspired by [1]. Since it is reasonable to assume that words appear around the same relative positions should be aligned together, we have

$$q(j|i,n,m) = \begin{cases} p_0 & j = -1 \\ (1-p_0)\times\frac{e^{-\lambda h(i,j,n,m)}}{Z_\lambda(i,n,m)} & 0 \le j \le m \\ 0 & \text{otherwise} \end{cases} \qquad h(i,j,n,m) = \left|\frac{i+1}{n} - \frac{j+1}{m}\right| \tag{3}$$

Algorithm 1 IBM Model 2

Input: A training corpus $\{(f^{(k)}, e^{(k)})\}_{k=1}^{n}$
Initialize $t(e|f)$ using Model 1's result and $q(\cdot)$ parameters using methods in section 1.2.
for iter $= 1 \dots T$ **do**
  Set all counts $c(\dots) = 0$
  `// For each training sentences`
  for $k = 1 \dots n$ **do**
    `// For each position in target sentences`
    for $i = 1 \dots n_k$ **do**
      $Z_i \leftarrow \sum_{j'=1}^{m_k} q(j'|i, n_k, m_k) t(e_i^{(k)}|f_{j'}^{(k)})$ `// Partition function`
      `// For each position in source sentences`
      for $j = 1 \dots m_k$ **do**
        $\delta(k, i, j) \leftarrow \frac{q(j|i,n_k,m_k)t(e_i^{(k)}|f_j^{(k)})}{Z_i}$
        $c(e_i^{(k)}, f_j^{(k)}) \leftarrow c(e_i^{(k)}, f_j^{(k)}) + \delta(k, i, j)$
        $c(j, i, n_k, m_k) \leftarrow c(j, i, n_k, m_k) + \delta(k, i, j)$
      end for
    end for
  end for
  Normalize to obtain $t(e|f) = \frac{c(e,f)}{c(f)}$ $q(j|i, n, m) = \frac{c(j,i,n,m)}{c(i,n,m)}$
  Check convergence using methods in section 1.2
end for

This initialization is parameterized by a null alignment probability p_0 and $\lambda \geq 0$ which controls how strongly the model favors alignment points close to the diagonal. When $\lambda \to 0$, the initialized distribution approaches $q(\cdot)$ in Model 1. When λ gets larger, the model is initialized to be less likely to deviate from a perfectly diagonal alignment, which is especially helpful for some particular language pairs (such as French-English). For more discussion, please see section 1.3.

To check convergence between iterations, we calculate the ℓ_∞ distance between the parameters in two successive runs. If one $\|\cdot\|_\infty$ is smaller than a predetermined thresold, the algorithm will terminate. Otherwise, it will only terminate util it reaches the maximum number of iterations.

For code efficiency, we encode triplets $\langle i, n, m \rangle$ into one integer so that we can use `CounterMap` in the skeleton code with primitive `int` types. Since i, n and m are small non-negative integers, we choose to use two successive *Cantor mapping* to do the encoding, which proves to be quite efficient.

1.3 Results and Discussions

The AER results of PMI/IBM1/IBM2 models on different language pairs are available in Table 1 (development set) and Table 2 (test set). We train our models using 10k sentence pairs (except for Hindi, which has only 3441 sentence pairs in total) with the maximum iteration number being 300 (AER won't change too much after 300 runs) and diagonal initialization for IBM2. Our PMI models take less than one minute to run, IBM1 models finish within five minutes. For our IBM2 models, it takes around 15 minutes to run 100 iterations.
In general, the performance of IBM2 is better than that of IBM1, whose performance is better than PMI's performance. An interesting observation here is Model 2 has significant improvement over

Dev Set	French-English	Hindi-English	Chinese-English
PMI	0.7327	0.8546	0.8361
Model1	0.3524	0.5847	0.5836
Model2	0.3129	0.5885	0.5634

Table 1: Different models' Alignment Error Rate (AER) on development sets

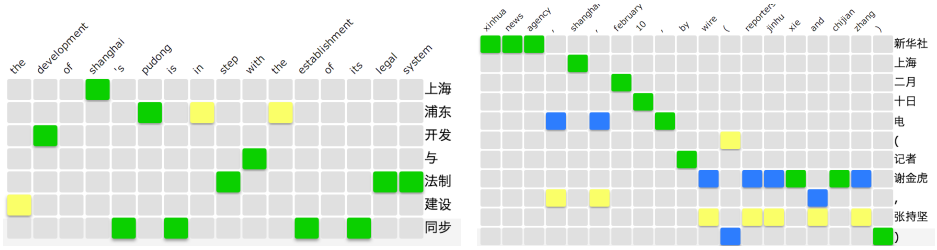
Test Set	French-English	Hindi-English	Chinese-English
PMI	0.7129	0.8102	0.8273
Model1	0.3496	0.5786	0.5857
Model2	0.2858	0.5777	0.5710

Table 2: Different models' Alignment Error Rate (AER) on test sets

Model 1 in French-English alignment, and the performance is higher (from 0.30 to 0.28) when using **diagonal** initialization in section 1.2 with large λ. This might be due to the fact that French and English words are comparatively well aligned with respect to their locations in the sentence.
For Hindi and Chinese, the word order changes are more significant than French, which explains why IBM2 gives much less improvement over IBM1 (than in French-English case). In both cases, using random initialization gives worse performance than using diagonal initialization with a small λ (flat probabilities). This is because random strategies may give us a bad local minima, while the flat stategies probably avoid these local minima by constraining EM to start from a IBM1 setup.
In the case of Hindi, using random initialization with IBM2 results in a worse performance than IBM1. This is because we don't have enough training data for Hindi, so that IBM2 with more parameters is more likely to overfit. Therefore, starting from a uniform distribution of $q(\cdot)$ is a good way to compensate inadequate training data in Hindi, but still IBM2 gives very little (almost no) performance boost over IBM1 when aligning Hindi with English.

1.4 Error Analysis and Discussions

Since both of us are native Chinese speakers, we focus ourselves on analyzing Chinese-English alignment. We observe that the alignment tables (see Figure 1) for Chinese-English are less concerntrated on the diagonal than French-English alignment, which means word orders do change a lot. However, our models seem to be capable of dealing with a vast range of word order differences. Two examples Chinese-English alignment are shown in Figure 1. Example 1 has successful aligned phrase pairs such as "shanghai" (correct), "pudong"(correct), "development"(correct), "with"(correct) and "legal system" (almost correct). The algorithm misaligns "establishment" and the last Chinese phrase. "Be in step with" together translates the last Chinese phrase. Because we model "be in step with" as four independent words, it is quite difficult for the algorithms to find the correct alignment.
What is interesting in Example 1 is that the alignment algorithm successfully aligned the location names, although they do have non-trivial word order changes. This observation leads to Example 2 which includes both location names and person's names. Person's names are more difficult to align because it is very unlikely that the same names will appear several times in the corpus and characters in names can also be used in regular phrases. In Example 2, both algorithms (IBM1 and



(a) Example 1 (b) Example 2

Figure 1: Two examples of Chinese-English alignment. Blue (IBM1), yellow (IBM2), green(IBM1 & IBM2)

Systems	Baseline	Dimension Feature	Vowel Feature	Derivational Feature
BLEU Score	14.954	15.098	15.219	15.173

Table 3: BLEU scores on baseline and different systems

IBM2), not surprisingly, failed at aligning both names ("jinhu xie" and "chijian zhang"). However, the mistake is somewhat tolerable because it approxmiately swapped the correct alignment for those names! This makes us believe that the algorithms succeeded in recognizing that these phrases are person's names. The algorithm failed at producing correct alignment simply because it did not see any enough repetitions of those names in the training data.

The word alignment model does sometimes align function words (such as *the*, *a*, and *of*) with content words. For example, in Example 1, the model aligned "the" to the phrase "establishment" in Chinese.

The most common alignment errors in Chinese-English are failures of aligning Chinese words with English phrases with multiple words. Most Chinese words just contains two characters, while the equivalent English phrases can span several words. More importantly, these equivalent English phrases are not necessarily adjacent to each other, like"as far as ... be concerned", which adds to the difficulty of alignment.

2 MT Features (with Extra Credit)

The BLEU scores for our experiments is in Table 3. Since BLEU scores vary between each run, all these results are averaged across three different runs for reliability. We tried the following features:

- *Dimension feature* We add source/target rule dimensions as indicator features.
- *Vowel feature* We add the number of vowel alphabets' appearances in source-target pairs as two-dimensional indicator features.

- *Derivational rule feature* We add the average target length appeared in the previous derivational rules as numerical features.

The best and most stable feature among those three is the vowel feature, which gives a BLEU score boost between 0.1 and 0.3. This is a language-specific feature, since we observe that some of the French words look very similar to the English words and the number of syllables tend to be very similar between these two languages. We further observe that the number of vowel alphabets (`'a', 'e', 'i', 'o', 'u'`) is correlated with the number of syllables. Devising a two-dimensional indicator feature for the number of vowel alphabets appearing in target and source rules is reasonable.

We have another feature of source/target dimensions, which just counts the number of words in both sides. This feature gives moderate improvement of BLEU score between 0.1 and 0.2.

Finally, we have a complex derivational feature (extra credit), which looks at the rules that have been applied and calculates the average target size of all those rules. The rationale behind this feature is that our average target language phrase lengths should give a rough estimation of the average phrase length of our target language, which is good complement to the language model.

2.1 Error Analysis

Vowel feature improves the translation by adding missing verbs. An example is as follows.

```
...according to the office of the czech statistical
confirmed to be a progressive decrease of economic growth.
...according to the office of the czech statistical
it is confirmed that we head toward a progressive reduction of economic growth.
```

Since we cannot speak French, we use Google Translate to help our error analysis (see attached text file `Google_Translate_Comparision.txt`). We sample 30 sentences from the data, and compare our translated results with Google's results.
Errors generally include grammatical errors, meaning distortion and inappropriate word usage.
There are 15 translations that have the same meaning with Google's translation, and they didn't have obvious errors for reading and understanding.
Two of our results have slight grammatical errors but they can still be understood by human readers, such as"is it" v.s. "it is".
For meaning distortion, three of our results dropped one adjective word, which do not affect a rough human understanding. Another three of our results missed some key words, which makes human difficult to guess their meanings. In two extremely bad cases, our translations simply do not make sense.
For improper word usage, there are five cases, but they are not very serious, such as "participate in the fire" instead of "participate in the shooting", "accept" instead of "agree with".

References

[1] Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL/HLT 2013*, pages 644–648, 2013.