Chapter 14

Word Alignment

14.1 Problem

A *parallel text* is a corpus of text that expresses the same meaning in two (or more) different languages. Usually we assume that a parallel text is already *sentence-aligned*, that is, it consists of *sentence pairs*, each of which expresses the same meaning in two languages. Conventionally, following Brown et al. (1993), the two languages are referred to as English and French even when other languages are possible. Here, we use English and Spanish.

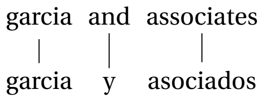
Here is an example parallel text:

1. garcia and associates
   garcia y asociados

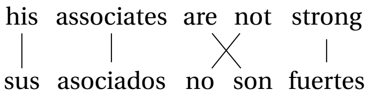
2. his associates are not strong
   sus asociados no son fuertes

The *word alignment* problem is to figure out which Spanish words correspond to which English words. This would be the correct word alignment for our example:

1.



2.



More formally: let

- $\mathbf{f} = f_1 \cdots f_m$ range over Spanish sentences

- $\mathbf{e} = e_1 \cdots e_\ell$ range over English sentences

- $\mathbf{a} = (a_1, \ldots, a_m)$ range over possible many-to-one alignments, where $a_j = i$ means that Spanish word j is aligned to English word i, and $a_j = \text{NULL}$ means that Spanish word j is unaligned. Thus the alignment for sentence (2) above is $(1, 2, 4, 3, 5)$.

We are given a sequence of $(\mathbf{f}, \mathbf{e})$ pairs. We are going to define a model of $P(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$ and our job is to estimate the parameters of the model to maximize the likelihood $P(\mathbf{f} \mid \mathbf{e})$.

14.2 Model 1

IBM Model 1 (Brown et al., 1993) is the first in a series of five seminal models for statistical word alignment. The basic generative story goes like this:

1. Choose m with uniform probability $\epsilon = \frac{1}{M}$, where M is the maximum length of any Spanish sentence in the corpus.

2. Generate an alignment $a_1, \ldots, a_m$, again with uniform probability.

3. Generate Spanish words $f_1, \ldots, f_m$, each with probability $t(f_j \mid e_{a_j})$ or $t(f_j \mid \text{NULL})$.

The model is usually presented as the following equation:

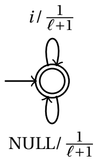
$$P(\mathbf{f} \mid \mathbf{e}) = \frac{1}{M} \prod_{j=1}^{m} \left(\frac{1}{\ell + 1} t\left(f_j \mid e_{a_j}\right) \right) \tag{14.1}$$

where M is the maximum length of any French sentence (say, 100). In practice, it doesn't actually matter what number you choose.

Here, we show how to break this down conceptually into a cascade of finite transducers. You may or may not find this helpful for Model 1, but for the more complex models (Model 2 and HMM), I think that the finite transducer formulation makes it much easier to work out what EM looks like.

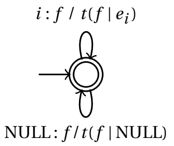
There isn't a one-size-fits-all finite-state machine that computes Model 1 for any sentence pair. Instead, for each $\mathbf{e}$, we can make a FSA that generates French sentences $\mathbf{f}$ according to the Model 1 probability $P(\mathbf{f} \mid \mathbf{e})$. It is a cascade of steps 2 and 3 above.

The alignment-generation model can be written as a very simple FSA:



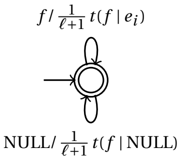
The arc labeled i is actually many arcs, one for each $1 \le i \le \ell$. Notice that because m is determined beforehand, this FSA doesn't need a stop probability.

The Spanish-word-generation model is also very simple:


$i : f \ / \ t(f \mid e_i)$
NULL : $f / t(f \mid \text{NULL})$

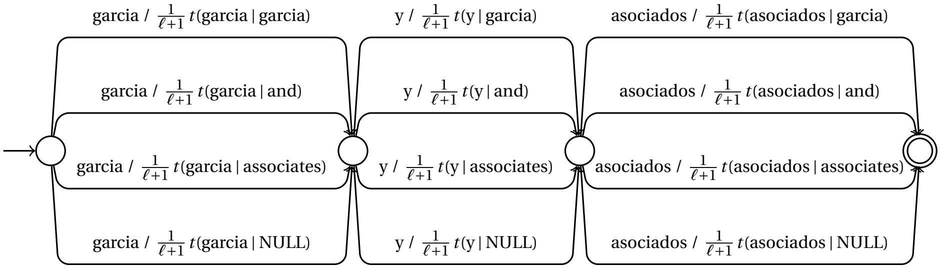
The arc labeled $i : f$ is actually many arcs, one for every Spanish word f and every English position $1 \le i \le \ell$.

Composing the two, we get IBM Model 1.


$f / \frac{1}{\ell+1} t(f \mid e_i)$
$\text{NULL}/ \frac{1}{\ell+1} t(f \mid \text{NULL})$

The arc labeled f is actually many arcs, one for every Spanish word f and every English position $1 \le i \le \ell$. This automaton can generate any Spanish string $\mathbf{f}$ with any alignment to $\mathbf{e}$, and the probability of the path is $P(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$. Note that although we can generate any Spanish string, the English string $\mathbf{e}$ remains fixed.

On intersecting this FSA with $\mathbf{f}$, we get:


garcia / $\frac{1}{\ell+1}$ t(garcia | garcia)
y / $\frac{1}{\ell+1}$ t(y | garcia)
asociados / $\frac{1}{\ell+1}$ t(asociados | garcia)
garcia / $\frac{1}{\ell+1}$ t(garcia | and)
y / $\frac{1}{\ell+1}$ t(y | and)
asociados / $\frac{1}{\ell+1}$ t(asociados | and)
garcia / $\frac{1}{\ell+1}$ t(garcia | associates)
y / $\frac{1}{\ell+1}$ t(y | associates)
asociados / $\frac{1}{\ell+1}$ t(asociados | associates)
garcia / $\frac{1}{\ell+1}$ t(garcia | NULL)
y / $\frac{1}{\ell+1}$ t(y | NULL)
asociados / $\frac{1}{\ell+1}$ t(asociados | NULL)

Then the $t(f \mid e)$ can be estimated using Expectation-Maximization.

1. Initialize $t(\cdot \mid e)$ to uniform: $t(f \mid e) = \frac{1}{|V_f|}$, where V_f is the Spanish vocabulary and e is any English word or NULL.

2. E-step: Reset all counts $c(f, e)$ to zero. For each sentence pair, use the Forward-Backward algorithm to calculate the expected number of times that word e is translated as f. Hard EM doesn't work very well here. But true EM is very easy to implement, because every path goes through every state. For each i, j, the transition that generates f_j from e_i "competes" with the transitions that generate f_j from the other English words (or NULL). So we update $c(f, e)$ as follows:

$$c(f_j, e_i) \leftarrow c(f_j, e_i) + \frac{t(f_j \mid e_i)}{\sum_{i'} t(f_j \mid e_{i'}) + t(f_j \mid \text{NULL})}$$
$$c(f_j, \text{NULL}) \leftarrow c(f_j, \text{NULL}) + \frac{t(f_j \mid \text{NULL})}{\sum_{i'} t(f_j \mid e_{i'}) + t(f_j \mid \text{NULL})}$$

3. M-step: let $t(f \mid e) \leftarrow \frac{c(f,e)}{\sum_f c(f,e)}$, where e is any English word or NULL.

4. Go to 2.

Interestingly, for this particular model, (true) EM is guaranteed to converge to a global maximum (although the global maximum is not unique).

Let's see how this works on our toy example. The initial model is uniform:

			f				
e	asociados	fuertes	garcia	no	son	sus	y
NULL	1/7	1/7	1/7	1/7	1/7	1/7	1/7
and	1/7	1/7	1/7	1/7	1/7	1/7	1/7
are	1/7	1/7	1/7	1/7	1/7	1/7	1/7
associates	1/7	1/7	1/7	1/7	1/7	1/7	1/7
garcia	1/7	1/7	1/7	1/7	1/7	1/7	1/7
his	1/7	1/7	1/7	1/7	1/7	1/7	1/7
not	1/7	1/7	1/7	1/7	1/7	1/7	1/7
strong	1/7	1/7	1/7	1/7	1/7	1/7	1/7

(Note that each row sums to one.)

E step: We want to know the *expected* (fractional) number of times that each French word is translated from each English word. Consdier sentence (1). First, for each English position i and French position j, compute the total probability of all alignments that translate f_j from e_i (that is, $P(a_j = i \mid \mathbf{e})$, which is just equal to $t(f_j \mid e_i)$):

		f_j	
e_i	garcia	y	asociados
garcia	1/7	1/7	1/7
and	1/7	1/7	1/7
associates	1/7	1/7	1/7

Then renormalize for each j to obtain the fractional count of times that f_j is translated from e_i (that is, $P(a_j = i \mid \mathbf{f}, \mathbf{e})$.

		f_j	
e_i	garcia	y	asociados
NULL	1/4	1/4	1/4
garcia	1/4	1/4	1/4
and	1/4	1/4	1/4
associates	1/4	1/4	1/4

Note that we're renormalizing so that the *columns* sum to one. This is because each French word occurs exactly once and is aligned to exactly one English word (or NULL). So for each French word, the fractional counts of which English word it's translated from should sum to one.

Similarly for sentence (2):

			f_j		
e_i	sus	asociados	no	son	fuertes
NULL	1/6	1/6	1/6	1/6	1/6
his	1/6	1/6	1/6	1/6	1/6
associates	1/6	1/6	1/6	1/6	1/6
are	1/6	1/6	1/6	1/6	1/6
not	1/6	1/6	1/6	1/6	1/6
strong	1/6	1/6	1/6	1/6	1/6

To complete the E step, we count up how many times each French word was translated from each English word:

			f				
e	asociados	fuertes	garcia	no	son	sus	y
NULL	5/12	1/6	1/4	1/6	1/6	1/6	1/4
and	1/4	0	1/4	0	0	0	1/4
are	1/6	1/6	0	1/6	1/6	1/6	0
associates	5/12	1/6	1/4	1/6	1/6	1/6	1/4
garcia	1/4	0	1/4	0	0	0	1/6
his	1/6	1/6	0	1/6	1/6	1/6	0
not	1/6	1/6	0	1/6	1/6	1/6	0
strong	1/6	1/6	0	1/6	1/6	1/6	0

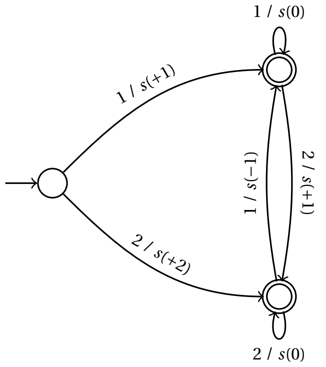
In the M step, we renormalize for each e (that is, for each row) to obtain probabilities $t(f \mid e)$:

			f				
e	asociados	fuertes	garcia	no	son	sus	y
NULL	5/19	2/19	3/19	2/19	2/19	2/19	3/19
and	1/3	0	1/3	0	0	0	1/3
are	1/5	1/5	0	1/5	1/5	1/5	0
associates	5/19	2/19	3/19	2/19	2/19	2/19	3/19
garcia	1/3	0	1/3	0	0	0	1/3
his	1/5	1/5	0	1/5	1/5	1/5	0
not	1/5	1/5	0	1/5	1/5	1/5	0
strong	1/5	1/5	0	1/5	1/5	1/5	0

After one iteration of EM, we can start to see what the model is learning. It correctly learns that *associates* most likely translates to *asociados*. It knows that *and* should translate to one of *garcia*, *y*, or *asociados*, but can't decide which; at the next iteration, it will start to learn that *and* does not translate to *asociados*, but it will never learn to distinguish *garcia* and *y*, because the model is too weak (it doesn't know anything about word order) and/or there isn't enough data (to observe sentences where *garcia* occurs without *y* or vice versa).

14.3 Hidden Markov Model

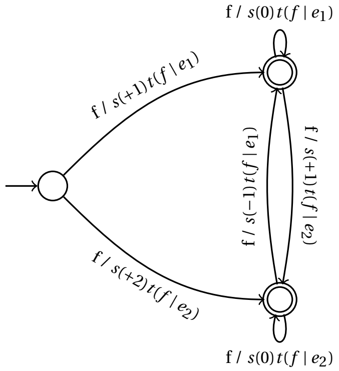
Model 1 doesn't care at all about word order. But we would like to capture the fact that if a Spanish word is translated from an English word, the next Spanish word is probably translated from the next English word. So we need some kind of dependence between the a_j. One easy and efficient way to do this is to make an alignment dependent on the previous alignment, i.e., a_j depends on a_{j-1} (Vogel, Ney, and Tillmann, 1996). Just as a bigram model could be represented as an FSA, so can this model. Here we show the FSA for $\ell = 2$.



The distribution $s(\Delta i)$ gives the probability that, if a Spanish word is translated from English word e_i, the next Spanish word is translated from English word $e_{i+\Delta i}$. We are treating the start of the sentence as position 0. So, if $\mathbf{a} = (1,2,4,3,5,6)$, its probability is $s(+1)s(+1)s(+2)s(-1)s(+2)s(+1)$. We expect the distribution to peak at +1 and decrease for larger $|\Delta i|$.

The model is deficient since it assigns a nonzero probability to alignments that fall off the edge of the sentence. We can fix this by renormalizing, but we haven't bothered to do so for simplicity's sake. Also note that there are no NULL alignments. These can be added in but were omitted in the original version (Vogel, Ney, and Tillmann, 1996) which we follow.

We compose this FSA with the same Spanish-word-generation model that we used before (again, assuming $\ell = 2$):


f / $s(0)\,t(f \mid e_1)$
f / $s(+1)\,t(f \mid e_1)$
f / $s(-1)\,t(f \mid e_1)$
f / $s(+1)\,t(f \mid e_2)$
f / $s(+2)\,t(f \mid e_2)$
f / $s(0)\,t(f \mid e_2)$

Then we compose with **f** (we don't show the result here because it would be too complicated). Then we have to estimate both the $t(f \mid e)$ and the $s(\Delta i)$. The algorithm goes like this:

1. Initialize $t(\cdot \mid e)$ and $s(\cdot)$ to uniform.
2. E-step:
   (a) Use the Forward-Backward algorithm to calculate a fractional count for each arc.
   (b) For each arc with weight $s(\Delta i)\,t(f \mid e)$ and fractional count p, let

$$c_s(\Delta i) \leftarrow c_s(\Delta i) + p$$
$$c_t(f,e) \leftarrow c_t(f,e) + p$$

3. M-step: let

$$s(\Delta i) \leftarrow \frac{c_s(\Delta i)}{\sum_{\Delta i} c_s(\Delta i)}$$
$$t(f \mid e) \leftarrow \frac{c_t(f,e)}{\sum_f c_t(f,e)}$$

4. Go to 2.

References

Brown, Peter F. et al. (1993). “The Mathematics of Statistical Machine Translation: Parameter Estimation”. In: *Computational Linguistics* 19, pp. 263–311.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann (1996). “HMM-Based Word Alignment in Statistical Translation”. In: *Proc. COLING*, pp. 836–841.