IBM Model 1 and IBM Model 2: Python Implementation

Zhengyuan Xu[1]

I. IBM MODEL 1

A. Description of IBM Model 1

IBM Models are translation models used for machine translation. They are an instance of a noisy-channel model. The models assign a conditional probability $p(f|e)$ to any Foreign/English pair of sentences. The parameters of these models are estimated from the translation examples. The goal is to model the conditional probability $p(f_1...f_m|e_1...e_l, m)$, where $f_1...f_m$ is the foreign sentence and $e_1...e_l$ is the English sentence. However, this can be hard to achieve without the help of **alignment**. The IBM Models instead define a conditional distribution $p(f_1...f_m, a_1...a_m|e_1...e_l, m)$ where $a_1...a_m$ is the alignment of foreign sentence with words $f_1...f_m$. IBM Model 1 uses only translation parameters $t(f|e)$, it is interpreted as the conditional probability of generating Foreign word f from English word e. So the final formula for IMB Model 1 is given by:

$$\begin{aligned} p(f, a|e, m) &= p(a|e, m) \times p(f|a, e, m) \\ &= \frac{1}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \end{aligned}$$

where f is the foreign sentence, a is the alignment of f to e, e is the English sentence and m is the length of the foreign sentence. In addition, we define e_0 to be the $NULL$ word. When the lexical probabilities $t(f|e)$ are estimated, we can then find the most probably alignment under the model:

$$\arg\max_{a_1...a_m} p(a_1...a_m|f_1...f_m, e_1...e_l, m)$$

and we simply define

$$a_i = \arg\max_{j \in \{0...l\}} t(f_i|e_j)$$

Limitations Since IBM Model 1 only uses the lexical parameters $t(f|e)$ in modeling, the model has very limited knowledge for other information such as length of foreign and English sentences, relative positions of i and j (namely, the connection of j'th foreign word and i'th English word is unknown). In practice, translation between sentences of different language could rely on such information.[1]

B. The Expectation-Maximization Algorithm

The EM algorithm is an efficient iterative approach to calculate the Maximum Likelihood (ML) estimate when some of the data are missing or hidden. [2] It is a way to find model parameters under many circumstances because data are often incomplete.

Strength The EM algorithm always improves a parameter's estimation through its process. It could be applied even when part of the data are missing. It is able to guess and estimate a set of parameters for your model under many situations.

Weakness The EM algorithm needs a few random starts to find the best model as the algorithm could end up stuck in a local maxima instead of the optimal global maxima. The EM algorithm can be very slow.[3]

C. Method

My implementation of IBM Model 1 includes two parts:

- The train portion. The training is done by running 5 iterations of EM Algorithm. The $t(f|e)$ parameters are initialized by:

  $$t(f|e) = \frac{1}{n(e)}$$

  where $n(e)$ is defined as the number of different words that occur in any translation of a sentence containing e. All counts are

initialized to zero at the beginning of each iteration. The algorithm runs through the training corpora in parallel and update the counts $c(e_j^k, f_i^k)$ and $c(e_j^k)$ by an increment of $delta$. In IBM Model 1, $delta$ is defined as:

$$\delta(k,i,j) = \frac{t(f_i^k|e_j^k)}{\sum_{j=0}^{l_k} t(f_i^k|e_j^k)}$$

and at the end of each iteration, $t(f|e)$ is updated by:

$$t(f|e) = \frac{c(e,f)}{c(e)}$$

The parameters $t(f|e)$ are eventually stored to a file ending in .ibm1.

- The test portion. The testing part reads from files the parameters trained by the training part. It then read the testing corpora in parallel and assign alignment variable to each sentences pair by the equation talked in Part A.

D. Results

The result of my implementation matches the expected output: Total tests: 5920. Precision: 0.413. Recall: 0.427. F1-Score: 0.420

E. Discussions

The F1 scores through the training process is given by Table 1.

Iteration	F1-Score
1	0.214
2	0.380
3	0.408
4	0.416
5	0.420

TABLE I

F1-SCORES THROUGH ITERATIONS IN IBM MODEL 1

As can be seen in Table 1, the F1-Score has been growing with number of iterations, although in a decreasing rate. From iteration 1 to iteration 2 the F1-Score increased by 0.166, whereas from iteration 4 to iteration 5 it only increased by 0.004. The EM algorithm with the initialization of $t(f|e)$ here guarantees the implementation to find the global maximum, and the estimation of model parameters is always improved, according to the F1-Scores reported.

II. IBM MODEL 2

A. Description of IBM Model 2

The IBM Model 2 is very similar to IBM Model 1 described in the above section. IBM Model 2 is different from IBM Model 1 as it has a set of new parameters, the **alignment** or **distortion** parameters:

$q(i|j,l,m) =$ Probability that j'th Foreign word is connected to i'th English word, given sentence lengths of e and f are l and m respectively.

and the conditional probabilities $p(f,a|e,m)$ is given by:

$$\begin{aligned} p(f,a|e,m) &= p(a|e,m) \times p(f|a,e,m) \\ &= \prod_{j=1}^{m} q(a_j|j,l,m) \times t(f_j|e_{a_j}) \end{aligned}$$

and the alignment for any word in the sentence pair $e_1...e_l$ and $f_1...f_m$ is defined by:

$$a_j = \arg\max_{a \in \{0...l\}} q(a|j,l,m) \times t(f_j|e_a)$$

for $j = 1...m$. The delta in EM Algorithm changes to:

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^k|e_j^k)}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^k|e_j^k)}$$

IBM Model 2 is outperforms IBM Model 1 because it takes into consideration the alignment parameter, **q**. With the help of q parameters, IBM Model 2 can model the translation of a source sentence word in position i to a target language sentence word in position j using an alignment probability distribution. Then each source word has two associated terms: first, a choice of alignment variable, specifying which target word the word is aligned to; and second, a choice of the source word itself, based on the target language word that was chosen by the $t(f|e)$ parameter.

Limitations IBM Model 2 still uses only word-to-word translations between languages where as in reality many translations could only be achieved if group of words(phrase) are translated together.

Also it some times maps an English(target language) word too many times to other Spanish words(source language). One example will be discussed in the Discussion section below.

B. Method

My implementation also has two parts for IBM Model 2:

- My implementation trains a IBM Model 2 with the $t(f|e)$ parameters initialized by reading the t parameters trained from IBM Model 1 by running EM Algorithm for 5 iterations. The alignment parameters are initialized by:

  $$q(j|i,l,m) = \frac{1}{l+1}$$

  Then the t and q parameters are updated by running 5 iterations of EM Algorithm. Final parameters are stored in files to be read later when testing. Other training details are discussed in the above section.
- Testing of IBM Model 2 is done by reading trained q and t parameters, and process parallel corpora, finding the alignments that maximizes $q \times t$.

C. Results

The result matches the expected F1-Score. The final results of the dev set are: total length: 5920, Precision: 0.442, Recall: 0.456, F1-Score: 0.449

D. Discussions

In this section two (almost) correctly aligned sentence pairs and two incorrectly aligned sentence pairs would be illustrated, in order to elaborate the performance of my implementation of IBM Model 2.

For illustration of these examples, please refer to the Appendix.

Iteration	F1-Score
1	0.439
2	0.442
3	0.446
4	0.446
5	0.449

TABLE II

F1-SCORES THROUGH ITERATIONS IN IBM MODEL 2

Table II shows the F1-Scores in different iterations. It is a clear improvement compared to the results of IBM Model 1. The first iteration gives an F1-Score of 0.439, which is 0.019 higher than the final F1-Score in IBM Model 1, which is a big improvement. The reason behind this improvement is that IBM Model 2 takes the final t parameters as the initial t parameters, and introduced the alignment q parameters, which accounts the length of English and Spanish sentences and positions of words when assigning alignments to word pairs. Then from iteration 2 to 5, the magnitude of improvement gradually decreases. Iteration 3 and 4 have same F1-Scores, but iteration 5 witnesses another improvement. This again proves the EM Algorithms always generate better parameters for the model.

E. Critical Thinking

Three possible ways to further improve the IBM Models:

- Add a restriction to how many source words could be aligned to a target word. In the Spanish-English example talked in the implementation, sometimes an English word is aligned to too many Spanish words, regardless of the positions. In real-world translation, we know that a word should not be aligned other words in foreign language too many times. A restriction parameter should be added to the model to limit the number of source words that are aligned to a target word.
- Pre-process the sentences. Pro-processing the sentences could better structure the sentences grammatically. For example, in two different languages, words with different part-of-speech could be in different positions. If in the source language the adverb comes before the verb and the target language might have the verb in front of the adverb. We could pre-process the sentences so they are in same structure.
- Eliminating impossible word pairs. Parameters could be added to indicate impossible pairs of words in two languages. For example, a noun should never be translated to a preposition. But in the IBM Models 1 and 2 the parameters result in positive probabilities of

such impossible translations between words. We could define part-of-speech tags to eliminate such problems.

III. GROWING ALIGNMENTS

A. Method Overview

To further improve the model performance, we need to use phrase translation techniques. The method is to train IBM Model 2 to calculate $p(f|e)$ and $p(e|f)$ as a starting point. Take the alignments given by two sets of parameters help us to evaluate the alignments produced by IBM Model 2 in two directions: from English to Spanish and from Spanish to English. I first train the IBM Model 2 for $p(f|e)$ and use the parameters in the model to produce the most likely alignment for each (e, f) pair. Then I train IBM Model 2 for $p(e|f)$ and produce most likely alignments for each (e, f) pair. Now with two sets of alignments, I take the intersection of the two sets as a starting point.

The heuristic method used in the implementation starts with the intersection of the two sets, and grow the alignments accordingly. Any alignment point in the union of $p(f|e)$ and $p(e|f)$ could be a candidate when growing. One alignment point is added each time, and that one alignment point should be only chosen from those pairs who are currently without alignment assigned. To grow the alignments, word pairs without assigned alignment that are close to those who have been assigned would be explored first. After the initial intersection has stopped growing, we now turn to other alignment points who are not neighbors of these points in the alignments.

B. Results and Discussions

The implementation is not finished. Now I only get the intersections of the alignments. I did not implement the growing part of the algorithm. The F1-Score is only 0.024.

C. Critical Thinking

New alignment points could be assigned to a word pair if that word pair is not in the union but are close to each other. We could keep a set of parameters to see how likely neighboring alignment points could be assigned even if some of the point is not in the union of $p(f|e)$ and $p(e|f)$.

REFERENCES

[1] Collins, Michael, Notes on Statistical Machine Translation: IBM Models 1 and 2

[2] Borman, Sean, The Expectation Maximization Algorithm A short tutorial, June, 28, 2006.

[3] Dempster, A., Laird, N., and Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, no. 1, pp. 1ñ38.

APPENDIX

A. Correctly Aligned Sentences

Fig. 1. Alignments generated by IBM Model 2. There are two misaligned word pairs in the figure. "que" is misaligned to "hope" and "puedan" is misaligned to "use". Other than these, other word pairs are correctly aligned to each other. The accuracy is high enough to put it under the correctly aligned category. The gold alignment is given in Fig.2. The reason why this pair of sentences are aligned most correctly is that the sentences are in relatively simple structures. Also there are no complex phrases in the sentences. The model lacks the ability to translate phrases, which cannot be demonstrated in these sentences.

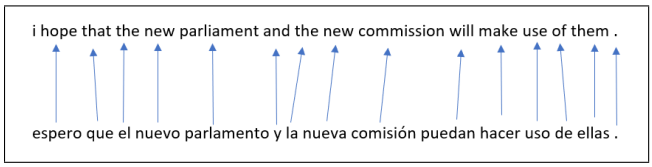

Fig. 2. The gold alignments for the sentence pair

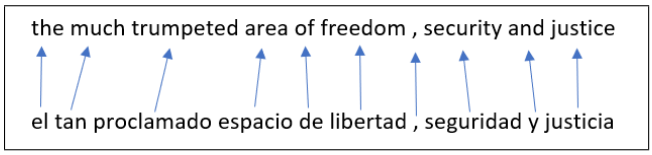

Fig. 3. This example was extracted from part of a sentence pair. In this example, every word pair has correct alignment. The graph suggests that the alignments of these word pairs are one-to-one. There are not complex grammatical structure or phrases in these sentences. The mapping of words is straight forward.

The sentences given in Fig. 1,2 and 3 have the following properties:

- Relatively short in length.
- No complicated sentence structure.
- No complicated phrases.

IBM Model 2 does not have the ability to translate phrases properly. It only uses lexical parameters and distortion parameters to calculate the alignments. Therefore, when sentence pairs do not have complicated phrases or sentence structure, the performance of the model tends to be high.

B. Misaligned Sentences

In Figure 4, a misaligned example is shown. Different from the sentences in the correctly aligned section, this sentence has a phrase ”in time” and its counterpart in Spanish ”a la larga”. The Model clearly failed to catch the meaning of this phrase. It did not assign alignment to the phrase ”in time” in the English sentence. This reveals the weakness of the model, which is not able to process phrases correctly.

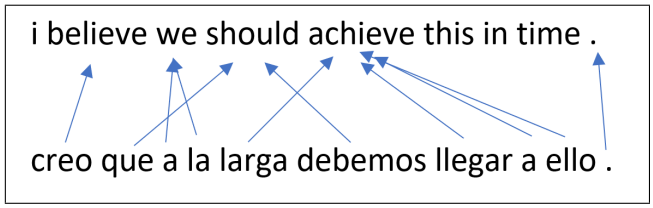

Fig. 4. Misaligned sentence pair

1	2		3	4	5	6	7	8	9
i	believe		we	should	achieve	this	in	time	.
1	2	3	4	5	6	7		8	9	10
creo	que	a	la	larga	debemos	llegar		a	ello	.
2		7,8	7,8	7,8	4	5			6	9

Fig. 5. Gold alignment for the Misaligned sentence pair in Fig 4. Note that to better illustrate the alignments arrows are replaced by indexes numbered at the bottom of the Spanish sentence. A 7,8 under ”a” in the Spanish sentence means the word ”a” is aligned the 7th and 8th words in English sentence.

Another misaligned example is given in Fig 6 and 7. Fig 6 is the output of the Model and Fig 7 is the gold alignments.

These sentences have more complicated structure when translating. The word in Spanish ”hablando” is translated to ”are we talking”. Our IBM Model could not discover this relation between ”hablando” and ”are we talking” and thus failed the task of assigning correct alignments. IBM Model 2 will assign every foreign word

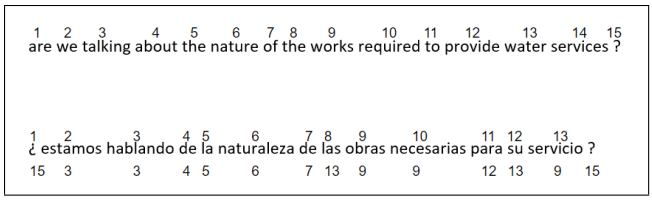

Fig. 6. Misaligned sentence pair

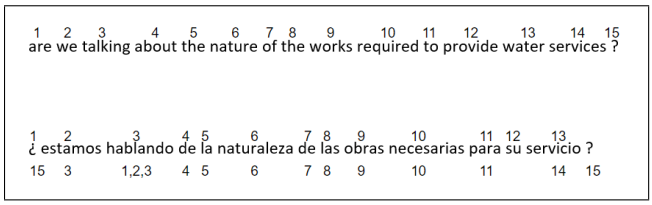

Fig. 7. Gold alignment for the Misaligned sentence pair in Fig 6. Note that to better illustrate the alignments arrows are replaced by indexes numbered at the bottom of the Spanish sentence. A 1,2,3 under ”hablando” in the Spanish sentence means the word ”hablando” is aligned the 1st, 2nd and 3rd words in English sentence.

once in the alignments. Every Spanish word here is guaranteed to receive a corresponding English word, but only one. However in reality, this is not always the case, as illustrated in Fig 7.