Programming Assignment 4 : Machine Translation

Gitika Meher
Department of Electrical and Computer Engg.
University of California San Diego
9500 Gilman Dr, La Jolla, CA 92093
gkarumur@eng.ucsd.edu

1 Machine Translation models

IBM models 1 and 2 are implemented as a part of this assignment. Both the models assume a noisy channel approach. Each of them has 2 components: 1. A language model that assigns a probability p(e) for any sentence e. 2. A translation model that assigns a conditional probability p(fle) to any Foreign/English pair of sentences. Following the above mentioned models given by the noisy channel approach, the output of the translation model on a new Foreign sentence f in English is:

$$e^* = \arg\max_{e \in E} \quad p(e) \times p(f|e)$$

where E is the set of all sentences in English. Thus the score for a potential translation e is the product of two scores: first, the language-model score p(e), which gives a prior distribution over which sentences are likely in English; second, the translation-model score p(fle), which indicates how likely we are to see the French sentence f as a translation of e. Using the Bayes Rule:

$$p(e|f) = \frac{p(e)p(f|e)}{\sum_e p(e)p(f|e)}$$

$$\begin{aligned} \arg\max_{e \in E} p(e|f) &= \arg\max_{e \in E} \frac{p(e)p(f|e)}{\sum_e p(e)p(f|e)} \\ &= \arg\max_{e \in E} p(e)p(f|e) \end{aligned}$$

The models make direct use of the idea of alignments, and as a consequence allow us to recover alignments between French and English words in the training data. The different approaches used by the models for the same is mentioned below.
Alignments: Alignment variables specify an alignment for each Foreign word to some word in the English sentence. The above mentioned conditional probability is marginalized to realize the dependence on the alignment variables.

$$p(f_1 \dots f_m|e_1 \dots e_l) = \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} \sum_{a_3=0}^{l} \dots \sum_{a_m=0}^{l} p(f_1 \dots f_m, a_1 \dots a_m|e_1 \dots e_l)$$

2 IBM Model 1

2.1 Description of the IBM model 1

IBM model 1 is derived from the above explanation of machine translation models. It uses an estimation algorithm to statistically estimate the alignment variables. An IBM-M1 model consists of a finite set E of English words, a set F of Foreign words, and integers M and L specifying the maximum length of French and English sentences respectively. t(fle) characterizes the parameters of the model where for any f ∈ F, e ∈ E ∪ NULL, the parameter t(fle) can be interpreted as the conditional probability of generating French word f from English word e.
Given these definitions, we define the conditional distribution over French sentences given the french sentences and alignments as: The conditional probability of French word f being aligned to English

$$p(f_1 \dots f_m, a_1 \dots a_m | e_1 \dots e_l, m) = \prod_{i=1}^{m} \frac{1}{(l+1)} \times t(f_i|e_{a_i}) = \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(f_i|e_{a_i})$$

word e , given the French length m and the English length l is uniformly modeled over all l+1 english words.
IBM Model 1 is weak in terms of conducting reordering or adding and dropping words. In most cases, words that follow each other in one language would have a different order after translation, but IBM Model 1 treats all kinds of reordering as equally possible. Also, this model doesn't consider the fact that a single foreign word can be aligned to multiple english words. Although, it does take into consideration the inverse.

2.2 Description of the EM algorithm

The parameters of IBM Model 1 can be estimated using the EM algorithm. The algorithm is iterative. t-parameters are initialized uniformly based on the counts for the english words. In every iteration, the Maximum likelihood counts are used for calculating the t-parameters as the ML estimates maximize the log-likelihood function and the t-parameters are re-estimated till we observe convergence. The

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

delta updates are calculated to update the counts accordingly. These updates are in-turn used to estimate the t-parameters in the next iteration.

$$\delta(k,i,j) = \frac{\frac{1}{(l^{(k)}+1)} t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} \frac{1}{(l^{(k)}+1)} t(f_i^{(k)}|e_j^{(k)})} = \frac{t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} t(f_i^{(k)}|e_j^{(k)})}$$

We have a guarantee of convergence to the global optimum of the log-likelihood function if it is a convex function. Because of this, the EM algorithm will converge to the same value, regardless of initialization if the log-likelihood function is convex. EM assumes convexity and proceeds as far as the implementation is concerned. The EM algorithm will converge to a local maximum of the log-likelihood function and it is dependent on the initialization of the t-parameters in this case. Since the problem of estimating a maximum is NP-hard, the EM algorithm provides a convenient and a simple solution for estimating the parameters but cannot guarantee global optimum.

2.3 Overall Method Overview

The psuedo code for implementing the IBM model-1 EM algorithm is described below. It depicts how the model's parameters can be estimated using the EM algorithm. The outer loop controls the number of iterations of the algorithm. Every iteration starts from setting the counts pertaining to the words from english and foreign languages appearing in the same sentence to 0. They are re-calibrated based on the parameters as shown in the code.

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$. An integer S specifying the number of iterations of training.
Initialization: Initialize $t(f|e)$ parameters (e.g., to random values).

Algorithm:

- For $s = 1 \ldots S$
  - Set all counts $c(\ldots) = 0$
  - For $k = 1 \ldots n$
    - For $i = 1 \ldots m_k$
      - For $j = 0 \ldots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\
c(j|i,l_k,m_k) &\leftarrow c(j|i,l_k,m_k) + \delta(k,i,j) \\
c(i,l_k,m_k) &\leftarrow c(i,l_k,m_k) + \delta(k,i,j)
\end{aligned}
$$

where

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

  - Set

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

Output: parameters $t(f|e)$

The t-parameters gathered help us to retrieve the underlying lexical probabilities between both the sentences. Because the q parameters (they are explained in the next section) are uniformly modeled in the case of the IBM model 1, the estimated alignments only depend on the maximizing t-parameter.

$$a_i = \arg \max_{j \in \{0 \ldots l\}} (q(j|i,l,m) \times t(f_i|e_j))$$

2.4 Results and Discussion

The F1 score and the other scoring evaluations for the IBM model 1 are presented below after running the EM algorithm for 5 iterations.

Type	Total	Precision	Recall	F1-Score
total	5920	0.413	0.427	0.420

The table below shows the F1 score obtained for different iteration of the EM algorithm.

Iteration	0	1	2	3	4	5	7
F1 score	0.04	0.214	0.38	0.409	0.416	0.42	0.427

Discussions : We observe that the F1 score gradually increases by the iteration. Initial t-parameters which are uniformly set achieve a very low F1 score but it increases by the iteration when the parameters are calculated based on the ML estimates.

3 IBM Model 2

3.1 Description of the IBM model 2

IBM model 2 is an improvement over the IBM model 1. Similar to the IBM model 1, it also is derived from the machine translation models described in section 1 of this report. IBM model 2 is characterized using two parameters: **1.** Translation parameters similar to the IBM model 1 and **2.** Distortion parameters(q-parameters) which model the probability of seeing a foreign word and an english word in the same sentence given the lengths of the foreign sentence and the english sentence. Counts for q-parameter estimation are calculated in every iteration along with the ones in the IBM model 1 to estimate the t-parameters. These are obtained to calibrate the ML estimates of the t and the q parameters. As mentioned in the IBM model 1, EM algorithm is used in this case too for parameter estimation. The t-parameters obtained in the end of 5 iterations are used as the initial values. q-parameters are all initialized uniformly to the inverse of (1 + length of the english sentence). The overall method is described below.

Formal Definition: An IBM-M2 model consists of a finite set E of English words, a set F of French words, and integers M and L specifying the maximum length of French and English sentences respectively. The parameters of the model are: **1.**t(fle) for any f ∈ F, e ∈ E ∪ NULL. The parameter t(fle) can be interpreted as the conditional probability of generating French word f from English word e.**2.** q(jli, l, m) for any l ∈ 1 . . . L, m ∈ 1 . . . M, i ∈ 1 . . . m, j ∈ 0 . . . l. The parameter q(jli, l, m) can be interpreted as the probability of alignment variable a_i taking the value j, conditioned on the lengths l and m of the English and French sentences.

$$p(f_1 \dots f_m, a_1 \dots a_m | e_1 \dots e_l, m) = \prod_{i=1}^{m} q(a_i|i,l,m)t(f_i|e_{a_i})$$

This model is an improvement over the IBM model 1 as it also considers the distortion parameters in addition with the transition parameters. Distortion parameters also take into consideration the lengths of the sentences. The IBM Model 2 has an additional model for alignment that is not present in Model 1. As in, There's a step for alignment after lexical translation which can be represented as the follows:

$$p(e, a \mid f) = \in \prod_{j=1}^{l_e} t(e_j \vee f_{a|j}) a(a(j) \vee j, l_e, l_f)$$

In this equation, the alignment function a maps each output word j to a foreign input position a(j). Also, the q-parameters allow, for example, to capture the tendency for words close to the beginning of the French sentence to be translations of words close to the beginning of the English sentence. It is an important aspect of data not captured by IBM Model 1.

3.2 Overall Method Overview

As mentioned for the IBM Model 1, IBM model 2 also estimates it's parameters(transition and distortion) based on Maximizing the log-probability of the data. Every iteration gathers different counts for the estimation of the same. The algorithm is iterative. We begin with some initial value for the t and q parameters and compile the following counts: c(e), c(e, f), c(jli, l, m) and c(i, l, m) based on the data together with our current estimates of the parameters. The first two counts were introduced in IBM model 1. The last two counts indicate the number of times we see an English sentence of length l, and a French sentence of length m, where word i in French is aligned to word j in English. Finally, c(i, l, m) is the number of times we see an English sentence of length l together with a French sentence of length m. We then re-estimate the parameters using these counts, and iterate. The pseudo code for the same is presented below.

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$
    - For $j = 0 \ldots l_k$

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l_k, m_k) &\leftarrow c(j|i, l_k, m_k) + \delta(k, i, j) \\
c(i, l_k, m_k) &\leftarrow c(i, l_k, m_k) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output:

$$t_{ML}(f|e) = \frac{c(e, f)}{c(e)} \qquad q_{ML}(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$$

3.3 Results

The F1 score and the other scoring evaluations for the IBM model 2 are presented below after running the EM algorithm for 5 iterations using the t-parameters obtained from IBM model 1 and q-parameters being initialized uniformly.

Type	Total	Precision	Recall	F1-Score
total	5920	0.443	0.457	0.450

The table below shows the F1 score obtained for different iteration of the EM algorithm.

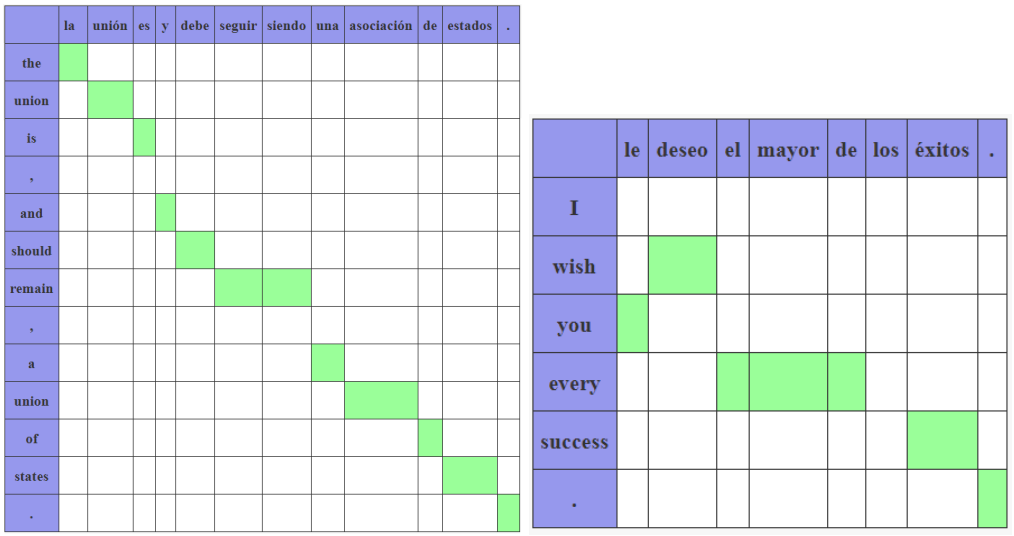
Iteration	0	1	2	3	4	5
F1 score	0.42	0.432	0.439	0.442	0.445	0.45

3.4 Discussions

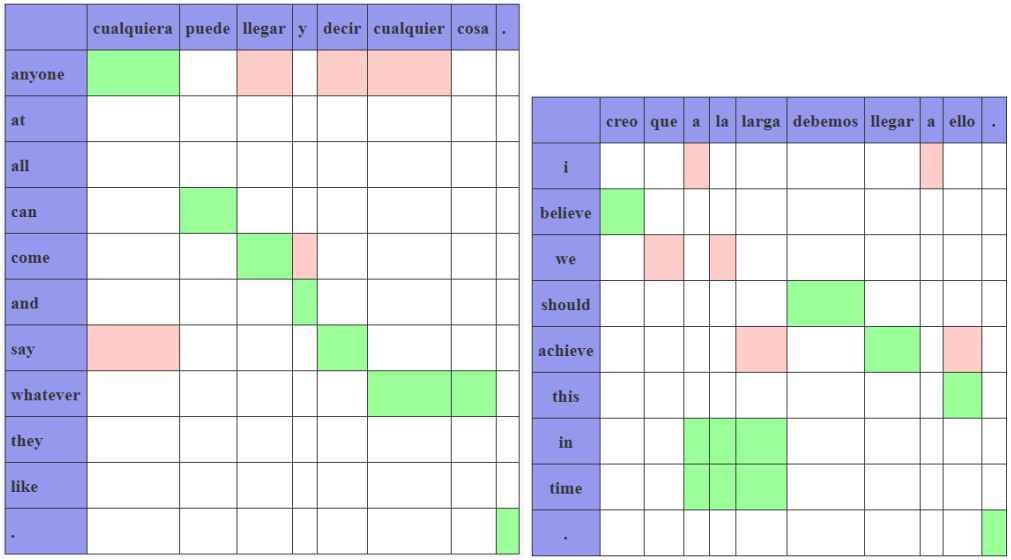
We observe that the F1 score increases by the iteration. The score in the 0th iteration is that of the IBM model 1 as the t-parameters and q-parameters are initialized from the IBM model 1.

Correctly aligned examples:
Below shown are two correctly aligned examples. The green boxes show the alignment of the respective english and spanish words. Both the examples have examples in which words like 'the', 'is', 'of','and','you', 'I' appear which are frequent and the model knows the translation to these words. For the other words, since the IBM model 2 has both lexical translation and absolute alignment steps, words like 'union', 'states' are correctly aligned in the first sentence and words like 'wish' and 'success' are correctly aligned in the second sentence.

Misaligned aligned examples:
Below shown are two misaligned examples. The green boxes show the correct alignments of the respective english and spanish words. The red boxes indicate additional alignments indicated by our model which are not present in the key. Alignments for words like 'and', 'this', 'in', 'can' are correctly estimated because they are seen frequently by our model. But the key doesn't show



alignments for the words like 'we', 'at','all' in this context but our model guesses alignments in this case because they are frequent words too and the model assumes that they correspond to something in the sentence. The other misaligned words like 'say', 'anyone', 'we', 'achieve' also are also words known by the model. The estimated alignments are in accordance with that of the training set and these sentences present ambiguous alignments than seen by the model before.



3.5 Critical Thinking

- IBM model 2 captures the alignments better than IBM model 1. However, the performance can be further improved by considering a phrase based model because the fact that each word is dependent on the previously aligned word and on the word classes of the surrounding words can be exploited.
- Another improvement I can think of is by using MAP estimates of the count values by modeling a prior instead of using the MLE estimates. MAP is a lot like Maximum likelihood distribution but employs an augmented optimization objective which incorporates a prior distribution which acts as a regularization over the ML distribution.

4 Growing Alignments

I followed the algorithm in the assignment sheet to first compute p(f|e) and then compute p(e|f). I then calculated the intersection and union of all the alignments. I additionally computed F1 scores for two more procedures. The table in the results section compares all these methods. The description for both the procedures in mentioned below.
Procedure-1: Add alignments of only the english words not aligned to any spanish words as a part of the intersection from the union.
Procedure-2: Add alignments of both spanish and english words not aligned to any other words in the intersection from the union.
The brief overall picture for following either of the procedures in depicted below.

```
while new points added do
    for all English word e ∈ [1...e_n], foreign word f ∈ [1...f_n], (e, f) ∈ A do
        for all neighboring alignment points (e_new, f_new) do
            if (e_new unaligned OR f_new unaligned) AND (e_new, f_new) ∈ union(e2f,f2e) then
                add (e_new, f_new) to A
            end if
        end for
    end for
end while
```

4.1 Results

Metric	Precision	Recall	F1-Score
IBM-1	0.413	0.427	0.42
IBM-2	0.443	0.457	0.45
Intersection	0.794	0.367	0.502
Procedure-1	0.476	0.465	0.470
Procedure-2	0.366	0.519	0.429
Union	0.350	0.541	0.425

4.2 Discussions

The intersection usually contains good alignment points and that's why we observe high precision. The union on the other hand, usually contains most of the desired alignment points and achieves a high recall, but also faulty points with very low precision.We see that intersection has the highest precision and the lowest recall from the table and also union has the highest recall and lowest precision. All the other procedures lie in the middle of these two. By adding any additional alignment points to the intersection, we observe that the precision falls and the recall rises. Only by adding english words from the union not containing alignment in the intersection, precision falls from 0.794 to 0.476 and this is the first step of any addition that can be done. Other methods from published papers suggest including neighboring words to grow alignments. A possible way to mitigate the same is presented in the next section but is not implemented.

4.3 Critical Thinking

Use of Alignment Error Rate (AER) which is a common metric for evaluation word alignments can solve the above problem. We could only add the words which would reduce AER to the set of Intersection points. AER is defined as follows:

$$\text{AER}(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

where S stands for Intersection points, P stands for the union points and A stands for the alignment procedure to be improved. AER is 0 for the intersection procedure.