In this programming assignment, we have to develop the IBM1 and IBM2 machine translation models. After implementing IBM1 and IBM2 our goal is to find alignments of English/Spanish words using these models developed. The aim of both the model is to estimate the conditional probability of a Spanish sentence $s_1, s_2, \dots s_m$ and aim to find alignment $a_1, a_2, \dots a_m$ for the Spanish word in English sentence $e_1, e_2, \dots e_l$. The first IBM model defines conditional probability using only one parameter t:

$$P(s_1 \dots s_m, a_1 \dots a_m | e_1 \dots e_l, m) = \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(s_i | e_{ai})$$

In the IBM2 its same conditional probability is defined using 2 parameters t and q:

$$P(s_1 \dots s_m, a_1 \dots a_m | e_1 \dots e_l, m) = \prod_{i=1}^{m} q(a_i | i, l, m) t(s_i | e_{ai})$$

4.1 IBM Model 1

a. Description of IBM Model 1 - What are IBM Models used for? What are the limitations?

IBM model consists of a finite set of E (English words) and a set of F foreign words, in our case Spanish words, and integer M and L, which denotes the maximum length of English and Spanish words respectively. IBM models are instances of noisy channels and contain two components.

1. A language model which calculates the probability of English words for a sentence $e = e_1 \dots e_l$ in English.

2. A translation model which calculates conditional probability p(s|e) to any Spanish/English pair of sentences.

IBM models are used to find alignments for a sequence in machine translation. In our case, we need to find alignment in English words for Spanish words in order to properly translate Spanish to English i.e., machine translation. The main objective is to find conditional probability $P(s_1 \dots s_m, a_1 \dots a_m | e_1 \dots e_l, m)$ i.e., we need to find the probability of Spanish words and alignments given English words and length of Spanish words in a sentence. In our case, we also add {*} to denote a null word in other words we define * as a special word in English sentence e_0, which represents that if we are unable to find the right alignment in the English sentence for the word s_j we mark that it is generated from *(NULL) word.

We need to find alignment for the words in an English sentence corresponding to Spanish words in a sentence. Hence for any pair of words in an English sentence $e_1 \dots e_l$ and foreign words (Spanish sentence) in a sentence $f_1 \dots s_l$ and length l for English sentence and m for a foreign sentence with corresponding alignments $a_1 \dots a_m$ in English sentence, we define the conditional probability as:

$$P(s_1 \dots s_m, a_1 \dots a_m | e_1 \dots e_l, m) = \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(s_i | e_{ai})$$

IBM models are used to model conditional probability distribution for machine translation p(s|e), where s represents a Spanish word in a sentence and e represents English words in a sentence. It uses the concept of alignments based on conditional probability distribution and tends to find the correct alignment for a given Spanish word in an English sentence. Since IGM is part of the generative model we tend to generate the conditional probability distribution using the EM algorithm. We will generate the distribution based on these and will try to find the proper alignments.

In the IBM model, we tend to map/align each Spanish word to the corresponding English translation in a sentence. Since for each Spanish word we align with one English word it is many to one mapping. So, it might be possible that this many-to-one mapping holds true for all cases. It might be possible that many English words tend to form one word in Spanish. IBM algorithm fails to accommodate such type of conditions. Because of many one mapping of an IBM model, it might be possible that some of the English words are not mapped to any of the Spanish words. Since tries to calculate the probability distribution of P(s|e), it fails to consider the one-to-many mappings as well as relative alignments in English words. IBM model 1 also fails to consider varying alignment probability P(j|i, l,m), (probability of alignment variable a_i taking the value j given length l and m) thus id does not consider the length of English and Spanish sentences. So, it fails to compute $q(a_i|i, l, m)$ which means varying positional alignment instead of assuming uniform distribution. We also need to consider the corresponding alignment probability in the sentence i.e., $q(a_i|i, l, m)$. This issue is resolved by IBM model 2.

b. Description of EM Algorithm. What are the strengths and weaknesses?

EM is referred to as the Expectation-Maximization algorithm. EM algorithm finds maximum likelihood estimation by computing posterior probabilities and tends to maximize log-likelihood. In the EM algorithm there are 2 steps, first is to compute E-step which estimates the expected value for each latent variable. The second is the M step, to maximize the log-likelihood parameters. Log-likelihood function is given by the following:

Log-Likelihood= $\sum_{k=1}^{n} \log \left(\sum_{a \epsilon A(l_k, m_k)} p(f^{(k)}|e^{(k)}, m_k; t, q)\right)$

Since the log is a monotonically increasing function hence when we maximize the log function it also maximizes f(t|q). In the EM algorithm, we tend to iteratively maximize the log-likelihood.

Pros of EM Algorithm: EM follows an iterative approach since, in IBM model 1, we tend to maximize log-likelihood estimation. So using EM estimation we can converge at the global optimal solution since our log-likelihood contains p(f|a,e), thus when we try to maximize the log function and since the log is a monotonically increasing function we tend to maximize p(f | a, e), thus maximizing the conditional distribution for finding alignment for given Spanish word in English. The EM algorithm is used when we need to generate conditional distribution since data is missing, so the EM algorithm tends to estimate the parameters for the model in other words it helps to generate conditional distribution by computing posterior probabilities.

Cons of EM Algorithm: In the IBM model 2 we are computing q(a|i, l,m), so in log-likelihood, we tend to maximize log-likelihood, but in IBM 2 we are taking alignment and length into account which leads to non-convex function. Since we are initializing weights randomly it might be possible that we might end up at the local optimum. Another con is that convergence might take time as well as for convergence we need to compute forward and backward posterior probabilities.

c. Method Overview. Provide a high-level description of your implementation

I have implemented IBM model as described in Collins notes, following is the algorithm:

Input: A training corpus ($s^{(k)}$, $e^{(k)}$), for k=1…n, where $s^{(k)}$=$s_1^{(k)}$, … $s_m^{(k)}$, and $e^{(k)}$= $e_1^{(k)}$… $e_l^{(k)}$. Initially if the value of t(s|e) is not present it has been declared to 1/n(e), as mentioned in the assignment.

Algorithm:

```
t=dictionary
For epochs=1,2…5
    Set c(s,e)=defaultdict()  // If it doesn't exist return 0
    Set c(e)=defaultdict ()  // If it doesn't exist return 0
    For k = 1… n: total length of the data
        Add * in start of English sentence, this represents NULL
        For i=1…$m_k$, for every word in Spanish for a given sentence
            For j=1…$l_k$, for every word in english for a given sentence
                $c(s_j^{(k)}, e_j^{(k)}) = c(s_j^{(k)}, e_j^{(k)}) + \delta(k,i,j)$; updating estimation
                $c(e_j^{(k)}) = c(e_j^{(k)}) + \delta(k,i,j)$; updating estimation
            where $\delta=\frac{t(s_i^k|e_j^k)}{\sum_{j=0}^{l_k} t(s_i^k|e_j^k)}$,   if t(s|e) is not present, initialize it with 1/n(e)
            Set $t(s|e) = \frac{c(e,f)}{c(e)}$
Output(t(s|e))
```

The above is the EM algorithm for partially observed data. All c(s, e) and c(e) are constructed using dictionary, where key is tuple of Spanish word and English word in c(s,e). Before iterating over any words in a sentence * is appended at the start of a sentence.

For output file

$$a_i = \arg max_{j\epsilon 0\ldots l}\, t(s_i|e_j)$$

I have stripped all the spaces at the start and end of sentence using rstrip () function. For finding the right alignment for each Spanish word we have taken the argmax $t(s_i^{(k)}|e)$, for a i^{th} Spanish word in sentence k, we find the maximum $t(s_i^{(k)}|e)$, in English words for k^{th} sentence.

d. Results. Report your F1 score. It should match the expected result. If not, partial credits will be given if you discuss some challenges and issues in your implementation.

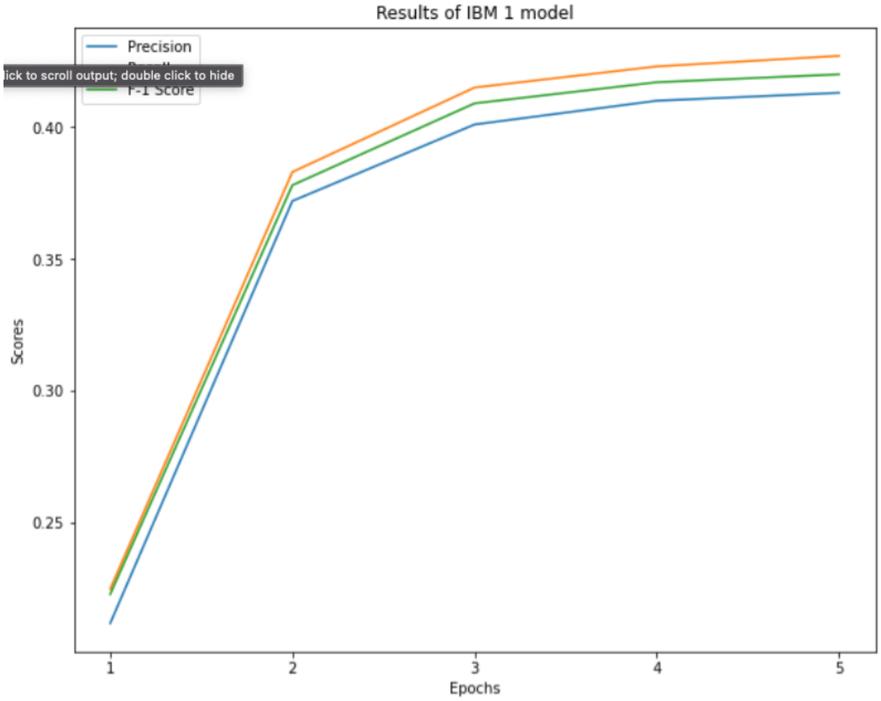
The above EM algorithm is performed for 5 epochs. After 5 epochs following are the results obtained.

	Precision	Recall	F1-Score
5-Epochs	0.413	0.427	0.420

From the above table we can see that EM algorithm of IBM 1 model has achieved F1-Score of 0.420 as mentioned in the assignment. The precision and recall obtained are 0.413 and 0.427 respectively after 5 epochs. Following section talks about the precision, recall and F-1 score is further details with epochs.

e. Discussion - Show how F1 score changes after each iteration, comment on the findings.

	Precision	Recall	F1-Score
1-Epochs	0.212	0.225	0.223
2-Epochs	0.372	0.383	0.378
3-Epochs	0.401	0.415	0.409
4-Epochs	0.410	0.423	0.417
5-Epochs	0.413	0.427	0.420



Thus, we can see that number of epochs increases F1-Score, precision and recall also increases. The F1-score is 0.420 after 5 epochs as mentioned in the document. The we can also see that from the graph that all the three scores precision. From epoch 1 to epoch 2 we see that there is huge increase in precision, recall and F-1 score: 0.372, 0.383 and 0.378 respectively. Recall and F1-Score grows very similar to each other. We can see from the graph that as for 4 epochs to epoch 5 there is very little increase in the scores, which shows that convergence happens pretty fast. The graph shows that convergence happens very quickly with EM algorithm in IBM 1 model.

4.2 IBM Model 2

a. Description of IBM Model 2. Why is it better than IBM Model 1? What are the limitations?

IBM model 2 is the advancement of IBM model 1. IBM model 2, while computing the conditional probability distribution it considers various parameters like alignment, Spanish word corresponding to the English word, length of the Spanish word, length of English word, distortion, and alignments. The main additional step in this algorithm is how we compute delta.

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

The alignment probability of words in an English sentence (length l) and Spanish words from a corresponding Spanish sentence (length m), then the conditional probability distribution obtained for all the alignment of these words in English and Spanish sentences is calculated using the following formula:

$$\mathrm{P}(\mathrm{s}_1 \ldots \mathrm{s}_\mathrm{m}, \mathrm{a}_1 \ldots \mathrm{a}_\mathrm{m} |\ \mathrm{e}_1 \ldots \mathrm{e}_\mathrm{l}, \mathrm{m}) = \prod_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{q}(\mathrm{a}_\mathrm{i}|\mathrm{i},\mathrm{l},\mathrm{m})\mathrm{t}(\mathrm{s}_\mathrm{i}|\mathrm{e}_{\mathrm{ai}})$$

$$a_i = \arg max_{j\epsilon 0\ldots l}\ (t(s_i|e_j) * q(j|i,l,m))$$

The implementation of IBM model 2 has one more parameter as compared to IBM 1 model. It includes the computation of q(j|i, l,m). This probability represents that for kth sentence the Spanish word at index j translation with i index in English sentence k, where the length of English sentence is l and Spanish sentence is m. IBM model 2 we also take distortion into consideration while generating conditional probability distribution.

Since in IBM-2 model is a non-convex function, so it might be possible that while applying the EM algorithm it might converge at local minima instead of converging at global minima. Since we are initializing the weights in a certain manner which can lead to convergence at local minima.

b. Method Overview. Provide a brief, high-level description of your implementation.

I have implemented the IBM model as described in Collins's notes, following is the algorithm:

Input: A training corpus ($s^{(k)}$, $e^{(k)}$), for k=1…n, where $s^{(k)}=s_1^{(k)}$, … $s_m^{(k)}$, and $e^{(k)}= e_1^{(k)}… e_l^{(k)}$. Initially if the value of t(s|e) is not present it has been declared to 1/n(e), as mentioned in the assignment.

Algorithm:

```
t=dictionary
For epochs=1,2…5
    Set c(s, e)=defaultdict()  // If it doesn't exist return 0
    Set c(e)=defaultdict ()  // If it doesn't exist return 0
    Set c(j, i, l, m)=defaultdict () // If it doesn't exist return 0
    Set c(i, l, m)=defaultdict () // If it doesn't exist return 0
    For k = 1… n: total length of the data
        Add * in start of English sentence, this represents NULL
        For i=1…$m_k$, for every word in Spanish for a given sentence
            For j=1…$l_k$, for every word in English for a given sentence
                $c(s_j^{(k)}, e_j^{(k)}) = c(s_j^{(k)}, e_j^{(k)}) + \delta(k, i, j)$; updating estimation
                $c(e_j^{(k)}) = c(e_j^{(k)}) + \delta(k,i,j)$; updating estimation
                $c(j|i, l_k, m_k) = c(j|i, l_k, m_k) + \delta(k, i, j)$; updating estimation
                $c(i, l_k, m_k) = c(i, l_k, m_k) + \delta(k, i, j)$; updating estimation
            where $\delta(k,i,j)=\frac{q(j|i,l_k,m_k)\ t(s_i^k|e_j^k)}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(s_i^k|e_j^k)}$,if t(s|e) is not present, initialize it with 1/n(e)
            if q(j|i,l,m) is not present initialize it with 1/(l+1)
            Set $t(s|e) = \frac{c(e,f)}{c(e)}$, $q(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$
Output(t(s|e) and q(j|i,l,m))
```

The above is the EM algorithm for partially observed data. All c(s,e), c(j,i,l,m), c(i,l,m) and c(e) are constructed using dictionary(default dict in python), where key is tuple of Spanish word and English word in c(s,e) and indexes in c(j,i,l,m). Before iterating over any words in a sentence * is appended at the start of a sentence. For output file

$$a_i = \arg max_{j\epsilon 0\ldots l}\,(t(s_i|e_j) * q(j|i,l,m))$$

I have stripped all the spaces at the start and end of sentences using the rstrip () function. For finding the right alignment for each Spanish word we have taken the argmax $t(s_i^{(k)}|e) * q(j|i,l,m)$, for an ith Spanish word in sentence k and in q (j, i, l, m) indexes, we find the maximum $t(s_i^{(k)}|e)*q(j|i,l,m)$, in English words for kth sentence.

c. Results. Report your F1 score. It should match the expected result. If not, partial credits will be given if you discuss some challenges and issues in your implementation.

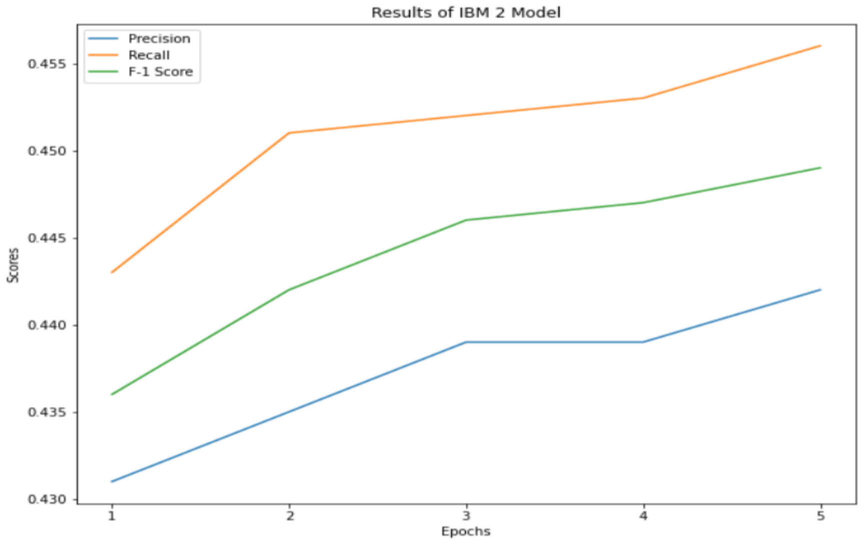
The above EM algorithm for IBM model 2 is performed for 5 epochs. After 5 epochs following are the results obtained.

	Precision	Recall	F1-Score
5-Epochs	0.442	0.456	0.449

From the above table, we can see that the EM algorithm of the IBM 2 model has achieved an F1-Score of 0.449 ~0.45 as mentioned in the assignment. The precision and recall obtained are 0.442 and 0.456 respectively after 5 epochs. The following section talks about the precision, recall, and F-1 score in further detail with epochs.

d. Discussion. Show how F1 score changes after each iteration, comment on the findings. Use the type of table/graph as in the slides or in the instructions, show a correctly aligned example + a misaligned example. Discuss the examples.

	Precision	Recall	F1-Score
1-Epochs	0.431	0.443	0.436
2-Epochs	0.435	0.451	0.442
3-Epochs	0.439	0.452	0.446
4-Epochs	0.439	0.453	0.447
5-Epochs	0.442	0.456	0.449



From the above graphs and table, we can see that there is an improvement in all the scores (Precision, Recall, F-1 Score). The IBM 2 model with EM algorithm has shown an increase in scores, thus considering the alignment we can see that it improves the scores. As the number of epochs increases, the F-1 score also increases. We can see from the graph that as compared to the IBM-1 model the scores obtained for epoch 1 are much higher in the IBM-2 model. In fact, after epoch 1 there is very little increase in the scores i.e., the F-1 score shows little increase from 0.436 to 0.449 from epoch 1 to 5. Thus, considering the additional step to include q (j|i, l, m) increases the model scores.

Misalignment:

Example of Misaligned Sentence using IBM-2 Model:

English: divergent national regulations should not be allowed to develop at all , or become established over and above the current scope .

Spanish: más allá del margen de maniobra existente no deberían surgir ni consolidarse en ningún caso regulaciones nacionales diferentes .

For the above example more than 80% of the alignment obtained was wrong.

Dev Alignment: [(1, 18), (2, 17), (3, 16), (4, 9), (5, 8), (8, 10), (9, 10), (10, 14), (10, 15), (10, 13), (11, 13), (11, 14), (11, 15), (13, 11), (14, 12), (15, 12), (18, 1), (18, 2), (19, 3), (20, 7), (21, 6), (21, 4), (21, 5), (22, 19)]

Alignment using IBM-2: [(1, 1), (1, 2), (21, 3), (1, 4), (15, 5), (7, 6), (1, 7), (5, 8), (1, 9), (1, 10), (1, 11), (1, 12), (14, 13), (7, 14), (7, 15), (1, 16), (2, 17), (1, 18), (22, 19)]

	más	allá	del	margen	de	maniobra	existente	no	deberían	surgir	ni	consolidarse	en	ningún	caso	regulaciones	nacionales	diferentes	.
divergent	■	■		■			■		■	■	■	■				■		■	
national																	■		
regulations																			
should																			
not								■											
be																			
allowed						■								■	■				
to																			
develop																			
at																			
all																			
,																			
or																			
become													■						
established					■														
over																			
and																			
above																			
the																			
current																			
scope			■																
.																			■

In the above example, we can see that there are a lot of misalignments between English and Spanish words. We can see that most of the Spanish words are mapped with the first word of the English sentence. In fact, many of the English words are not mapped with any of the Spanish words.

Example of perfectly aligned sentence using IBM-2 Model:
English: it was an animated , very convivial game .
Spanish: jugaban de una manera animada y muy cordial .

Dev Alignment: [(4, 5), (6, 7), (7, 8), (9, 9)]
Alignment using IBM-2: [(1, 1), (2, 2), (3, 3), (4, 4), (4, 5), (8, 6), (6, 7), (7, 8), (9, 9)]

	de	jugaban	una	manera	animada	y	muy	cordial	.
it	■								
was		■							
an			■						
animated				■	■				
Game						■			
very							■		
convivial								■	
,									
.									■

In the second example, we can see that all the English words are perfectly aligned with Spanish words. And all the words are mapped with at least one word of the other sentence, unlike the first example where most of the words were aligned with one word.

From the above two examples, we can see that the second one is perfectly aligned whereas the first one is not aligned. In the first example, we can see that the mapping is done in a haphazard manner.