Machine Translation

Statistical Natural Language Processing - PA4

Saideep Reddy Pakkeer
University of California San Diego
06.06.2019

INTRODUCTION

In this exercise we'll see about Machine Translation, and in particular statistical machine translation (SMT) systems. We will first look at IBM models 1 and 2, then look at a heuristics to get better alignment between the foreign and english words. For initial IBM 1 and 2 models, Spanish(f) will be the source language and the target language will be English(e).

IBM Model 1

1.1 Description of IBM Model 1:

IBM models are introduced to in the late 80s and are the basis of many SMT models now. In IBM model 1 we model the conditional distribution p(f/e) for any pair of sentences from two different languages $f = f_1 f_2 f_3 .. f_m$ *and* $e = e_1 e_2 ... e_l$ where m is the length of sentence from Spanish(in our case) and l is the length of sentence from English. In IBM model 1 we fix the length of spanish sentence "**m**" for modeling p(f/e). It is difficult to model $p(f_1 f_2 .. f_m / e_1 e_2 ... e_m, m)$ directly, hence, we introduce the idea of alignments where $a_1 . . . a_m$ are alignments of each Spanish word to English word. So each a_i takes values in $\{0, 1, \ldots l\}$. Here, 0 means it is aligned to special word NULL in the English sentence.

So, the way we design it is using the formula to model the distribution

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}, m) = \prod_{i=1}^{m} p(a_i|e, m) p(f_i|e_{a_i})$$

$$= \prod_{i=1}^{m} \frac{1}{l+1} t(f_i|e_{a_i}) = \left(\frac{1}{l+1}\right)^m \prod_{i=1}^{m} t(f_i|e_{a_i})$$

where we assume that each spanish word is equally probable $1/(l+1)$ to be aligned to any of the english words in the sentence. And, the parameters "t" can be thought of as the conditional probability of generating spanish word "f" from English word "e". We will see an algorithm called EM algorithm to estimate the parameters "t".

Limitations:

One major limitation for IBM model 1 is because of our simplistic assumption that all alignments (their orderings) are equally likely $\{1/(l+1)\}$, where as in reality most often the order is different when it is translated. Hence, we do not get accurate estimates for our conditional distribution. Also, it doesn't address how to take care of multiple words producing one word or no words at all when translated.

1.2 EM algorithm description:

EM (Expectation Maximization) algorithm is a very general technique (not specific to Machine Translation) to estimate parameters. It is kind of similar to k-means where we alternate between assigning points/data to a distribution and estimating the parameters and it is a popular algorithm in mixture models. Here, we start with some initialization of our "t" parameters and from the partially observed data we estimate it until the parameters converge.

Strengths and Limitations of EML:

One of the major advantages of EM algorithm is it's simplicity to implement and use. It is guaranteed to converge to a local optimum, hence the parameters always converge and it is the most important property for us. Given on partial data is available in practice, it's a very useful algorithm to work with and the estimates get better with each iteration.. But, one should not use it blindly without knowing it's limitations. Although, it is simple to implement and there are many variants to accelerate it's convergence, it may not converge as quickly as we want. Also, it depends on the choice of initialization as it only converges to a local optimum. Hence, initialization becomes very important while setting up the problem. The speed of convergence also depends on the initialization because of the local optimum.

1.3 Method Overview

Here is a brief description of the algorithm. We iterate through 5 EM iterations in our implementation. In each iteration we do the following:

We initialize the counts $c_{ef}(*,*) = 0$ *and* $c_e(*) = 0$. As we iterate through the parallel corpus we update the counts with delta (δ). $\delta(k, 5, 6)$ can be thought of as the probability of word f_5 being aligned with word e_6 in the data. We estimate

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

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j)$$
$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k,i,j)$$

and we update $t(f/e) = c_{ef}(e,f)/\ c_e(e)$ after each EM iteration

1.4 Results

Dataset	Precision	Recall	F1- Score
Dev	0.412	0.425	0.419

With our choice of initialization, we get an F1-score of 0.419. Below we try another initialization just to see whether the numerical results are stable.

1.5 Discussions

Below are the results for 5 EM iterations:

```
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.211          0.218         0.215
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.372          0.384         0.378
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.401          0.414         0.407
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.408          0.421         0.415
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.412          0.425         0.419
```

In each iteration, we estimate the maximum likelihood parameters "t" and use those to recalculate the counts . We also see that with each the f1-score seems to converge. We try with different initialization (1/len(spanish_vocabulary)) and see the score converge to something very close. It also shows that the EM is a stable algorithm numerically

```
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.204          0.211         0.207
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.376          0.389         0.382
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.405          0.418         0.411
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.409          0.422         0.416
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920      0.414          0.428         0.421
```

IBM Model 2

2.1 Description of IBM Model 2:

In IBM model 2, we introduce parameters "q" to model the alignments between the words in two corpora. IBM model 1 can be thought of as a special case of IBM model 2 where all the alignments are equally probable. This is the reason, it is better than IBM model 1, as not all alignments are equally likely (as mentioned in the limitations of IBM model 1).

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})} \quad \text{and} \quad q_{ML}(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$$

Limitations:

Although it addresses the problem of alignment by adding additional step for modeling the alignments, it still doesn't address how to take care of one word producing multiple words or no words at all when translated. Also, there is the problem where multiple spanish words can have the same english word when translated and some English words are not aligned to any spanish word (problem with all word based MT systems).

2.2 Method Overview:

We start from the t values we calculated in our IBM model 1 implementation as the starting point(initialization) and implement IBM model 2 the following way. We again iterate through 5 EM iterations and in each iteration we do the following:

We initialize the counts $c_{ef}(*,*) = 0,\ c_e(*) = 0,\ c_{jilm}(*,*,*,*) = 0,\ c_{ilm}(*,*,*) = 0$ *and* $q(j/i,l,m)$ *to* $1/(1+l)$. As we iterate through the parallel corpus we update the counts with delta (δ). We estimate

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}, \quad \begin{aligned} c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\ c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \end{aligned}, \quad \begin{aligned} c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\ c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j) \end{aligned}$$

and we update $t(f/e) = c_{ef}(e,f)/\ c_e(e)$ *and* $q(j/i,l,m) = c(j/i,l,m)/c(i,l,m)$ after each EM iteration

2.3 Results

Dataset	Precision	Recall	F1- Score
Dev	0.442	0.456	0.449

As you can see the results (both precision and recall) are clearly better than IBM model 1. So, rather than treating all the alignments equally probable, creating an additional step to model the alignements clearly helped in the translation.

2.4 Discussions

Below are the results for 5 EM iterations:

```
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920        0.432        0.446         0.439
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920        0.435        0.449         0.442
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920        0.439        0.453         0.446
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920        0.439        0.453         0.446
      Type        Total    Precision       Recall      F1-Score
============================================================
     total         5920        0.442        0.456         0.449
```

Two correctly (or almost) aligned examples:

1. Sentence 192 in the dev set:

Word Number		1	2	3	4	5	6	7	8	9	10	11	12
		una	cuestión	que	nos	separa	es	la	guerra	civil	en	chechenia	.
1	one	K,P											
2	issue		K,P										
3	that			K,P									
4	separates					K,P							
5	us				K,P								
6	is						K,P						
7	the							K,P					
8	civil								K,P				
9	war									K,P			
10	in										K,P		
11	chechnya											K,P	
12	.												K,P

2. This is an example of a correct prediction where P are the prediction alignments and K is the key provided. The Spanish sentence is horizontal and the corresponding English sentence is vertical. Below is another example (sentence 35)

Word Number		1	2	3	4	5	6	7	8
		gracias	por	sus	palabras	,	señor	comisario	.
1	thank	K,P							
2	you	K							
3	for		K,P						
4	your			K,P					
5	statement				K,P				
6	,					K,P			
7	commissioner						K,P	K,P	
8	.								K,P

Here, except that the word “you” is not aligned with any word in Spanish, everything else agrees with gold standard.

Two misaligned examples:

1. Sentence 46: K for gold standard in dev.key and P for prediction by IBM model 2

Word Number		1	2	3	4	5	6	7	8	9	10	11	12	13
		10	-	limita	considerablement	las	exenciones	vigentes	para	las	instalaciones	en	cementeras	;
1	10	K,P												
2	.													
3	considerably				K,P									
4	limits			K										
5	the					K								
6	present							K						
7	derogations						K							
8	for								K					
9	cement					P	P			P	P	P	K	P
10	kilns		P	P				P	P		K		P	
11	.													K

2. Sentence 71:

Word Number		1	2	3	4	5	6	7	8	9	10	11
		en	este	momento	aún	debemos	avanzar	considerablemen	en	ambos	aspectos	.
1	at	K,P	K	K					P			
2	present	K	K	K								
3	,											
4	there											
5	is											
6	room			P	P	P	P	P				
7	for											
8	considerable		P					K				
9	improvement						K			P		
10	with								K			
11	regard								K			
12	to								K			
13	both									K		
14	of											
15	these											
16	aspects										K,P	
17	.											K,P

To get the above examples, I used intersection over union heuristic for best and worst alignments. We can defend why the model performed well for the first two examples but in case of next cases, the second one specifically, the word avanzar means advancement but semantically in the training corpus we see many different alternatives like move forward, advance but not the word improvement too often. This could be the reason the model couldn't find the alignment in the English sentence. In sentence 46, after looking at the training data, we see that in the entire training data we see cement, kilns only once and this might have made it hard for our model to get it right.

2.5 Critical Thinking

Possible ways to improve the model

1. We know that some words get dropped, some words are aligned to more than one word and some to just one word. Both our IBM models 1 and 2 fail to capture this aspect. Hence, we can introduce additional parameters to address this. Like for example, a model for the distribution of number of words each Spanish maps to in English (which includes 0 to capture dropping)
2. In the above two models, we don't bring the context anywhere and they are just word based. One improvement could be to use context like word classes and to take into account the previous word's alignment.

2.6 Non-trivial ideas for improvement

In IBM models 1 and 2 each alignment is treated independently by not taking into account of what are already aligned. It is more likely that words that are already aligned have less chance being aligned again. We could bring additional parameters to keep track of the alignments that are already taken and using only the places that are available.

Growing Alignments

3.1 Method Overview

For this part, we implement the method given in the write-up (and in the class). After we estimate p(f/e) using IBM model 2, we do the same thing but the languages reversed (with English as foriegn language) and estimate p(e/f). We get alignments from both the estimates and then we take the intersection of those alignements and their union. Now starting from the intersection we grow the alignments confined to those in the union. We grow them using the neighbours approach where we only consider the ones adjacent or diagonal to the existing ones. So starting from the intersection, we grow if an alignment exists in any of the following positions when added with {(1,0),(0,1),(-1,0),(0,-1),(1,1),(-1,-1),(1,-1),(-1,1)} to the current position.

After we grow the neighbouring alignments, we can fill the remaining alignments in the union using other heuristics. For example, in this exercise I considered growing only those where both the index of an English word and a Spanish word are not in the existing set of alignments. This did not improve the f1-score though.

3.2 Results and Discussions

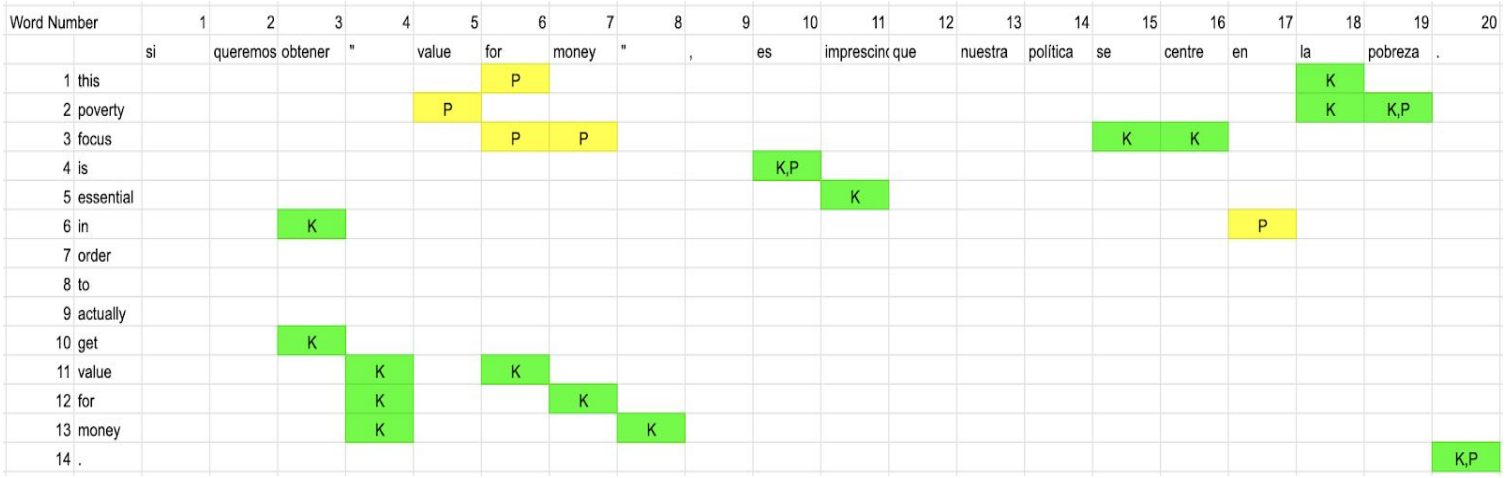
Using the above heuristic the results are as follows:

Dataset	Precision	Recall	F1- Score
Dev	0.657	0.467	0.546

Here is an example for correct(almost) alignment with one Spanish word aligned to more than one English word.

Word Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	ha	sido	una	pena	que	las	conclusiones	de	lisboa	no	exigiesen	un	calendario	.
1 it														
2 was	K	K												
3 a			K,P											
4 pity				K,P										
5 that					K,P									
6 the						K,P								
7 lisbon									K,P					
8 conclusions						P	K,P							
9 did											K,P			
10 not										K,P				
11 call											K,P			
12 for											K,P			
13 a													K,P	
14 timetable														K,P
15 .														

Misaligned example:



Word Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	si	queremos	obtener	"	value	for	money	"	,	es	imprescin	que	nuestra	política	se	centre	en	la	pobreza	.
1 this						P												K		
2 poverty					P													K	K,P	
3 focus						P	P								K	K				
4 is										K,P										
5 essential											K									
6 in			K														P			
7 order																				
8 to																				
9 actually																				
10 get			K																	
11 value				K		K														
12 for				K			K													
13 money				K				K												
14 .																				K,P

The misaligned example has a quote in English that is being translated to English causing the model to fail. In case of the correctly aligned example, everything was aligned correctly except the starting words because they failed to make into our growing alignment set. The reverse task we did (translating from English to Spanish) had no alignements for those starting Spanish words.

3.3 Critical Thinking

I tried different heuristics to try to improve the performance. For example, neighbours beyond one (0,2),(2,0),(1,2),(2,1),(-1,2) etc and then adding the alignments where both e, f are not neighbours and both the indices (i,j) are not in the existing grown list but in the union. It did not improve the performance and below are the results:

Dataset	Precision	Recall	F1- Score
Dev	0.645	0.468	0.542

One possible way to improve the method of growing alignements:

After we get the intersection of alignments from p(f/e) and p(e/f) we grow the alignments using neighbours that are in the form of phrases. This is the basis of phrase based translational models. For any phrase (f,e) we observe in the training data we can give a score log(c(e,f)/c(e)). This can be interpreted as an estimate of the log-conditional probability of foreign phrase f, given English phrase e and in a sense a heuristic t grow alignments.

3.4 Non-trivial improvements

When we try to include the dependence on previous words in our model - like HMM for example, we can improve it using interpolation or smoothing (as have seen in language modelling). Another improvement could be using a dictionary for translations (as additional parameter) and incorporate that into our model. $\mu(e) * p(f/e)$ where μ is the size of the dataset considered for modelling p(f/e)

3.5 Different alignment heuristics

1. We can do something similar that we proposed earlier - a model to capture the number of words each word in the source language is getting mapped to target language. We can do something similar like the pointwise mutual information and try to model the the co-occurrence of (e,f) / c(e)*c(f).
2. One more idea is to include word classes in alignments. We can bring the dependence on previous words using HMM like we did in sequence tagging. The difference being we are trying to model the alignments numbers rather than the words themselves.

REFERENCES:

- http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf
- http://cseweb.ucsd.edu/~nnakashole/teaching/256_sp19.html - lectures