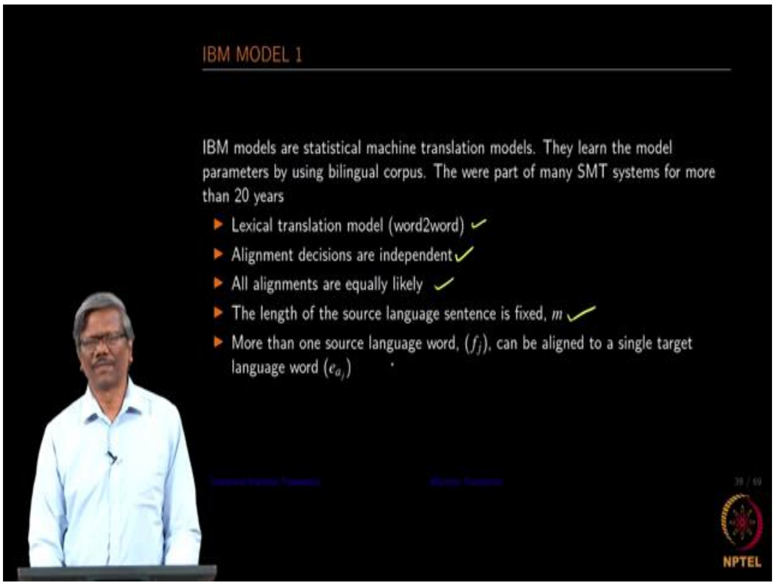
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture – 67
IBM Model 1

(Refer Slide Time: 00:15)



IBM MODEL 1

IBM models are statistical machine translation models. They learn the model parameters by using bilingual corpus. The were part of many SMT systems for more than 20 years

- Lexical translation model (word2word) ✓
- Alignment decisions are independent ✓
- All alignments are equally likely ✓
- The length of the source language sentence is fixed, m ✓
- More than one source language word, (f_j), can be aligned to a single target language word (e_{a_j})

So, now we will talk about some of the models that research researchers have created ok. So, this is one of the very interesting research topics in the early 80s and 90s; people were trying to really estimate the parameters from the data that is available, ok. And IBM researchers have really found some ways to estimate those data from them or the model from the data that is available to them ok. So, these are the seminal models that IBM has created and many translation models were dependent on this particular model, that we are going to be describing here.

So, this is not really used nowadays, modern systems do not use IBM models nowadays for the translation; but there are certain translation probabilities that we had shown earlier, and the alignment model that we have shown earlier would be useful and the modern translation models. So, for that sake we definitely have to understand what IBM models are and there are about five models in this and I am going to be only taking Model 1 and 2 for this session, alright ok. This model that we have chosen as IBM

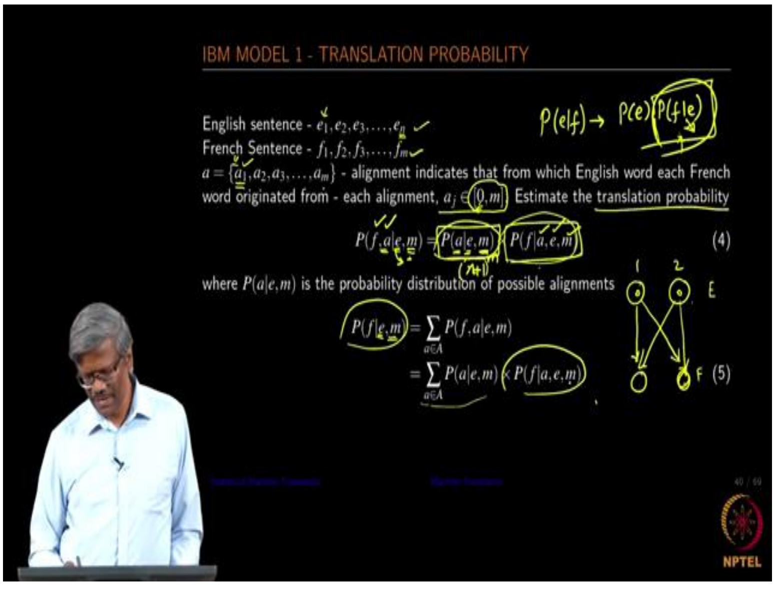
Model 1, is a lexical translation model it is a word to word translation that uses the alignment that we had seen earlier.

In this case, the alignments are independent; the decision that we are going to be making in terms of aligning the word Le with the, is independent you know it does not depend on the previous word or the next word and so on ok. So, it does not have any contextual knowledge that is why these are decisions are independent.

And then all alignments are equally likely; that means, since it is going to be computing the alignments, so it has to start from somewhere. So, it thinks that every alignment is possible. So, all alignments are equally likely that is where it starts it. And we do not use any of these models, especially the models using the lexical translation which in turn uses a word by word translation, right. So, now, we have gone ahead and we use something called a phrase-based model; phrase-based models are very advanced, we will talk about that in the next session.

And they utilize some aspects of alignment and translation probability from the IBM model that is why we are studying this, ok. The length of the source or language sentence is fixed; we saw that earlier right. So, fix the size of the or the length of the French sentence to a fixed number and then move from there, ok. And it is possible that more than one source word can be aligned to a single target language word, right this also we saw. So, these are the simple assumption that is made in the IBM Model 1.

(Refer Slide Time: 03:41)



IBM MODEL 1 - TRANSLATION PROBABILITY

English sentence - $e_1, e_2, e_3, \ldots, e_n$

French Sentence - $f_1, f_2, f_3, \ldots, f_m$

$a = \{a_1, a_2, a_3, \ldots, a_m\}$ - alignment indicates that from which English word each French word originated from - each alignment, $a_j \in [0, m]$. Estimate the translation probability

$$P(f, a|e, m) = P(a|e, m)P(f|a, e, m) \quad (4)$$

where $P(a|e,m)$ is the probability distribution of possible alignments

$$P(f|e, m) = \sum_{a \in A} P(f, a|e, m)$$
$$= \sum_{a \in A} P(a|e, m) \times P(f|a, e, m) \quad (5)$$

And then for you to really translate from the French to English, you know we use the noisy channel model where we want to find out. So, we had the language model and then we had the translation model, right. So, since it is very difficult to estimate this directly, alignments are brought into the picture ok. So, let us see how the alignments really help in terms of breaking this into smaller steps, right. So, any model that you want to create; if you are not able to create using the original assumption you start breaking that ok.

You start breaking that into small pieces and then each piece you call it as a smaller model within itself and then try to estimate the parameters of those models and finally, end up solving this. So, in the same fashion here also, since it is not directly possible to estimate this, we are choosing the alignment as our option to break this into smaller chunks. So, again in this model we are going to be considering the English sentence, whose length is going to be driven by this number n and we are going to use a French sentence whose length would be given by m ok.

And then we are going to use an alignment variable; the alignment variable will be a set of alignment variables, where a_1 corresponds to the alignment from French to the English word right. So, that means, this corresponds to the value in the English sentence; for example, if this is the first word in English we will write this as 1, this we have seen few times the earlier, right.

So, I do not need to explain this further. So, this is an alignment that indicates from which English word each French word originated from ok. We also use a 0 here, as I mentioned earlier null would be integrated as part of the alignments; if there is a word that I am not able to really align it with French and then I align that particular French world with the null word. So, that is given as the index has 0 here ok. Using the chain rule again, the translation probability is broken into two models, ok.

So, one is the alignment model and another one is the translation model. So, this is nothing, but the probability of the French word, given the length of the French sentence and the English word and an alignment word, right. So, in this case we are going to be finding these two provided e and m are available to us, right. So, there are two models here one is independent of the French. So, it only worries about the count in the French sentence, right.

And then given the number of words that we want to constrain and then the English sentence you will find the alignment ok. Again if you take the second one, so you want to find the probability of the French word, given the alignment, the English sentence, and your condition that on the number of words in French sentence. So, this gives you a probability distribution, right.

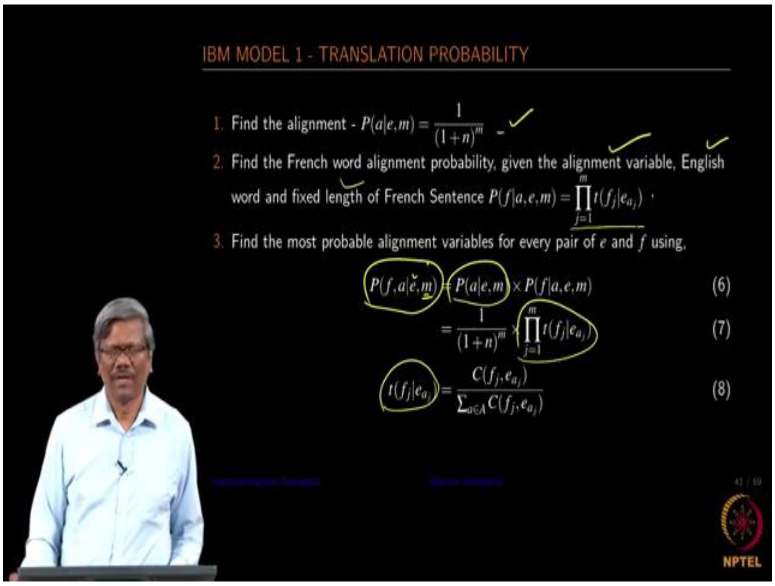
So, it is not going to be one single value that you will get when you have parts of the sentence and you use the translation model to find the alignment probability and the translation property, it is going to be a distribution ok. So, when you want to find the translation probability which is conditioned on the number of words in French and the English word; we can rewrite this as the sum of you know, it sums over all the alignments possible, you know how we have so many alignments, right. So, I do not need to go over that.

So, what we had shown earlier is only one type of alignment. If I have, let us say that there are three words or let me consider only two for the sake of. So, I can align it in this fashion there is one alignment possible, the second alignment possible right, and then third alignment possible, fourth alignment possible, right. So, this word could be let us say if this is the English and then this is French; this French word could depend on or could align with the second word in English, it can also align with the first word in English.

So, there is a possibility of several alignments that you will have since we are going to be having n-words in English; so the number of possible alignments is n plus 1. And then since we are going to be doing it over the words in French, you will have this number; so the total number of possible alignments are n plus 1 to the power m, alright.

So, the translation probability now is rewritten with respect to all the alignments possible, the sum of all the alignments that you have, and then again the probability of finding the French word given me and the alignment alright ok. So, this is very crucial to understand this equation.

(Refer Slide Time: 10:50)



IBM MODEL 1 - TRANSLATION PROBABILITY

1. Find the alignment - $P(a|e,m) = \frac{1}{(1+n)^m}$
2. Find the French word alignment probability, given the alignment variable, English word and fixed length of French Sentence $P(f|a,e,m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$
3. Find the most probable alignment variables for every pair of e and f using,

$$P(f,a|e,m) = P(a|e,m) \times P(f|a,e,m) \tag{6}$$

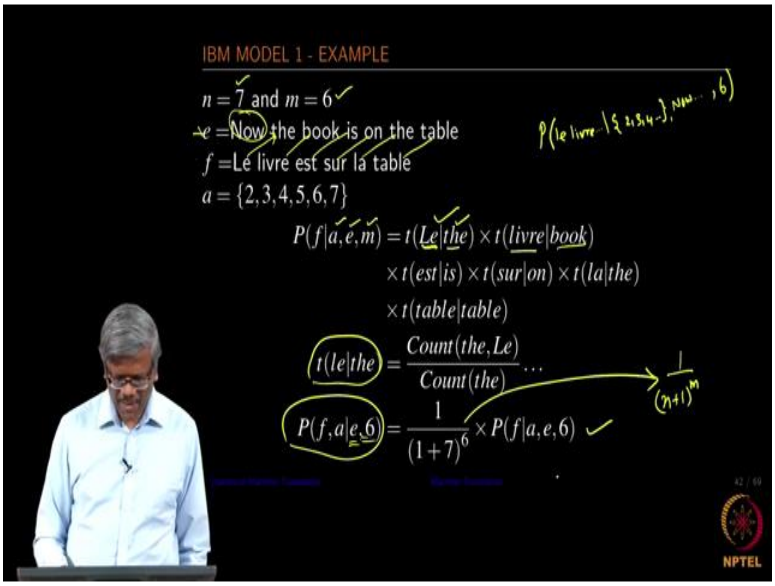
$$= \frac{1}{(1+n)^m} \times \prod_{j=1}^{m} t(f_j|e_{a_j}) \tag{7}$$

$$t(f_j|e_{a_j}) = \frac{C(f_j,e_{a_j})}{\sum_{a \in A} C(f_j,e_{a_j})} \tag{8}$$

So once we have done this. So, the translation probability can be computed based on what are you going to be describing here, ok. So, we know that the alignments are equally likely, right. And then to find the French word alignment probability what do you need; you have the alignment variable, you have the English word and then you have the fixed length of the sentence; you can find the translation probability using this, right. So, that it is the product of the translation probability of the French word, given the English word to which it is aligned.

So, this is a product, we will show by an example of how it is computed ok. And then if you replace those values in this equation, what you get is this, ok. So, in the end what you get is, this is based on the count, right. So, we can find the count. So, this is how the translation probability is computed and then we have the alignment probability as well. And then using which now we can compute the probability of the French word, the alignment gave the length of the French word and the length of the French sentence and the English word, ok.

(Refer Slide Time: 12:27)



IBM MODEL 1 - EXAMPLE

$n = 7$ and $m = 6$

$e =$ Now the book is on the table

$f =$ Le livre est sur la table

$a = \{2,3,4,5,6,7\}$

$$P(f|a,e,m) = t(Le|the) \times t(livre|book) \times t(est|is) \times t(sur|on) \times t(la|the) \times t(table|table)$$

$$t(le|the) = \frac{Count(the, Le)}{Count(the)} \ldots$$

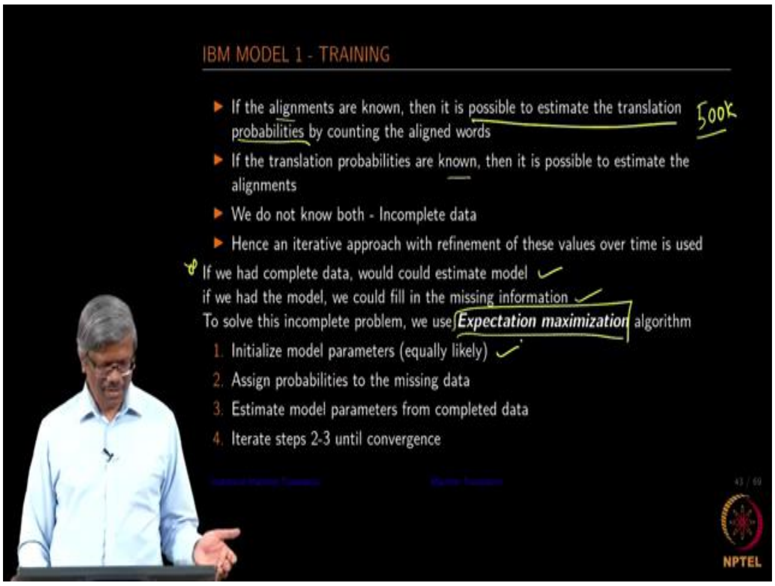
$$P(f,a|e,6) = \frac{1}{(1+7)^6} \times P(f|a,e,6)$$

So, let us take one small example and then see how we can compute it for one alignment. Let us assume that we have n equal to 7 which is the total number of words in the English sentence and m equal to 6 the total number of words in the French sentence. And we have the English sentence here right and then we have the French sentence. And then the alignment if you look at, this is not aligned; in this case we have 2 3 4 5 6 and 7 are aligned to this, right.

So, if you want to find the probability of the French word, given e and the English sentence and the alignment value; so how this is translated? So, we want to find out P of this sentence right, given the alignment variables 2 3 4 and so on and the English sentence and finally the total number of words. So, this is actually it is translated in this form. Now we need to estimate the translation probability, ok.

So, we can now find the translation probability using for every English word that is aligned or for every Le aligned to them and leave the book and so on. You take the product we mentioned that earlier, right. And then for every translation probability you get the count, I spoke about that earlier how we can get the count of this. And then if you want to find out the probability of the French word and the alignment, given the English word and the number of words in the French sentence and this is how if you do it, ok. So, you have this is one by n plus 1 to the power m right. And then the value that you obtain here, is fostered and that is computed ok.

(Refer Slide Time: 15:05)



IBM MODEL 1 - TRAINING

- If the alignments are known, then it is possible to estimate the translation probabilities by counting the aligned words
- If the translation probabilities are known, then it is possible to estimate the alignments
- We do not know both - Incomplete data
- Hence an iterative approach with refinement of these values over time is used

If we had complete data, would could estimate model

if we had the model, we could fill in the missing information

To solve this incomplete problem, we use ***Expectation maximization*** algorithm

1. Initialize model parameters (equally likely)
2. Assign probabilities to the missing data
3. Estimate model parameters from completed data
4. Iterate steps 2-3 until convergence

Let us look at the training of this first, ok. So, this is a very difficult problem to crack ok. So, our translation is not an easy problem, it is a very hard problem to solve, it is an np-complete problem, right. So, if in this case if the alignments are known, that is a very ideal condition then it is possible to estimate the translation probabilities, right. So, we know how to do that, when we know the alignments, beforehand. If the translation probabilities are known, then it is possible to estimate the alignments.

So, unfortunately we do not know both, we what we have is only incomplete data you know; if you look at the number of sentences that you want to translate or use it for estimating the model, it is pretty huge right. As I mentioned, you need to have really a good set of or good pairs of sentences in order for to create, for you to create a model right, in order for you to create a good model.

So, in this case we cannot expect the manual labor to go into this to really align them. So, what you can probably give is, this sentence is the same as or this sentence in English is the translated version of the French sentence that you have here. So, that is how we can provide. So, we can only provide the translate, we can only provide the sentences or aligned sentences and not the aligned words.

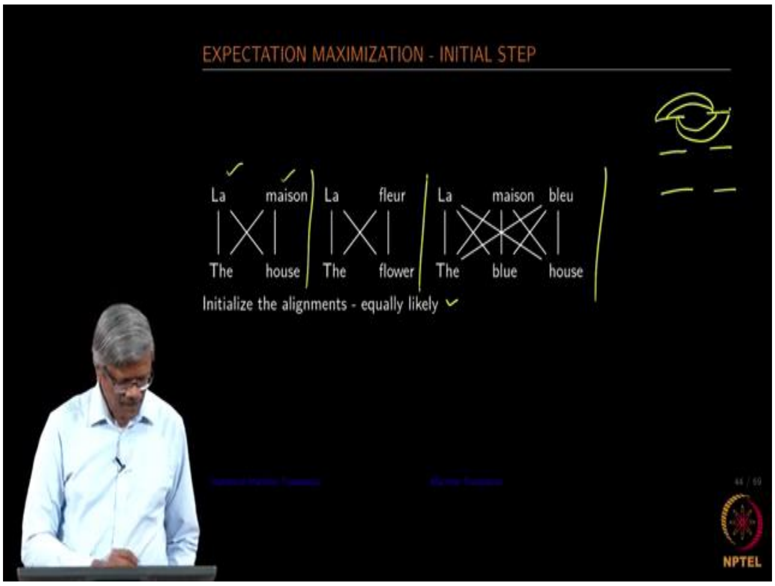
So, it is going to be hard to estimate that. So, what we do is, we need to figure out an iterative approach and then refine the values as we go along ok. So, what we have is an incomplete data. So, we need to fill in the missing information as we go along in the iterative process. And then finally, come out with the model that we can use it to decode the incoming sentence, ok. So, in this case we would use the em algorithm, to solve this problem. So, as I mentioned if you had the complete data, we can easily estimate the model.

If you had the model, we can fill in the missing information both are not available to us in full. So, that is why we go for the m algorithm here. So, how do you start this, you initialize the model parameters; what are the model parameters that we spoke about? The one, this is one model parameter that we want to find out ok; and then this is another one that we want to estimate. So, in this case, both are an incomplete way that is why we have to use an iterative approach to fill in the missing information and then finally, go from there.

So, we initialize the model parameters and then assign the probabilities to the missing data. Estimate the model parameters from the completed data. And then iterate 2 and 3 until the conversion happens ok. It is very similar to what we saw in the initial Swahili model ok, we have about 3 or 4 sentences right. We actually were doing the e m approach in terms of a figuring out of which word in English is aligned with the Swahili word or the vice versa, if you are doing from Swahili to English translation.

So, we found that while doing the exercise, we found out certain words aligned with certain words several times so; that means, we can pick up that alignment and call it as the alignment probably that I want to use, if I encounter that particular word. So, this e m algorithm exactly does what we have done in a manual way.

(Refer Slide Time: 19:43)



EXPECTATION MAXIMIZATION - INITIAL STEP

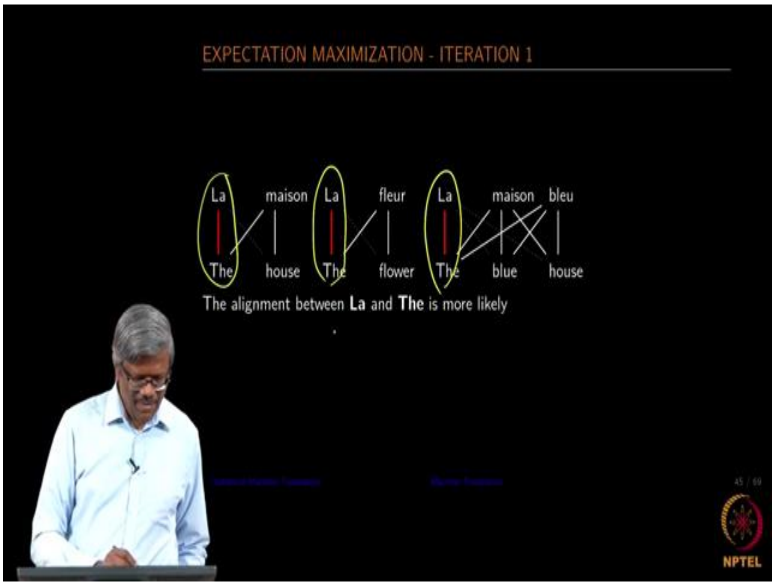
La maison | La fleur | La maison bleu
The house | The flower | The blue house

Initialize the alignments - equally likely

So, to take you through these steps here, I have three sentences. So, this is taken from one of the books or I think from one of the research papers that I had seen. I think it is most likely the covenant's paper on statistical machine translation. So, we have three sentences here, and then which are aligned; because we mentioned that the alignments are equally likely right.

So, La is aligned to house, La is aligned to the Maison is aligned to the, and then Maison is aligned to house ok. So, this is how we initially align. So, you will only get those sentences without any alignment. So, now, we have made those alignments right like this, alright.

(Refer Slide Time: 20:51)



EXPECTATION MAXIMIZATION - ITERATION 1

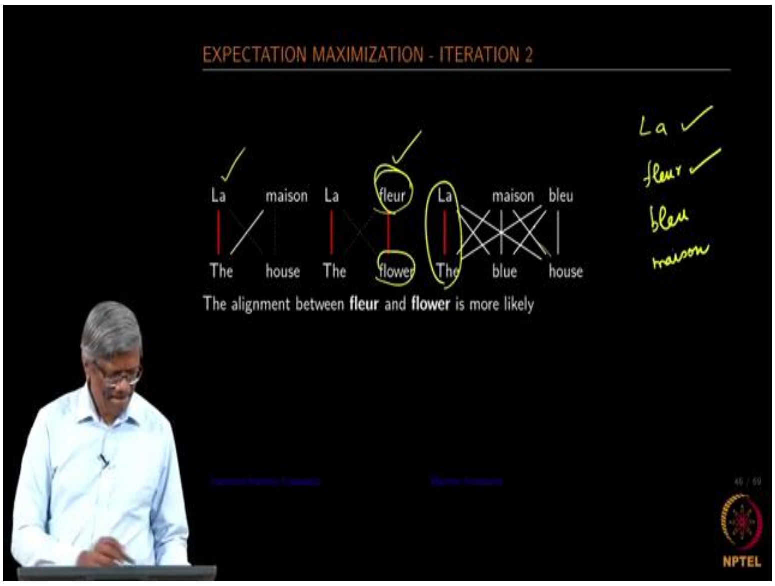
La maison | La fleur | La maison bleu
The house | The flower | The blue house

The alignment between **La** and **The** is more likely

So, as you move along in this iterative process, what happens is you start finding out La is aligned to a few more times than any other words; then we can start isolating that, this is the this has a higher probability than any other alignment. So, we can give up or the computational process will give you value higher than any other alignment in this case. La and the house will have a very low value.

So, that is why it is in the dotted form, ok. So, as we move along in this iterative process, we will start figuring out that the flower is aligned to this French word, ok. So, we figured out La, we can figure out the second one, and then the third one we have to figure out; how many words we have in this?.

(Refer Slide Time: 21:52)



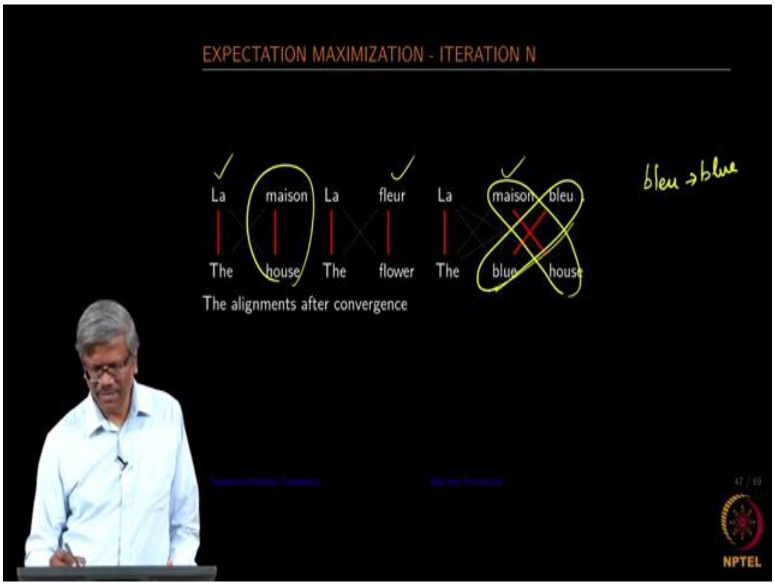
EXPECTATION MAXIMIZATION - ITERATION 2

The alignment between **fleur** and **flower** is more likely

We have La, that we need to align right. So, we found this and then based on the alignment this particular word is not aligned with anything else except the; but the is already or La is already aligned with the. So, it cannot be aligned to, the right. So, this is aligned. So, in the third sentence we already found this. So, now, we still have to figure out the alignment for Maison and bleu, right.

So, as you iterate more and more, you will see that in the earlier case there is one alignment that is possible between these two paths of words. So, we can assume that this aligned to this, and then finally, the missing one is aligned to this.

(Refer Slide Time: 23:05)



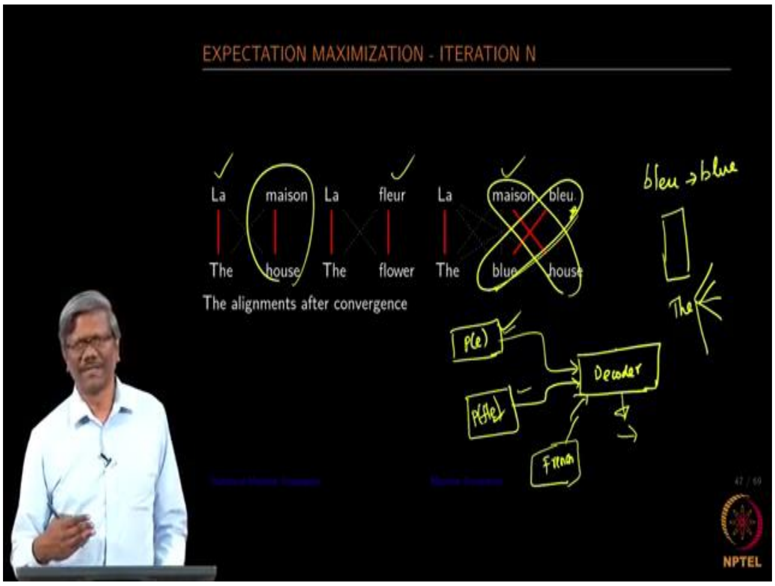
EXPECTATION MAXIMIZATION - ITERATION N

The alignments after convergence

So, you do not see it several times. So, whatever is missing now just fill that since the alignments are equally likely, initially all are aligned to each one of those and then we slowly iterated this process and figured out each one of these ok. And then since the alignments are one on one in this case, we can assume that this will be aligned to that, ok.

So, there will be some marginal error in this case, in the case of bleu ok. So, that is the first model IBM Model 1. So, we have used the data to really find the translation probability, here. This is our translation probability and then what else is missing. So, at the end of this, we need to have a decoder, the decoder will have two things for it, right. So, one is the.

(Refer Slide Time: 24:18)



EXPECTATION MAXIMIZATION - ITERATION N

The alignments after convergence

So, let us say this is the decoder. So, the decoder receives data from the language model and the translation probability model right, so, so both are combined. So, when you combine this, the decoder now should be able to identify all the possible translations for each of the words. And then for every word that you have in this, the language model will find out what is the next word, possible next word ok. Again if you look at this the language model will throw in the Softmax right. The for example, if, the is the first word and you want to estimate the next word here right.

So, it could be anything in the corpus through which this one was trained, there could be several possibilities that you may have correct, right. So, we need to combine the language model, we need to combine the translation probability model and then finally, get a huge collection of sentences, right. So, this language model is going to throw in several combinations for you took, and you cannot ignore them. And then, based on the alignment the translation model also is going to throw a huge collection of translation probabilities.

So, when you combine the language model and this, you are going to have a very huge collection of translated sentences; not all of which are very meaningful. So, we need to find out the ideal sentence; that really represents the French sentence that we started with, ok. So, the decoder's job is to really search through the large collection of sentences that it has created; and finally, find out what is the right sentence that makes and the decoder should be able to find the right translated sentence based on the heaps of sentences that it has found, ok.

So, that is the job of the decoder, right. So, once this is trained. So, what going to happen is we are going to be giving a new sentence and then the decoder, so; for example, this is the French sentence that I am providing, the decoder should output the translated version of this. So, using these two models that we have computed ok. So, this model is pretty naive I do not think it is very useful; but it really gets us started with respect to finding the alignment for us.