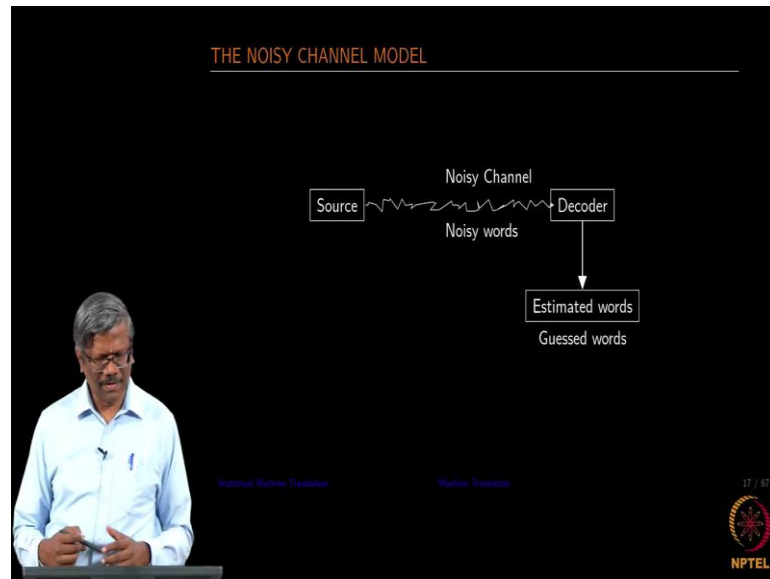


Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 65
Noisy Channel Model, Bayes Rule, Language Model

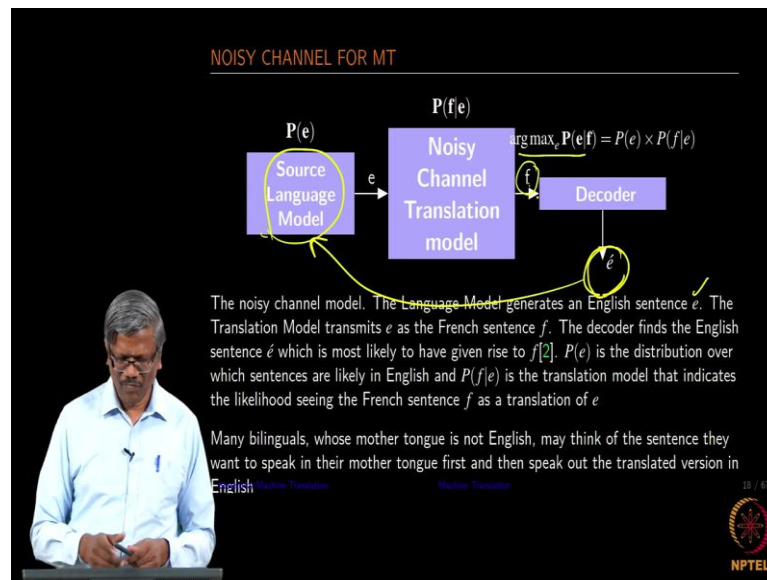
(Refer Slide Time: 00:15)



Another one that would like you to understand is the Noisy Channel Model as I mentioned earlier right. So, we can look at there is look at a source and there is a target right. So, when the source is transmitting certain data through the noisy channel and then the decoder receives the corrupted version of the source ok. The idea of the decoder is to really estimate what really created the data that is received.

So, ideally it should find the exact source of it right. So, in real situations since it is corrected by the noise it is not possible for you to exactly get this source. So, we get some estimated data. So, are some guessed data and then the guessed data would be close to a certain confidence level that we prescribe. So, if it is close to that confidence level that we have prescribed then we pick up that as the probable source and move on.

(Refer Slide Time: 01:33)



If you map the noisy channel for the machine translation we have the source language model that generates the source language and then the noisy channel can be considered to be your translator ok. So, given an English sentence translate it into French and assuming that the noisy channel receives the language receive the sentence in English and the translated model outputs the French sentence and then what we need to really estimate is what actually the source had in mind or the source language model had in mind and that is what we want to find out. So, decoder actually tries to estimate the source which created the French language ok.

So, this is the idea here in this case again there could be multiple decoders might be giving multiple versions of e dash. We use the odd map we really find out what exactly or what could be the probable sentence that I want to use as the translated version. So, the language model here generates an English sentence e ok. So, as I mentioned it generates an English sentence the translation model transmits e as a French sentence as it is output ok.

The decoder finds the English sentence e dash which is most likely to have given rise to f . So, we want to find out which one really would have created this; that means, we are going back into this ok. So, if you are able to really estimate the accurately what really created f then we have really gotten the right or the best-translated version of the source ok. So, I think this is if you are bilingual you understand this extremely well.

Now, as a bilingual you know when you start to speak English and your mother tongue is not English you form your sentences using certain words in your native language in your mind and then output the English sentence ok. What the decoder really wants is what you really had in your mind that output the English sentence so that means, I want to really understand what kind of words and the language model that you have used internally to generate the translated version of English and then I just want to estimate that source that it created the English language.

So, this I think is very common for people who learn to speak English. Initially they frame their sentences in their native in their mother tongue and then speak out the English sentence ok. So, this model is exactly similar. We want to estimate what really created this f. So, we want to estimate this source using the output that it was generated and then estimate to a certain level of probability and then see how close we can match with the language models that we had used because the language model gives us you know what is the probability of this sentence that is created here right.

So, if the sentence is not in the right order then the probability is going to be very low right. So, we have already seen that there is a probability model that we can generate for a language using bigrams and trigrams we will have some recap after this and then use that or apply that onto the decoded language and then see how close it is ok. So, that so we would be able to estimate this and then find out whether this e that is a decoded is good enough to be called a good translated sentence using the model ok.

(Refer Slide Time: 06:35)

BAYES' RULE FOR MT

By applying Bayes' Theorem, the translation problem is broken down into two smaller problems. Assume that we have a French sentence f and we would like to translate into an English sentence e .

From the probabilistic perspective, we want to find the English sentence e that has maximal probability given the French sentence f . Using Bayes rule we can write this problem as

$$P(e|f) = \frac{P(f|e)P(e)}{P(f)}$$

We can find the English sentence using the $\arg \max$

$$\begin{aligned} \arg \max_e P(e|f) &= \arg \max_e \frac{P(f|e)P(e)}{P(f)} \\ &= \arg \max_e P(f|e)P(e) \end{aligned}$$

$P(f|e)$ - the translation model and
 $P(e)$ - the English Language Model
The problem is reduced to modeling these 2 distributions ✓
Now we have to estimate the parameters of the $P(f|e)$ from the training examples (f^k, e^k) for $k = 1 \dots n$

19 / 87
NPTEL

So, there is another one that we want to look at here this is a Bayes' rule. It is a very simple rule that breaks down the problem into two smaller problems. So, again we are going to be assuming French sentence and we like to translate that into an English sentence right. From the probability perspective we want to find the English sentence that has maximal probability given the French sentence. So, using this rule we can write the equation of this type where the probability of estimating this sentence in English given the French sentence is given by this form.

It is a product of the conditional probability and the product the language model and in the denominator we have a language model for French. Since it is the numerator is independent of this we can actually ignore this and we can write this as ok. We have to now estimate two things ok. We have 2 problems in hand; one is to estimate the parameters of this and then the next one is to make the model for this right.

So, we call this one as the translation model and this one is our English language model. So, if you are able to estimate this 2 distribution. So, we should be able to find out the English translation given the French sentence ok. So, this is something that we have obtained using the Bayes' rule ok. If you have to get the parameters for the translation model we have to estimate the parameters of this using all the training samples that we have in the parallel corpora ok.

(Refer Slide Time: 09:19)

BIGRAM AND TRIGRAM PROBABILITIES A quick refresh on the Language Model- P(e)

$$P(w_2|w_1) = \frac{f(w_1, w_2)}{f(w_1)}$$

$f(w_1, w_2)$ is the number of times w_2 appeared after w_1

$$P(w_3|w_1, w_2) = \frac{f(w_1, w_2, w_3)}{f(w_1, w_2)}$$

$f(w_1, w_2, w_3)$ is the number of times w_3 appeared after w_1 and w_2

20 / 67
NPTEL

I will just rush this through it is possible to estimate the word second word in a; in the bigram given the first word using the count right. So, in the same fashion in the trigram given 2 context words should be able to estimate the third one and so on again it is the; it is based on the counts ok. So, we know very well.

(Refer Slide Time: 09:47)

SPARSITY AND SMOOTHING A quick refresh on the Language Model- P(e)

- ▶ Newer ways of forming a sentence is common.
- ▶ It is possible that a trained model will see a new n-gram
- ▶ These new n-grams results in $P(x|y) = 0$
- ▶ $P(x|y) = 0$ will propagate through and produce a zero probability for the entire sentence
- ▶ Smaller probabilities too create a very small value

To avoid $P(x|y) = 0$, linear interpolation is used.

$$P(w_3|w_2, w_1) = \lambda_1 P(w_3|w_2, w_1) + \lambda_2 P(w_3|w_1) + \lambda_3 P(w_3|w_2) + \lambda_4 P(w_3)$$

where $\lambda_1(0.95) + \lambda_2(0.04) + \lambda_3(0.008) + \lambda_4(0.002) = 1$ ✓

For new words and n-grams, $P(x|y)$ will always have a small value

21 / 67
NPTEL

And then we also know that the data is sparse and there are always newer ways of constructing sentences. So, we should be able to allow the new sentences in the model and we do not want the model to fail when new sentences are given and if it does not

find the distribution for the new sentence it should not say that time I have not found anything right. So, there should be some mechanism to really smooth that the reason why we want to have that is you know if the conditional probability is 0 in one of those cases the entire sentence would fail.

So, we just want to avoid that zero probability. So, we will try to smoothen that using some mechanism here. So, even a newer sentence in this case will have a very small value.

(Refer Slide Time: 10:55)

KNOWLEDGE CAPTURED BY THE MODEL A quick refresh on the Language Model- $P(e)$

I want to eat Chinese food. I want English food. I want to eat english food

$P_1(\text{english} \text{want}) = 0.0011$	$P_6(\text{food} \text{english}) = 0.015$
$P_2(\text{chinese} \text{want}) = 0.0065$	$P_7(\text{food} \text{chinese}) = 0.15$
$P_3(\text{to} \text{want}) = 0.66$	$P_8(\text{chinese} \text{eat}) = 0.34$
$P_4(\text{eat} \text{to}) = 0.28$	$P_{10}(\text{english} \text{eat}) = 0.001$
$P_5(\text{order} \text{to}) = 0.18$	$P_{11}(i < s >) = 0.25$
$P_5(\text{want} \text{I}) = 0.32$	$P_{12}(< /s > \text{food}) = 0.12$

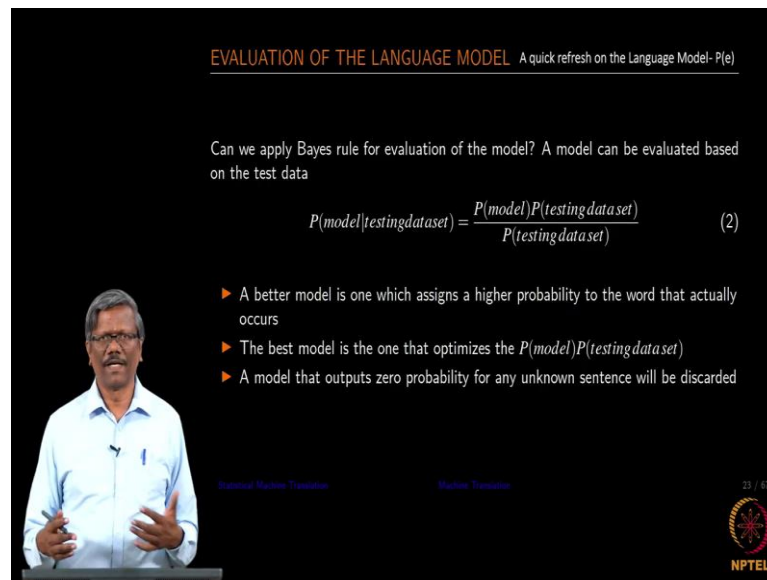
I want _____ food
I want to _____ food.

To avoid underflow values of multiplication to find $P(e)$, one can use \log
 $\log(P_1 * P_2 * P_3 * P_4 \dots P_n) = \log(P_1) + \log(P_2) + \log(P_3) + \log(P_4) \dots \log(P_n)$

22 / 67
NPTEL

We also know that we it is possible to estimate the parameters given the corpus and then using the corpus we know that it is possible to estimate the next word depending on the context that is provided by the previous words right or using the Word2vec model we should be able to get these words filled in either using this Skip- gram or CBOW. In some cases instead of the multiplication of all the probabilities we use the log so that the entire multiplication becomes additions right to avoid some underflow problems; alright.

(Refer Slide Time: 11:41)



EVALUATION OF THE LANGUAGE MODEL A quick refresh on the Language Model- P(e)

Can we apply Bayes rule for evaluation of the model? A model can be evaluated based on the test data

$$P(model|testingdataset) = \frac{P(model)P(testingdataset)}{P(testingdataset)} \quad (2)$$

- ▶ A better model is one which assigns a higher probability to the word that actually occurs
- ▶ The best model is the one that optimizes the $P(model)P(testingdataset)$
- ▶ A model that outputs zero probability for any unknown sentence will be discarded

NPTEL

So, how do we evaluate; I think there is always some gold standard available so that the translated version is how close it is to the gold standard ok. So, we probably would use the same Bayes' rule to evaluate this model as well, and then we also spoke about the perplexity earlier. So, the lower the perplexity the higher the confidence the model has with respect to the sentence ok. So, again this also we know well.