

Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute Technology, Kharagpur

Lecture – 09
N-Gram Language Models

Welcome to the forth lecture of second week. So in the last lecture we were discussing about a spelling correction and also seeing how do we do that it using the contexts. And remember the last problem that we are facing is how do you find the probability of sequence of words. So I have sequence of words in my sentence that might be candidate what the probability of that sequence. And that is why we said we will be using language models for it. So today the talk, that the topic of this lecture is to see what are my N-gram language models and we will start with some motivation of what are the different applications were we might need them. So N-gram language modeling is a very nice technique in NLP and it is applied in many different applications. It is very one of the very basic concepts in NLP. Let us see what are my N-gram language models.

So I will start with the motivation of context sensitive spelling correction that was a topic of just the previous lecture.


(Refer Slide Time: 01:29)

Context Sensitive Spelling Correction

The office is about fifteen minuets from my house

min-u-et noun \min-yə-'weɪʃl;
: a slow, graceful dance that was popular in the 17th and 18th centuries
: the music for a minuet

Use a Language Model
 $P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from})$



Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L1

Suppose I am having the sentence the office is about 15 minuets from my house. So here you can clearly see that this is just a spelling error minutes has been typed as minuets,

but suppose the word in minuet is also in my vocabulary. Suppose I see my vocabulary and I find that minuet is some sort of slow graceful dance that was popular in 17th and 18th centuries. So this word in my vocabulary, now, that means, I cannot easily detect that this is the incorrect word by using some isolated word as correction. So I should be using the context for that. So what is the probability that, so I will try to use some sort of language model to say with a probability of this a trends about 15 minutes from it is higher than the probability of a trends about 15 minuet from.

If you observe these 2 a trends you can yourself see that for it will should be more probable, but how do you formally defined these probabilities. And what is the model that we use for it. And that is what we will be seeing in this language model.

(Refer Slide Time: 02:47)

Probabilistic Language Models: Applications

Speech Recognition

- $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$

Machine Translation

Which sentence is more plausible in the target language?

- $P(\text{high winds}) > P(\text{large winds})$

Other Applications

- Context Sensitive Spelling Correction
- Natural Language Generation
- ...

Pravraj Goyal (IIT Khargpur) N-gram Language Models Week 2: L

So what are some of the other applications for this is helpful? Remember one of the earlier slides in our last week you were saying, in this case of speech recognition you face this problem that when you are uttering something you have to transcribe that. So whenever I am saying I saw a van it might also sound like eyes awe of n. So among the 2 a trends is which one is more likely. So again can I give some sort of probability value to each of these sequence in the say that probabilities I saw a van is more probable than the other.

Similarly, in machine translation there is a problem of collocations. That certain words even if they are correct translations do not occur much in the language with a in the

context of others like if I say if I have the word winds and before that as an adjective I can put either high or large. So for example, I have an Hindi (Refer Time: 03:48) [F] and I want to translate that into English – [FL] will be winds. Now for [FL] it might happened that both translations high and larger possible, in English. Now among those which one should I choose? And suppose from my corpus again find this language knowledge that high winds occur with more probability then large winds then I would say that high is a more probability translation than large.

And there are other applications is spelling correction. We have seen and you might have to use that for generation. Whenever you have to generate sentences again which particular generation has the higher probability?

(Refer Slide Time: 04:31)

Completion Prediction

- Language model also supports predicting the completion of a sentence.
 - ▶ Please turn off your cell ...
 - ▶ Your program does not ...
- Predictive text input systems can guess what you are typing and give choices on how to complete it.

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Ls

Another possible application that you might be using it in say google search or other places whenever you are trying to type something that they will be predict what will be the next word. So whenever I am saying please turn off your cell what will be the next one you can bit probably phone. Please turn off your cell phone. And your program does not probably compared or something. So given in a sequence of words how they predict what is the next for that is going to come in this sentence. So that is my auto completion task. That is huge a lot in a coyote completion or even when you are typing something in your SMS or (Refer Time: 05:08) you might have certain softwares that do that. So they also use the concept of language modeling.

So these are also called the predictive text input systems. So given whatever you are typing they will give you choices in how do you complete it. What does the various possibilities were, which will complete?

(Refer Slide Time: 05:32)

Probabilistic Language Modeling

- **Goal:** Compute the probability of a sentence or sequence of words:
$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$
- **Related Task:** probability of an upcoming word:
$$P(w_4 | w_1, w_2, w_3)$$
- A model that computes either of these is called a **language model**

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L1

So given these applications in background, let us see what is my language model what is the goal what do we gone to achieve. So we talked about this. In the previous lecture we said we I want to complete probability of w that is what is the probability of the sequence w 1 to w n. This is the sequence of words what is the probability of the sequence. So this is one of the gold. Compute the probability of a sentence or a sequence of words, P w. What is the other related task that we saw in prediction system that given 3 words w 1, w 2, w 3 what you will be my w4? So probability of w 4 given the 3 previous words, probability of an upcoming words.

Now, in general any model that computes either of these probability of the sequence or probability of the word given the previous a trend is called a language model. We were see language model using these definitions.

(Refer Slide Time: 06:35)

Computing $P(W)$

How to compute the joint probability
 $P(\text{about, fifteen, minutes, from})$

Basic Idea
Rely on the Chain Rule of Probability

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L

So now how do I actually compute probability of the sequence, w means sequence of words here. So suppose from the examples that we were taking initially I have the sequence about 15 minutes from. I want to compute the probability of the sequence. Now suppose I do not tell you anything else what is the simplest model that you will applied to compute this probability. So you will probably apply the chain rule of probability. Yes. So remember chain rule of probability.

(Refer Slide Time: 07:12)

The Chain Rule

Conditional Probabilities

$$P(B|A) = \frac{P(A, B)}{P(A)}$$

$P(A, B) = P(B|A) P(A)$

$P(A, B, C) = P(A) P(B|A) P(C|A, B)$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L

So it derives from conditional probabilities. So your conditional probabilities you might remember. So that is what is the probability of B given A that is probability of the joint. The joint probability $P(A, B)$ divided by probability A. Now you can also use that to writing in some other way. So you can say probability A B which probability B given a times probability A. And if you can do that for any number of words in my sentence or in general any number of events in probability. So you can say probability A B C which probability A probability B given a probability C given A B and you can keep on doing that finite numbers and that is my chain rule of probability.

(Refer Slide Time: 08:11)

The Chain Rule

Conditional Probabilities

$$P(B|A) = \frac{P(A, B)}{P(A)}$$

$$P(A, B) = P(A)P(B|A)$$

More Variables

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$

The Chain Rule in General

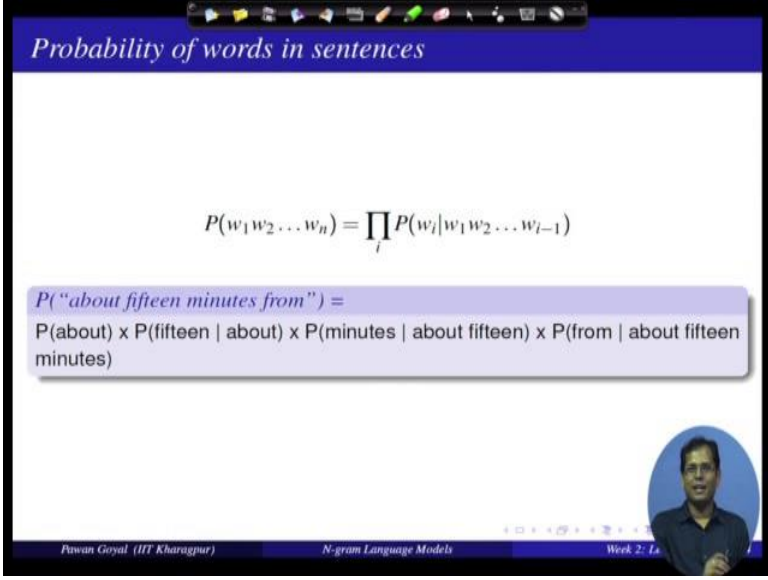
$$P(x_1, x_2, \dots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_n|x_1, \dots, x_{n-1})$$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lecture 4 7 / 24

I can write $P(A, B, A, P, A)$ times $P(B|A)$. In general, even if I am more variables I can do the same. Same chain rule of probabilities idea and this is the general formulation probability x_1 given x_1 is probability x_1 probability x_2 given x_1 and so on and probability x_n given x_1 to x_{n-1} .

So now given this chain rule now go back to an initial problem. Probability of about 15 minutes from, so how do I write it using the chain rule of probabilities?

(Refer Slide Time: 08:49)



Probability of words in sentences

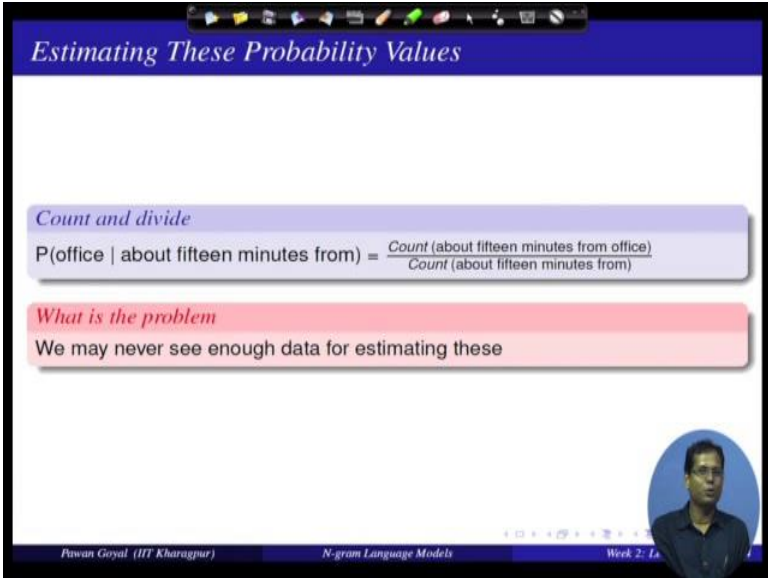
$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i | w_1 w_2 \dots w_{i-1})$$

$P(\text{"about fifteen minutes from"}) =$
 $P(\text{about}) \times P(\text{fifteen} | \text{about}) \times P(\text{minutes} | \text{about fifteen}) \times P(\text{from} | \text{about fifteen minutes})$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L2

So how will you right this probability? You will say this is same as probability of about times probability of 15 given about times probability minutes given about 15 probabilities from given about 15 minutes. So you can right it like that. Yes, now suppose I even increase this to about 15 minutes from office what will you do.

(Refer Slide Time: 09:15)



Estimating These Probability Values

Count and divide

$$P(\text{office} | \text{about fifteen minutes from}) = \frac{\text{Count}(\text{about fifteen minutes from office})}{\text{Count}(\text{about fifteen minutes from})}$$

What is the problem

We may never see enough data for estimating these

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: L2

And you will write it in terms of this probability of this given about 15 minutes from. So now, how the question that comes is, find, I can write it in terms of chain probability chain rules. How do I actually compute these probabilities? Now what is probability of

office given about 15 minutes from that would be in my corpus how many times do I observe about 15 minutes from. Among those how many times I observe office after that. Now what is one particular proper you will face in doing it in this work? You might not have seen sufficient number of about 15 minutes from in the corpus. Suppose it occurred only once it occur with the office. This probability will be one, but does not mean that only office can occur this about this occurrence. And this problem becomes severe as you keep on going to the higher and higher number of words in the conditional.

So we will never see enough data for estimating these. So I cannot estimate easily what is the probability of office given about 15 minutes from so that means, this will not work as it is. So we will need to do certain simplifications.

(Refer Slide Time: 10:32)

Markov Assumption

Simplifying Assumption: Use only the previous word
 $P(\text{office} \mid \text{about fifteen minutes from}) \approx P(\text{office} \mid \text{from})$

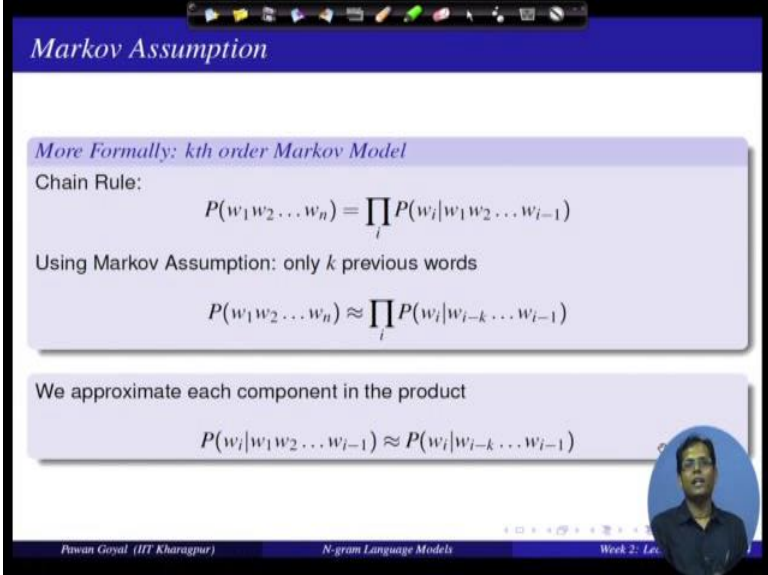
Or the couple previous words
 $P(\text{office} \mid \text{about fifteen minutes from}) \approx P(\text{office} \mid \text{minutes from})$

Pravun Goyal (IIT Kharagpur) N-gram Language Models Week 2: L...

So what is the simplifying assumptions that we make? We say we can probably use only the previous word and forget about everything else. So I have to compute the probability of office given about 15 minutes from and it is; that is approximated by probability of office given from. I forget the other terms. So or I can use the previous 2 words. So this can be written as the probability of office given minutes in from. So this is simplification that we are making. Why we are doing that? Because now it is easy to find complete the probabilities. What are the words that come of that from and how what session of times office comes after from, but this was not possible when I was taking condition on 4

previous words? Because information was probably not seeing them enough in my data, so this helps us giving a better estimate of these probabilities.

(Refer Slide Time: 11:32)



Markov Assumption

More Formally: kth order Markov Model

Chain Rule:

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i | w_1 w_2 \dots w_{i-1})$$

Using Markov Assumption: only k previous words

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

We approximate each component in the product

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-k} \dots w_{i-1})$$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Les

More formally we can use kth order Markov model. So I have this information have to compute the probability of the sequence of w_1 to w_n . By using chain rule I write it as this w_i given the previous $i-1$ words multiply for all the words. Now in kth order Markov model assumption what I will do? I conditioned it only on the previous k words. So I will write probability w_i given $w_1 w_{i-1}$ as w_i given only the previous k words. So here you are using only 1 up to $k-1$ previous words not all the $i-1$ words. So this is kth order Markov model assumption.

(Refer Slide Time: 12:35)

N-Gram Models

$P(\text{office} \mid \text{about fifteen minutes from})$

An N -gram model uses only $N - 1$ words of prior context.

- Unigram: $P(\text{office})$
- Bigram: $P(\text{office} \mid \text{from})$
- Trigram: $P(\text{office} \mid \text{minutes from})$

Markov model and Language Model

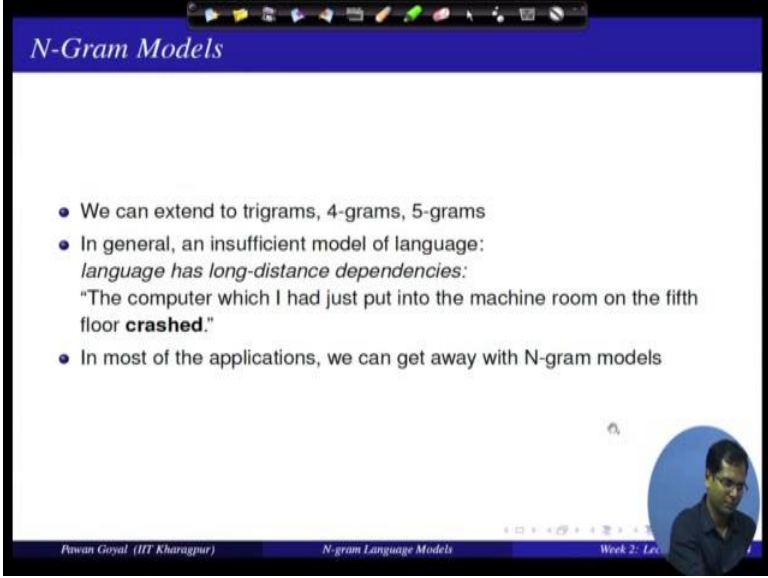
An N -gram model is an $N - 1$ -order Markov Model

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Les...

So we can do that for all the come all the components in this product. Now and that is how we will define ever N-gram models. So if I take this particular probability office given about 15 minutes from, if I am using only the previous word that will be using it 2-gram language model. So in general if I am using only n minus 1 words of prior context I am defining in N-gram language model. So here if I am using if I am not using any word from the context 0 words from the context this is a unigram language model, n is equal to 1 here. If I am using 1 word from the context, then it is a bigram language model, n is equal to 2. If I am taking 2 words from the context it is a trigram language model, n is equal to 3. Now can you try to relate in N-gram language model with some k th order Markova assumption?

In k th order Markova assumptions; you were using k th previous word. So if I am using k previous words I have a k plus 1-gram language model. So I can say an N-gram language model is an n minus oneth order Markova model. In N-gram language model I use n minus 1, words from the context. So this is the relation between an N-gram language model. And an n minus 1 order Markova model.

(Refer Slide Time: 14:06)



N-Gram Models

- We can extend to trigrams, 4-grams, 5-grams
- In general, an insufficient model of language:
language has long-distance dependencies:
"The computer which I had just put into the machine room on the fifth floor **crashed**."
- In most of the applications, we can get away with N-gram models

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lec

Now internal we can extended to trigrams we assume 2 words from the context. 4 grams 3 words from the context, of 5 grams 4 words from the context, but in general we cannot see that any N-gram will be sufficient model for the language. Why? Because language we also see some long term dependencies. So consider this sentence. The computer which I had just put into the machine room on the fifth floor crashed.

Now, what is the word on these the word crashed depends? If you were see here the word crashed depends on the about computer, but this cannot be captured by using 2 gram, 3 gram, 4 gram, 5 gram, 6 gram, so on to see. So you might have to go to 11 12 grams and that is probably not very advisable. So you use must always knew that any N-gram model is not a sufficient model of a language, but it captures many word ordering relative regularities. So in most of the applications we can use simple N-gram model with an, it will 2 3 and we will get up with it.

(Refer Slide Time: 15:19)

Estimating N-grams probabilities

Maximum Likelihood Estimate
Value that makes the observed data the "most probable"

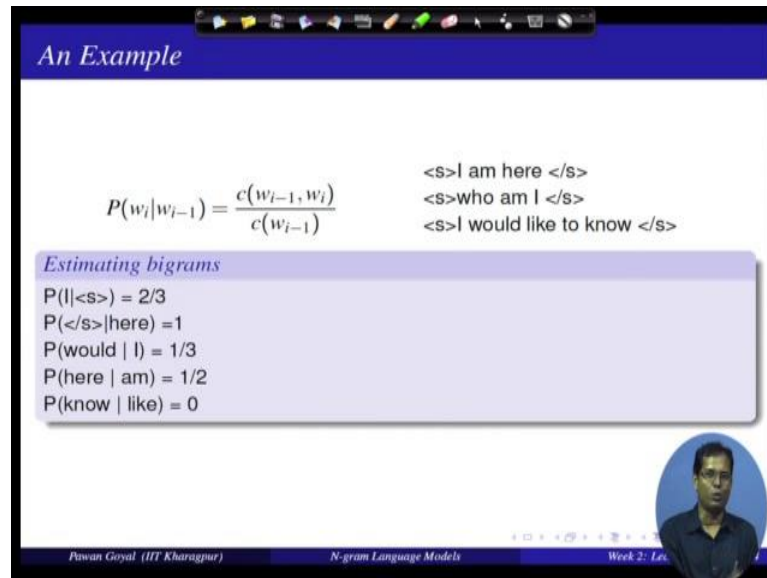
$$P(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$
$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Les...

So now the question is how do we estimate these N-gram probabilities from a corpus. So for that the simple method of estimating these probabilities is by using some maximum likelihood estimate. So what is that? Suppose I have to compute the probability of w_i given w_{i-1} . So I will find out in my corpus how many times the word w_{i-1} occurred. Among those what fraction of times w_i occurs after that. Remember we were doing it from the one of the earlier slides in versatile actress whose. So how many times the word actress comes after versatile what fraction of times? Same here what is the fraction of times that the word w_i occurs after the w_{i-1} , which defines a probability distribution. After w_{i-1} each word my vocabulary can come and this will be a probability distribution.

So in general I can give a notations of c instead of count $c(w_{i-1}, w_i)$ means number of times w_{i-1} occurred in my corpus and so on.

(Refer Slide Time: 16:31)



An Example

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Estimating bigrams

- $P(I | \langle s \rangle) = 2/3$
- $P(\langle s \rangle | \text{here}) = 1$
- $P(\text{would} | I) = 1/3$
- $P(\text{here} | \text{am}) = 1/2$
- $P(\text{know} | \text{like}) = 0$

<s>I am here </s>
<s>who am I </s>
<s>I would like to know </s>

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lecture 2

Now, let us take examples and see how do I estimate these probabilities. So I have 3 sentences in this corpus. I am here, who am I and I would like to know. And you also have some tokens on a start of the sentence and the end of the sentence. And you want to compute the bi bigram probability w_i given w_{i-1} . So how will you do that? So suppose I want compute all these probabilities. Probability of i coming at the start of the sentence; probability of here coming with this probability end of the sentence coming after here probability of would coming after again so on. So how do I compute these probabilities? So I just take the first one probability of i coming at the start of the sentence.

So I will find out how many of start of the sentences I have seen. Among those perfection of the times i occurred. So I have seen 3 start of the sentence, out of those twice highly occurred. So this probability would be 2 by 3. Here how many times I have seen here, one and among those how many times the end of the sentence occurred also one this will be one by one and so on.

So this is what I find 2 by 3 1; 1 by 3 1 by 2 and 0. So this is easy given a corpus you can find out the bi gram probabilities by it is just giving the counts.

(Refer Slide Time: 18:10)

Bigram counts from 9222 Restaurant Sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lecture 2

Now, this is some example from some restaurant corpus. So that had 9000 plus sentences and these some bigrams that were the numbers are given here. So can you see some regularities here. So i and i do not occur together much, but i and want occurred 827 times. So you are in a restaurant after I mostly say want I want some sort of food. So I want is a bigram that occurs a lot in this data. What are the others bigrams that occurs a lot, want to I want to do something? Eat Chinese, Chinese food. Eat lunch, to spend so all these bigrams that occurs a lot. So this tells in lot about that corpus. So I am talking about the restaurant corpus where it is more about ordering some food and eating some, so a kind of foods.

(Refer Slide Time: 19:09)

Computing bigram probabilities

Normalize by unigrams

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Bigram Probabilities

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lec.

Now, suppose I have to compute the probability, bigram probability. What do I need? I need the count number of times this were occurs without the word and you know divided by the unigram probability of the previous word. So suppose I give you the unigram counts also of all these words can you come with the bigram probabilities? Yes, it is becoming quite easy. So now, I just divided by the number of times i occurs. So number of times i amount of the together divided that number of times i occurs. So this gives you probability goes to 0.33. So probability of want occurring after i is 0.32 from this corpus and so on. I can compute all the words in this corpus.

(Refer Slide Time: 19:52)

Computing Sentence Probabilities

$P(<S> \text{ I want english food } </S>)$

$= P(I | <S>) \times P(\text{want} | I) \times P(\text{english} | \text{want}) \times P(\text{food} | \text{english}) \times P(</S> | \text{food})$

$= 0.000031$

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Lec.

So now given a corpus, you should be confident now that how do I compute the bigram probabilities. Now suppose you have computed bigrams probabilities from the corpus. Now can use that to solve over initial problem that is, how do I find the probability of this sentence this sequence of words. So I have the sentence here. A start of the sentence, I want English food and in end of the sentence. I want to find the probability of the sentence. So how do use bigram model do that. I will say this is probability of i given as the start of the sentence times probability want given i times probability English even want and so on, because I am actually using the chain rule of the probabilities, but I am using in first short of a Markova assumption. So that gives me all this probability and I multiply that and then it gives me the probability of this sentences each.

So if I use the previous probability that will be flip 0.000031. And now you can give me any sentence and I can use my bigram probabilities to find out the probability of the sentence. And I can solve all my problems of (Refer Time: 21:02) spelling corrections and other scenarios.

(Refer Slide Time: 21:09)

What knowledge does n-gram represent?

- $P(\text{english}|\text{want}) = .0011$
- $P(\text{chinese}|\text{want}) = .0065$
- $P(\text{to}|\text{want}) = .66$
- $P(\text{eat} | \text{to}) = .28$
- $P(\text{food} | \text{to}) = 0$
- $P(\text{want} | \text{spend}) = 0$
- $P(i | <s>) = .25$

Pawan Goyal (IIT Khairagarh) N-gram Language Models Week 2: Les

So from this corpus what does the knowledge N-gram represent? So these are some values. So what do you try to infer from there. So probability English given want is 0.0011. And probability Chinese given want is 0.0065 what does it that tell. So if you see that will tell that the any Chinese food it is 6 time more popular than the English food in whatever from whatever detail it was taken from. To give want is 0.66, so that is 2

occurs a lot with a corpus after I want. I want to do something eat after to is again 0.28. Someone want to eating to do something into eat occurs a 0.28 probability. Food given to a 0 that means, the word food never occurs after to and that is something that talks about the language.

In language generally use a verb after to, I want to do something and here the food is a noun so this will not occur in my data. Similarly, want given is spend as a 0, again some fact about language that 2 words. Generally, do not occur occurred simultaneously. And I give the start of the sentence 0.25 again gives some in instigation that most of the sentences start with i in that in that corpus. So mostly about I want to do something. So you started with i. So these N-grams might represent some knowledge about the language and grammar as such or about the particular data or domain that you are trying to build it this form. And this is some idea that we can use for even modeling different domains in my data separately. I can build different language models for each domain, will talk about this further.

(Refer Slide Time: 23:04)

Practical Issues

Everything in log space

- Avoids underflow
- Adding is faster than multiplying

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Handling zeros

Use smoothing

Pawan Goyal (IIT Khairagarh) N-gram Language Models Week 2: Les...

So there are certain practical issues that you might have to aware, of like when I am doing this probability computation for a sentence I am actually doing multiplication of many probability values. And all of these might be very small. So if I simply do multiplication of probabilities it might lead to some underflow and this might just end up in getting a 0 values for all the probabilities. So it is better that you do everything in log a

space in that way you are simply adding them and in any case adding each individual operation or more efficient operation than multiplication. So you are just storing the log of probabilities and you are adding these problem these logs. So you can if you aware to find out probabilities p_1 times p_2 times p_3 times p_4 , if you convert in log space that is nothing, but $\log p_1$ plus $\log p_2$ plus $\log p_3$ plus and so on.

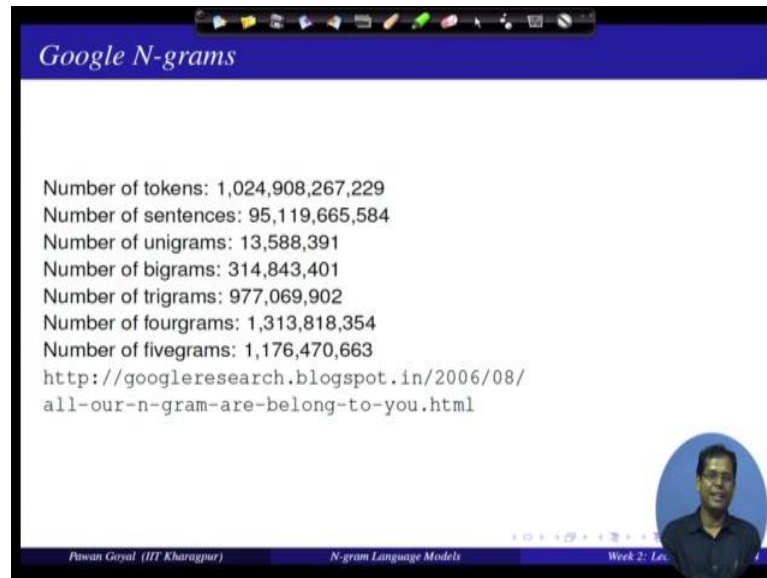
So you can restore a log values and you can just add this. There is another problem of handling zeros. Suppose in trends there is a particular bigram that you see that never occurred you a trend data. So what would happen? You will give it a value zero, but that will convert the whole, everything 0. So you need to do something for that, and we will see that in the concept of a smoothing.

(Refer Slide Time: 24:28)



So there are some popular toolkits are available. So SRILM is one popular toolkit, but there are many other toolkits that you can use for language modeling.

(Refer Slide Time: 21:40)

A presentation slide titled "Google N-grams" with a blue header. The slide lists statistics for the Google N-gram corpus: Number of tokens: 1,024,908,267,229; Number of sentences: 95,119,665,584; Number of unigrams: 13,588,391; Number of bigrams: 314,843,401; Number of trigrams: 977,069,902; Number of fourgrams: 1,313,818,354; Number of fivegrams: 1,176,470,663. It also includes a URL: <http://googleresearch.blogspot.in/2006/08/all-our-n-gram-are-belong-to-you.html>. A small circular inset photo of a man is in the bottom right. The footer contains "Pawan Goyal (IIT Kharagpur)", "N-gram Language Models", and "Week 2: Les...".

Google N-grams

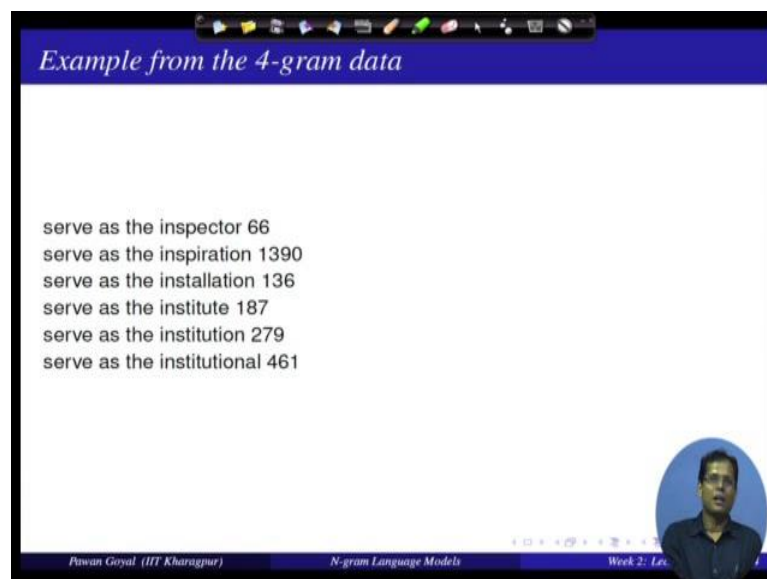
Number of tokens: 1,024,908,267,229
Number of sentences: 95,119,665,584
Number of unigrams: 13,588,391
Number of bigrams: 314,843,401
Number of trigrams: 977,069,902
Number of fourgrams: 1,313,818,354
Number of fivegrams: 1,176,470,663
<http://googleresearch.blogspot.in/2006/08/all-our-n-gram-are-belong-to-you.html>

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Les...

You can also; if you want to use some large corpus data you can try to use these google N-grams. So there again available on this link and there you will find out for each different N-gram what is the number of times they occurred in the corpus.

So if you go to this link you will find out there are huge number of tokens sentences and all. They gave you unigrams bigrams trigrams 4 grams and 5 grams.

(Refer Slide Time: 25:12)

A presentation slide titled "Example from the 4-gram data" with a blue header. The slide lists six phrases and their occurrence counts: "serve as the inspector 66", "serve as the inspiration 1390", "serve as the installation 136", "serve as the institute 187", "serve as the institution 279", and "serve as the institutional 461". A small circular inset photo of a man is in the bottom right. The footer contains "Pawan Goyal (IIT Kharagpur)", "N-gram Language Models", and "Week 2: Les...".

Example from the 4-gram data

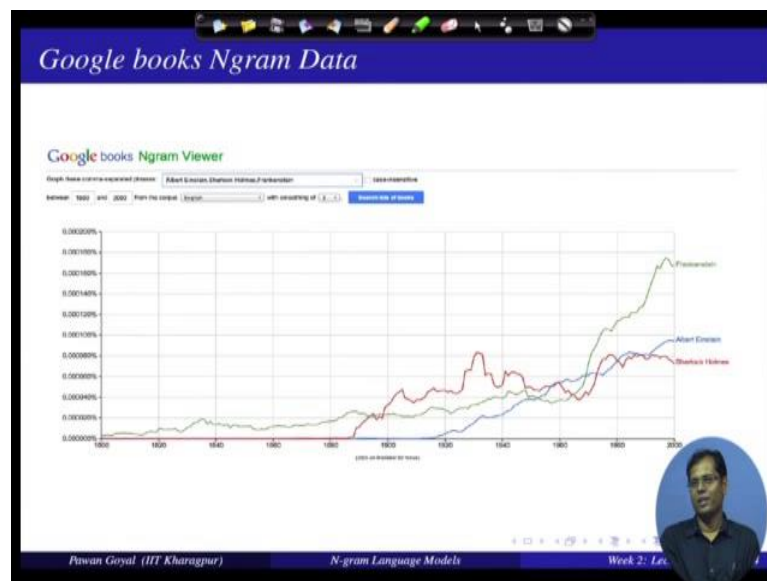
serve as the inspector 66
serve as the inspiration 1390
serve as the installation 136
serve as the institute 187
serve as the institution 279
serve as the institutional 461

Pawan Goyal (IIT Kharagpur) N-gram Language Models Week 2: Les...

So what are some of the examples? So suppose I am sing it after a starting from serve as the. So, that data will tell you serve as the inspector occurred 66 times in the corpus.

Serve as the inspiration occurs 1390 times. Serve as the installation occurred 136 times and so on. So this is a fourgram that you find from the data, but you see these are only the counts. Now how will you use that to compute the probability? That probability will be given by the formula: $P(\text{serve as the} \mid \text{inspiration}) = \frac{\text{count of "serve as the" given inspiration}}{\text{total count of "inspiration"}}$. For that you will also need to use the trigram count of serve as the; so again this 5 is available you can get this 4 gram of the data and trigram data and try to compute all the probability values.

(Refer Slide Time: 25:56)



So they also give a nice API by which you can visualize many interesting patterns in the usage of words. So because heritage divides across many different centuries you can also plot it temporally. So what we are seeing here, suppose I give 3 different queries Albert Einstein, Sherlock Holmes, and Frankenstein on Google N-gram viewer, it tells me over the years what is the probability of these bigrams. So can you see something interesting here? So Sherlock Holmes became popular around 1885 and so on. Before that is when nearly 0; Frankenstein probably there in the novels even before that. So even from the 1800s he finds the occurrence of Frankenstein that keeps on increasing even 2000, it is more than Albert Einstein and Sherlock Holmes.

Albert Einstein started getting popular in the data around 1970s. That is why most of the discoveries happened, and it came into the books and at some point of time it became more popular than even Sherlock Holmes. So you can do a nice sort of analysis of

what kind of words came into the language all my data in what times, and how the popularity changed over the years by using these kind of data.

So today we will we gave only some in intuition and what is language model, how do we compute that, and what we can use it from? For there were some problems that we saw that how do we handle the zeros. So with 0 is the whole probability of trends will go to 0. Even if one of the components has a probability of 0, so how do we avoid that problem, how do we solve that problem? That is where the concept of the smoothing will come in picture and that is what we will cover in the next lecture.