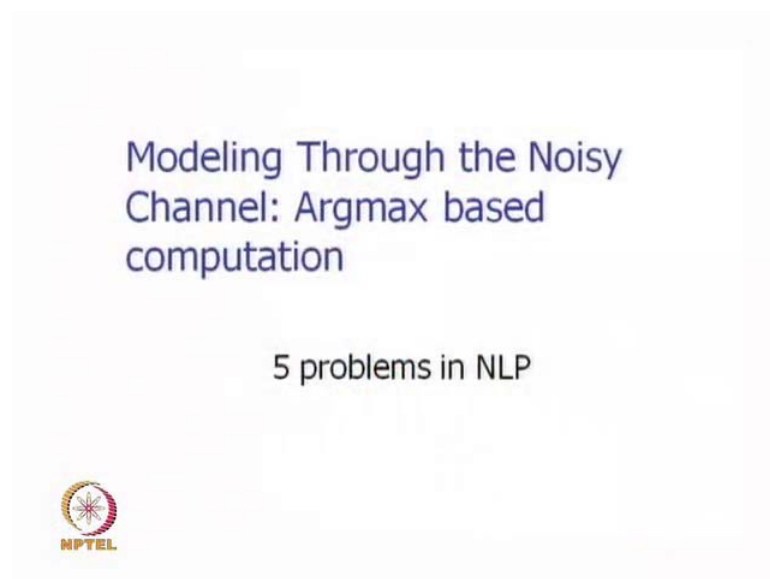


Natural Language Processing
Prof. Pushpak Bhattacharyya
Department of computer science and engineering
Indian Institution of Technology, Bombay

Lecture - 8
Noisy Channel Application to NLP

We will continue with our discussion on noisy channel application to natural language processing.

(Refer Slide Time: 00:26)



As we have said many times natural language processing problems, are modeled through the noisy channel and the foundation for this is the argmax based on computation. What we have been doing is, we are looking at some problems of natural language processing, which are amenable to this kind of argmax based computation. And we have choose 5 representative problems, with different characteristics to illustrate, these points.

(Refer Slide Time: 00:59)

Bayesian Decision Theory


- Bayes Theorem : Given the random variables A and B,

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

$P(A|B)$ Posterior probability

$P(A)$ Prior probability

$P(B|A)$ Likelihood



And at the heart of all this, is the Bayesian decision theory. Where we say that amongst multiple choices, or options the one to be chosen, will be the one, which has the highest probability and typically the probability is computed as a posterior condition probability. Where B is the conditioning variable and A is the variable of interest, many of the problems that we will discuss. Will fall into this kind of frame work and we will see that, A and B are formed a conditioned and condition appear. Now, we also saw previously that the, this conditioning condition probability is actually a many times converted into prior probability P A into likelihood P B given A, divided by P B, which is nothing but, the Bayesian theorem.

We have also, spend some time discussing, why it may be more useful to compute P A given B this way. Namely through a prior probability and the likelihood, the question that often arises, in the mind of the student is that both probabilities are condition probabilities. So, why should one we prefer over the other, this is the question of generative probability verse, discriminative probability and we will discuss these, in detail after some time. So, the question is one of modeling a problem, generatively, or discriminatively.

So, P A given B is the posterior probability. P A is the prior probability, P B given A is the likelihood and many times these prior probability acts like a filter, providing another

parameter for deciding on a on the best possible option. So, we have been illustrating this part through, a number of discussions through a number of problems.

(Refer Slide Time: 03:30)



The problems

- Part of Speech Tagging: *discussed in detail in subsequent classes*
- Statistical Spell Checking
- Automatic Speech Recognition
- Probabilistic Parsing
- Statistical Machine Translation




So, we now proceed to take the problems. Before that, the list of problems, we have been discussing are part of speech tagging, which will be discussed from the next class, in great detail. Statistical spell checking, automatic speech recognition, probabilistic parsing and statistical machine translation. So, these problems are widely diverse problems. So, what is common between these problems? One might wonder. But, we see that these problems share one particular idea, or on particular consideration, which is that all of these can be probabilistically faculty.

So, part of speech tagging: requires part of speech labels to be placed on the sentence. Statistically spell checking requires, finding the correct word. Given the wrongly spelt word, automatic speech recognition is concerned with transcribing speech. That means, from the spoken utterances obtain text document, which is machine processable. Then probabilistic parsing is concerned with producing the parts stray of a sentence. Statistical machine translation converts one language into another. So, all of them can be modeled as noisy channel problem and through argmax computation.

(Refer Slide Time: 05:00)

Problem 3: Probabilistic Speech Recognition

- **Problem Definition** : Given a sequence of speech signals, identify the words.
- **2 steps** :
 - Segmentation (Word Boundary Detection)
 - Identify the word
- **Isolated Word Recognition** :
 - Identify W given SS (speech signal)


$$\hat{W} = \arg \max_W P(W | SS)$$

So, we had spend some time previously on problem number 2, which was spell checking problem: Where we had four kinds of errors, selling errors; insertion, substitution, transposition and deletion. We now look at probabilistic speech recognition. The problem here is has defined, here given a sequence of speech signals identify the words. We have remarked before that, when a person speaks the error correctly recognizes the word boundary. Recognition of the word boundary is a very important problem and this also leads to a more basic problem that of isolated word recognition. So, when a word is utter, how can we decide, which word it is. Because, many times the words sound very similar and the word has to be decided of an from the contextual information.

So, this is an important problem, the other problem is of course, the problem of word boundary recognition. In case of word boundary recognition, what happens is that. The words are clearly separated and understood has token some information, has the speech goes on. So, if I say, I found a key on the road today. So, these sentence; contains a number of words and the error isolates words quite correctly. I found the key on the road today. However, many times these kind of boundary recognition; is not an easy task, as seen from the following example, which was discussed. When, we talked about ambiguity in natural language processing.

So, in English, if I say I got a plate very quickly. Then, there are two possible word boundaries, two possible word boundaries. So, one division could be, I got up late today.

I woke up late today. Another braking could be, I got a plate today. I got a plate today. So, this is the way the word boundary can be obtained and these sentences at this at the spoken level, could be ambiguous. With two compute a two different meaning. Similar, is the case, in case of the following in the utterance ((Refer Time: 07:58)) can be broken up in two different ways ((Refer Time: 08:03)) will come today, or ((Refer Time: 08:07)). So, we will come and therefore, this also is a case of correct word boundary recognition.

So, in spoken utterance says this is an important problem. We have also remark that to, when we understand speech. We unconsciously keep on producing disfluencies and we unconsciously, process disfluency also. So, many times the speaker takes a bit of time organizing, his or her thought and these are periods of disfluency, where meaningless utterances are produced like, I went to the school yesterday. But, so you notice those points of disfluency, where, the speaker takes some time the producing ((refer time: 09:00)) etcetera. And that is used to decide on, or organize the next segment of speech, which should reach the error.

So, these problems are basic problems, are speech recognition. The problem that, we will talk about very briefly: is the isolated word recognition. So, the problem definition as was described is given a sequence of speech signals identify the words. There are 2 step, segmentation namely word boundary detection and then, identifying the word. We concentrate on isolated word recognition and this is very well know in the speech community, as a basic problem. The problem is to identify the W the word, given S , the speech signal. So, these can be formulated as an argmax computation in the following way. What is the best possible word, given the speech signal and W is the condition variable here: Best possible in their sense of being the most probable word given this speech signal.

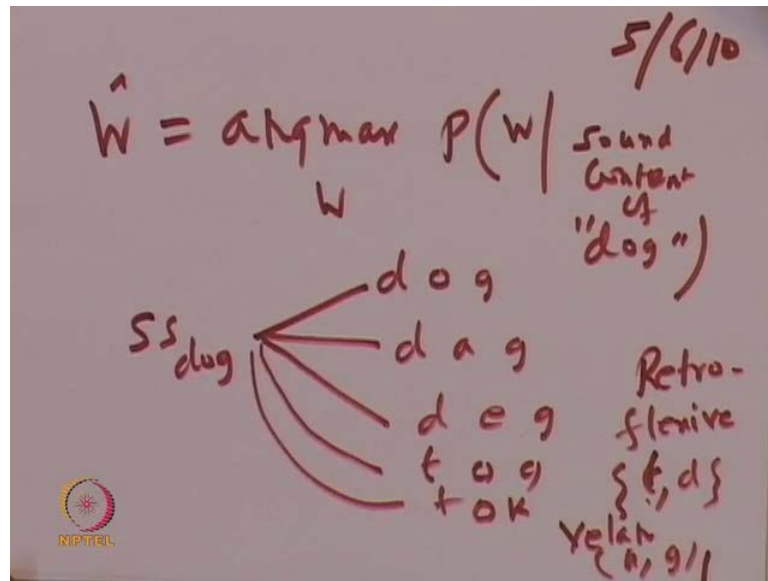
So, we compute $P(W)$ given S and take an argmax over all possible W , which could have come from this speech signal, ok. So, this becomes an argmax based computation and the word which has the highest probability, is chosen as the answer. Now, one could directly compute, these probability; from a training corpus. The way the training corpus is created for speech recognition situations is that; peoples speech, are recorded. When somebody produces a dialogue or you selected, or speaks over phone. These spoken utterances sequences of speech signals, are recorded. Then, what is done is that on these

speech utterances, one does the speech engineer does reading. Then, marking the boundaries and then, producing annotations of various kinds. For example, one could annotate the speech signals, corresponding towards by the parts of speech, ok.

So, where we spoken utterance like, I went to the bank. Here, I is a pronoun, went is a verb, to is a proposition, the, is a detement and bank is a noun. So, those kind of annotations are produces on the words. Even word boundaries are also, some kind of an annotation, on the spoken a utterance record. The record the speech, we produce these kind of word boundaries, part of speech emphasis and so on, ok. So, this kind of information is important for the machine to process and this is called annotation. Now, when we produce these kind of annotation, it is possible to train a machine with these speech data and the parameter, which is of interest here. P W, given S S that can be computed from the record data speech, ok.

So, very a simple approach to these would be, if the speech signal corresponds to that of dog. Where you have, Da sound Aa, sound and ((Refer Time: 13:02)) sound, ok dog. Then, d o g, dog is the spelling, which has to be the output. So, the speech signal is the speech signal, corresponding to the utterance dog with the sound of Da, Aa and ((Refer Time: 13:17)) and these produces the output d o g. Now you could imagine that, these particular signal corresponding to dog Da, Aa ((Refer Time: 13:34)) these can also be tagged and this can also be process then, the output has d a g, or let say even d e g, ok. All these possibilities exist.

(Refer Slide Time: 13:58)



So, what I mean is that, I will write it for clarity. What, I mean is that the word, which corresponds to the speech signal of dog. This is the sound content of the word dog, ok. The sound content of the word dog: So, one could see that, depending on how it is produced? and the way the speaker produces the sound, it can give rise to S S each is speech signal of dog, can give rise to d o g, or d a g, or d e g, or even things like t o g, t o k and so on, ok. You understand this particular point? The point, I making is that, it is possible to here, the sound pattern of dog. When somebody utters it, is possible to hear all these strings. So, the sound pattern of dog can correspond to d o g. It can correspond to d a g dag. It somebody says dag for example, and the output could be a, it could be hard as d e g confused as d e g.

And we know that, consonants t and d, they belong to what is called the Retro flexive group, Retro flexive sounds, ok. Retro flexive group's t and d belong to this group. So, it is possible to confuse these two sounds t and d and therefore, the machine could produce t o g. Similarly, there is a group the velar group, velar; which is k and g. So, k and g also belong to what is called the velar group or the ((Refer Time: 16:28)) sound comes from the from deep below, the throw and these two can be again confused. And therefore, it is possible for the machine to produce t o g and t o k. Now you will surely comeback and say that, all the strings are actually meaningless except for dog and therefore, this should be the winner. But, the point is that, it is not a human being who is doing this job, of

isolated word recognition. But, a machine is doing it, which a machine which has been trained, on people's recorded speech.

And we know there is an enormous amount of variation on the way, people speak. Particularly, the vowels, especially the vowels have amount of variation from language to language, speaker to speaker, dialect to dialect. Even the same English language speaker coming from different regions of the world, would produce the vowel sounds very differently. For example, the o sound is very prominent in ((Refer Time: 17:38)) bring all and therefore, it could become d, ok. Where the o sound is extremely prominent and it is possible to pronounce dog has dag, where the o sound borders on ((Refer Time: 17:50)) and a therefore, the machine may be confused, to be leave that it was it harden a ((Refer Time: 17:58)) sound and it produces d a g, ok.

So, this is the way a speech signal can be processed and converted into a sequence of characters. And all these are not actually a fictitious discussions. These are actual realities. If we speak to the error, as to what they heard: Especially noisy environment, a mixed a lot of other features going on. A lot of talking going on all around, it is possible for a machine to be confused and produce any of these strings, ok. The machine may not be robust to all possible noise, ok. So, one issue is noise, the other issue is the real linguistic problem, of confusing consonants. When they belong to the same group and also confusing vowels, because, there is an enormous variation in which now, comes the role of probability.


As you correctly said, that dog should have the highest probability, because, this is the string d o g is the string, which occurs in the language. Much more often, than, strings like d a g, d e g as individual words and this is the point. That will be the point is that, amongst all these strings which are, possible as output from the machine. One of them as highest probability and through this argmax computation ok, catch that string. Catch that particular string, which has the highest probability and this probability is computed by means of application of Bayesian theorem. And calculating the parameters learned from the corpora. These are the basic ideas. So, I hope the point has been properly made and we proceed with the discussion.

(Refer Slide Time: 20:06)

Speech recognition: Identifying the word

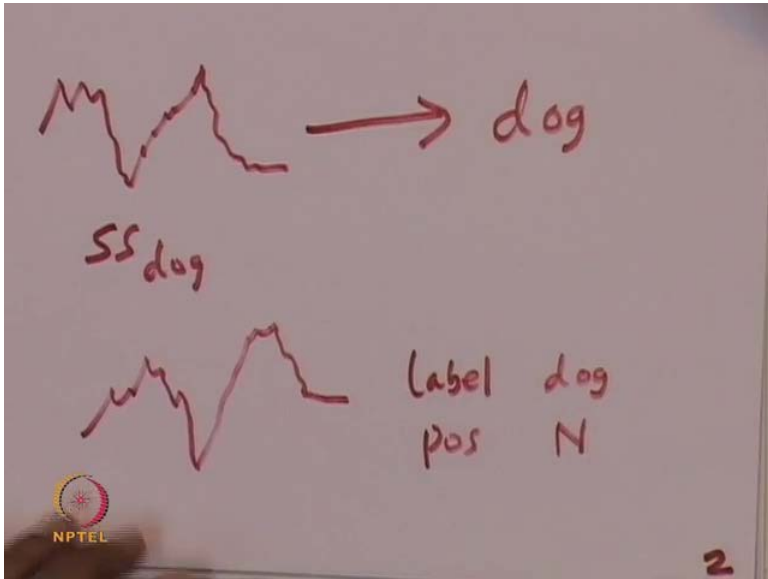
$$\hat{W} = \arg \max_W P(W | SS)$$
$$= \arg \max_W P(W)P(SS | W)$$

- $P(SS|W)$ = likelihood called "phonological model"
→ intuitively more tractable!
- $P(W)$ = prior probability called "language model"

$$P(W) = \frac{\# W \text{ appears in the corpus}}{\# \text{ words in the corpus}}$$



So, we have taken this expression of $\arg \max P(W | SS)$ over all possible W . W hat is the output string and this, we see has been converted into $P(W)P(SS | W)$. This is the likelihood. $P(W)$ is the prior probability. Who were all possible W . Now, as before we asked the question as to was it necessary to apply Bayesian theorem here. Could we not work with these probability, $P(W | SS)$. And let us see, how we obtain the. So, the recorded utterances have been transcribed. I will show you in a some kind of imagined situation. But, which illustrate the points well.

(Refer Slide Time: 21:17)



SS dog → dog

label dog
pos N



2

So, this is let us say the speech signal corresponding to dog. This is the S S dog. The speech signal of dog and the then, this corresponds to dog. So, what happens is that, we have the signal, as shown there and along with this, there is a label, which is dog and the part of speech of this is now. So, this is what is done, we have the signal, the label dog and part of speech now. There can be many other information plugged on to it, but, this is called transcribing and annotating the spoken data, ok. So, now it is should be cleared to you, as to what should be done with this data. A machine will see this kind of signals, ok and will also see the label on this. This is dog, ok. So, I am blasting about some details, which will come in a minute.

So, just given to this idea, as to what is going on? The speech signal and correspondence d o g dog, this is observed by the machine, ok. Amongst many other possible signals and there levels, the machine keeps note of, the signal it is saw and the level it saw, along sides. So, it saw the signal, speech signal of dog. It saw the level dog and increase the count. That this signal corresponds to, the dog, ok.

(Refer Slide Time: 23:16)

$$P(\text{dog} | \text{speech signal}) = \frac{\#(\text{dog}, \text{speech signal})}{\# \text{speech signal}}$$

data sparsity decoding
"ProGradination"
NIPITEL

Now, what is should be done is that? To compute the probability: of dog. Given; this speech signal, ok. We simply count, how many times the signal came? And how many times was it level as dog? Ok. So, we have the signal and the level dog on it. So, how many times the signal come? And how many times was it level as dog? Now the question that, might come to a mind is after all if, I get this signal, is not it same as do.

What is the point of calculating this probably, this going to be 1; because, every time the signals seen, it is associated with the label dog, ok. So, that is an important point and it brings in other things like data sparsity, ok. Data sparsity decoding, which will discuss as we described various statistical natural language process problems. The only thing I could mention to now is that, if the dog is a small string, ok.

Now if the word is long. For example, procrastination, consider the word procrastination, ok. So, these a long word and it will has to have a long signal and it is also possible that, this signal has such does not appear in the spoken corpora, ok. So, what you have is smaller signals, smallest speech signals. May be you have a signal corresponding to pro, may be you have a signal corresponding to shunt, ok. And it should be possible to construct the string procrastination, from those isolated parts and the probability values, ok. So, this is the question issue of decoding and the fact that, the signal corresponding to procrastination may not appear, in the training corpora. This is the, issue of the data sparsity.

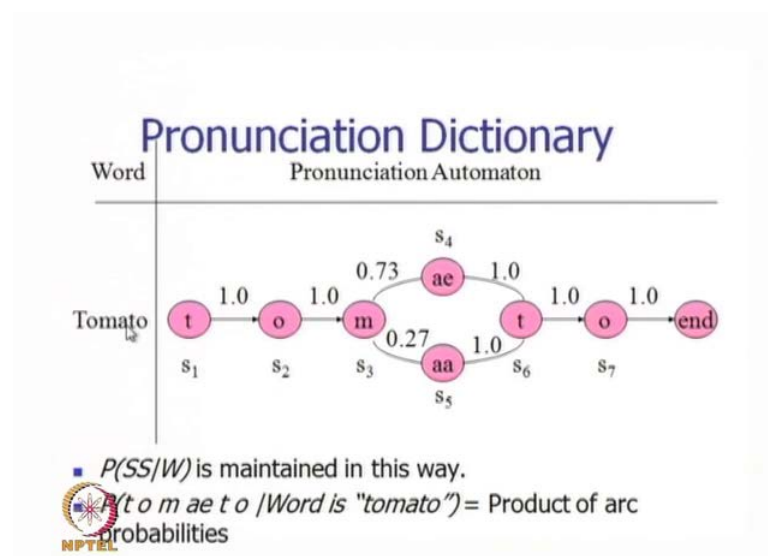
So, it should be possible for ask to construct, the string from the speech signals of smaller strings, ok. So, now it should be possible for you to appreciate. Why the Bayesian theorem application, could be a wide step. So, when we apply these process, of Bayesian theorem and convert the probability into posterior probability into prior probability into likelihood. What we are doing is that? We are taking advantage of a filter, in the form of $P(W)$. So, $P(W)$ is the probability of the word, prior probability of the word. So, given the speech signal of dog, we know that, it is possible to get a dog, dag, deg, tog, tag so on. However, form since dog is a valued word of the language and other strings are not valued. Therefore, $P(\text{dog})$ will be much higher than, others.

And therefore, it will have a higher chance, of veening, in terms of probability. The other point of course, is that. These probability seems to be easier to find from recorded corpora and it is also possible to do this calculation more reliably, compared to $S(S)$ given W . Because, speech signal; the device that catches the signal, is an electronic device, subject to error noise and so on. However, W is a return string of alphabet and return string of alphabet is likely to be more reliably produced. And the speech signal corresponding to the word, may be found more reliably from the recorded data. But, I request you to point over this question, as to reach probability is easier to compute and

which probability is more reliably found. From the corpora, from the recorded speech data, ok, this is the point, you should think about.

Now there are some technical terms, well established and well accepted for this probability values. So, $P(S|S)$ given W , is the likelihood. This is also call the a phonological model and this we remark here, is intuitively more tractable. Then, the probability $P(W)$ given S . $P(W)$ is the prior probability and it is also called the language model, to be more correct in this particular situation, we should call it the word model. That means, it these models the probability, or the phenomenon, or formation of a word, from the constituents alphabets. So, $d o g$ is a valued string in English, $d e g$ is not valued. It is unlikely that, this particular string will appear in the corpus. So, $P(W)$ is a computer has number of times W appears in the corpus, divided by number of words in the corpus

(Refer Slide Time: 29:14)



Proceeding further, how is the $P(S|S)$ given W computed? We just gave a very rough idea, some time back. When we wrote, the signal for dog, on a piece of paper and the string $d o g$. actually, what is done? Is that a very rich structure, a rich resource call pronunciation dictionary is created and maintained by the automatic speech recognition system. So, these shows a pronunciation automaton, for the word Tomato. Now, Tomato is the string $t o m a t o$. And the word typically is pronounced in two different ways. It could be tomato, or tomato. The a oval here, it can be pronounced in two different ways, as Aa, or

long oval AA, or ae. Which, is a combination of two oval sounds. So, the path corresponding to the pronunciations also are shown here, by means of a finite state machine, such that, the arcs so, have probability values.

So, now let us follow these finite state machine. T is the 1st letter and the machine is in state S 1, from t it goes to o, with probability 1.0 imagine here that, the whole spoken corpus has only the recording of these particular string namely, tomato. Now from o it goes to m with probability 1.0. Now there are two possibilities, one is so, one could pronounce tomato, in which case this path will be taken. It is a oval combination of a and e ok and in the other case it is the long oval case r and from m one could go to m m ae, or m aa. Thus given rise to Tomato, or Tomato or Tomato and each path, probability can be computed by multiplying the probability shown on the arc. We see here that, the pronunciation Tomato is more likely than, the pronunciation Tomato.

Because, the we here, the we have the probability 0.73. The probability here is 0.27. This simply means that, when we recorded the speech of people, uttering Tomato, or Tomato we found that, people mostly at a Tomato, compared to Tomato. That is why these probability, is higher. So, we will stop this discussion on automatic isolated word recognition. Because, this is a different topic all together, which is covered in speech quiet extensively, our goal here is to understand the argmax based computation. And all these, rescriptions have should have, will able to convenience you that, this way of finding the isolated word from the speech signal. In the form of an argmax computation is a well effective idea.


And this particular probability expression $P(W \text{ given } S)$, is converted into likelihood into prior probability, that also good idea, because, from the pronunciation dictionary we can compute $P(S \text{ given } W)$ a likelihood, $P(S \text{ given } W)$. And the prior probability can be computed from a corpora, which is simply the corpus of English language documents, ok. So, in English language documents, we find out how many times thus, d o g appear and these occurrence will fall out way. The number of occurrence is likely see t o k, or d e g, etcetera. Those are very unlikelihood sequences in English language corpora and we observe this kind of experience, into the probability expression, in the form of prior probability. So I have this point is clear.

(Refer Slide Time: 33:59)

Problem 4: Statistical Machine Translation

Source language sentences → *Noisy Channel* → Target language sentences

- What sentence in the target language will maximise the probability
 $P(\text{target sentence} / \text{source sentence})$



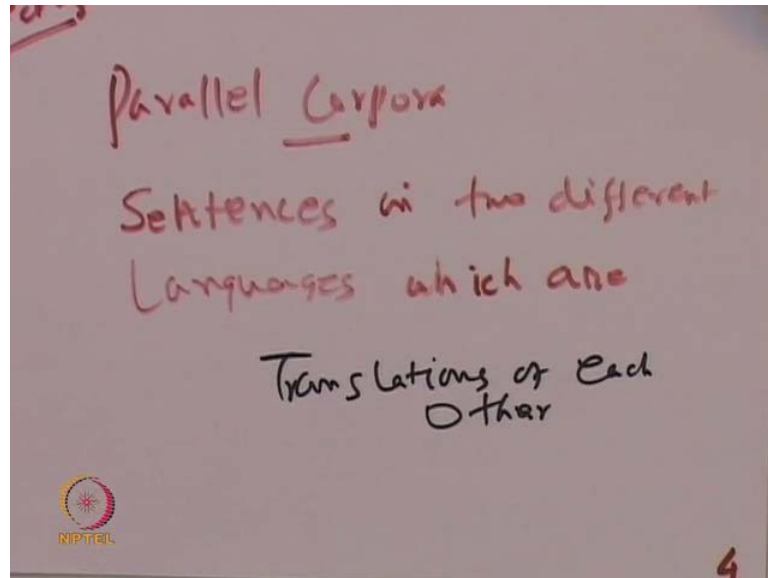
We move on to the problem of statistical machine translation, which is an immensely important problem today in the world, where the electronic data on the internet, is multilingual: in nature. There are multiple languages in the world; of web also is no exception. Languages are making their presence felt in the internet, by producing their content and the since, this is the a reality of current days world. It is considerable that, the statistical data base techniques, machine learning based techniques, which a training machine, from the electronic corpora, can be constructed.

So, this is the problem statistical machine translation. By the way the statistical approach to machine translation is relatively new. Translating from one language to another language, has been an extremely important problem, very old problem. In fact the old field of natural language processing is say to abstracted, when in the cold war there was this an important requirement of translating from English to Russian and Russian to English. And there was quite an amount of money spent on this particular problem. And there the, approach was to look up dictionaries and then create intricate rules. For transforming structures from one language to another and this, approach as remained. And produced very successful systems like, Sistrion, which is an effective system for translating between European languages and English.

Now, the statistical machine translation approach became popular in last let say, about 10 to 15 years. Where lot of electronic data, in different languages became available and an

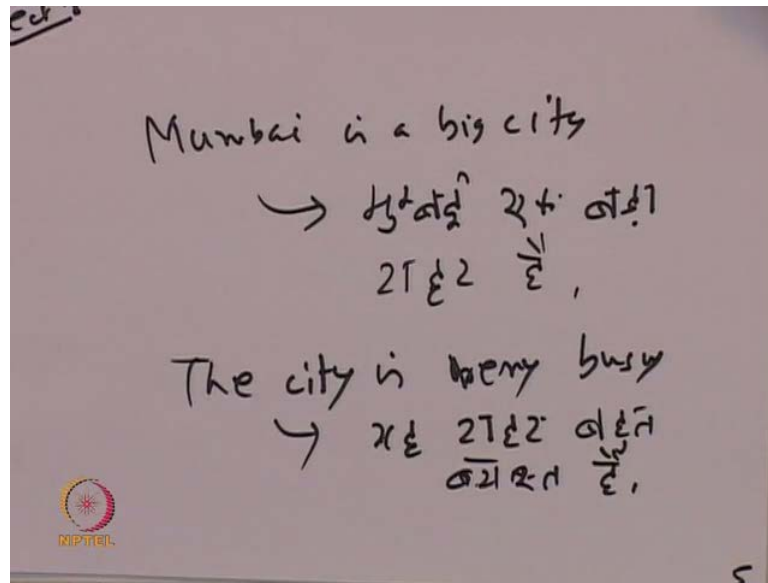
important resource which is needed in statistical machine translation, is what is called a parallel corpora. Let me give an example to make your understanding clear, on this point.

(Refer Slide Time: 36:25)



So, parallel corpora, just, I will take two minutes to annotate the slides. So, parallel corpora means, sentences, in two different languages, two different languages, which are, translation of with each other. So, essentially create, a very large resource based of sentences, in two different languages, the source language and the target language. Between which, the translation is required and these are kept aligned with one and other. That is why it is called a parallel corpora, to give an example.

(Refer Slide Time: 38:15)



Suppose, you say that, Mumbai is a big city. And that, the corresponding Hindi translation is ((Refer Time: 38:37)) ok. The city is very busy. This is the another sentence, the city is very busy. The corresponding translation is ((Refer Time: 39:09)) ok. So, this is an example of parallel corpora, of course, an extremely small corpora. Where we have English sentence and the corresponding, in this sentences: Mumbai is a big city, Mumbai ((Refer Time: 38:41)). The city is very busy. ((Refer Time: 39:44)). Now, when this kind of sentence have been aligned, ok, they have been placed in this kind of corresponds to each other. We obtained, what is the parallel corpora? And the key idea is that, a machine is shown this kind of parallel translations and it is trained from this parallel translations. The translation patterns and the units:

So, the machine alts that, m u m b a I, in English corresponds to ((Refer Time: 40:26)) the a city correspond to ((Refer Time: 40:31)) is corresponds to ((Refer Time: 40:34)) a correspond to ((Refer Time: 40:35)) and big correspond to ((Refer Time: 40:38)). So, you can see that the in English, for this particular sentence. There are five words Mumbai is a big city. Hindi has also five words. Mumbai ((Refer Time: 40:51)). But, the word order has changed, ok. The word city was the last word in this sentence. Whereas, this is the second last word in the Hindi sentence. Also see the movement of the word is, it is

optional word and from being a second word, in the source sentence. It has become the last word, in the target sentence.


So, these way word change the position, the many times. Words get dropped in the translation, or they get completed. That means a particular word can give rise to more than one word, ok. So, this is the problem of learning the patterns. When, the sentences are placed in closed correspondence. So, the city is very busy, this a five word sentence ((Refer Time: 41:55)), this is also a five word sentence. And one can established one to one correspondence, I will bit with the recognition of fact that, the words can occupy different positions. In source the sentence and the targeted language sentences. So, the machine creates a mapping of words, or double words, or triple words, technically called unigram, bigram, trigram established correspondences.

And these, correspondences I then used on a new sentence to produce the new target language output, ok. So, here again you can see there is a process of breaking apart the units. Learning the correspondences and using them conveniently, when a new sentence is given. This a basic idea and it again be formulated as a noisy channel argmax based computation. So, the question being asked is; what sentence in the target will maximize the probability P target sentence given the source sentence? The picture shown in the source the situation, source language sentence, goes to the noisy channel and becomes a target language sentence. These a metaphor, which is found convenient.

(Refer Slide Time: 43:30)

Statistical MT: Parallel Texts

- Parallel texts
 - Instruction manuals
 - Hong Kong legislation
 - Macao legislation
 - Canadian parliament Hansards
 - United nation reports
 - Official journal of the European Communities
 - Trilingual documents in Indian states



Now, we have discussed to an extended the importance of parallel states, where the translatable units, translation units and the mapping these are not. When, does the parallel states is from? Typically, the source is the instruction manuals, for English Chinese translation, which is an important problem in today is world, from many commercial reasons. With the Hong Kong legislations documents are used. Because: these documents are available in two languages, in English and Chinese. Macao legislation is simply a is again similarly, for transition between English and another language. Canadian parliament Hansards, is the parallel corpora used for English to transmission translation, which by the way, is of very high quality. United nation reports are available in six official languages of U N, namely: English, French, Spanish, Chinese, Arabic. United reports are available in these languages, the form parallel corpora.

Similarly, there is something called, euro park corpora, which is not mentioned here. Very famous, euro park corpora, is available in languages of Europe and they are in very good sentence aligned for. And they can be used for training a machine. In India, documents are typically available in three languages. Because: every state implements the three language formula. English, Hindi and the language of the state and therefore, that is a good source of parallel corpora.

(Refer Slide Time: 45:23)



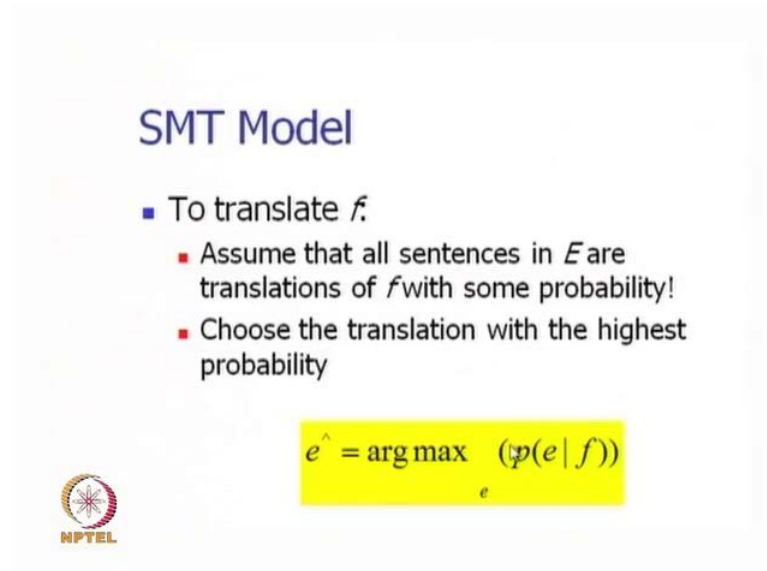
SMT: formalism

- Source language: F
- Target language: E
- Source language sentence: f
- Target language sentence: e
- Source language word: w^f
- Target language word: w^e



The statistical machine transition formalism would be as follows. Source language called F , the target language E , source language sentence are f , target language sentence is e . Source language word is W upper suffix f , target language word W upper suffix e .

(Refer Slide Time: 45:44)



The slide titled "SMT Model" contains the following text:

SMT Model

- To translate f :
 - Assume that all sentences in E are translations of f with some probability!
 - Choose the translation with the highest probability

$$\hat{e} = \arg \max_e (p(e|f))$$

The NPTEL logo is visible in the bottom left corner of the slide.

Then, to translate f , we assume that all sentences in e are translation of f with some probability ok. So, that is a interesting position. Because, we have the scoring function, in terms of probability any way and therefore, we can assumed, that the every sentence in E are candidates for transmission of f . And finally, we choose a transmission with the highest probability, which immediately makes the situation amenable to and arc argmax based computation. Where e hat is equal to argmax, over e of the expression $p e$ given f , ok. So, f is the source sentence, e is the target sentence.


(Refer Slide Time: 46:29)

SMT: Apply Bayes Rule

$$e^{\wedge} = \arg \max_e (p(e).p(f | e))$$

P(e) is called the **language model** and stands for **fluency**
and
P(f|e) is called the **translation model** and stands for **faithfulness**

Both these are computed by breaking them down into smaller components of n-grams





Now, we apply base theorem here and the reason we have bayes theorem means, quite clear in this particular case. Because, of you want to make use of the prior probability $p(e)$, is called the language model. More appropriately possible it should be called the a sentence model. Because, e is the sentence and the value of e , or a probability of e goes, if it is a valued sentence of the language. Because, this occurs much more frequently in the corpora, then ungrammatical wrong sentences, $p(f|e)$ is the modeling of the translation efficacy from e to f . This is called the transition model and this transfer faithfulness, which means, how much of the content of e is transfer to f ? Ok.

The better the meaning correspondence, where will this probability and $p(e)$ is the modeling of the fluency situation, ok. If a sentence is influent in the language grammatically correct, idiomatically correct, $p(e)$ value will go up and f if f has strong transition corresponding to with e . Then, this value will go of therefore, f is very faithful to e f is a faithful rendering of a e . So, both the result of the computed by breaking them to the smaller components of, what are called a n grams a language models, which will be a topic of discussion for oscillator.

(Refer Slide Time: 48:08)

Problem 5: Parsing

Source sentence  Target parse

$$\begin{aligned} T^* &= \underset{T}{\operatorname{argmax}} [P(T|S)] \\ &= \underset{T}{\operatorname{argmax}} [P(T) \cdot P(S|T)] \\ &= \underset{T}{\operatorname{argmax}} [P(T)], \text{ since given the parse the sentence is completely determined and } P(S|T)=1 \end{aligned}$$


Now, we come to a very very important problem of natural language processing, a fundamental problems, as natural language processing called parsing. Those of you who have done a codes in compilers, would do know, what parsing is? A fragment of program ok, a c program, photon program, passes through the parsing stage, for the correctness in structure and this parsing produces what is called the part stream. Here also we can think of the problem is as the a problem of parsing, the source sentence through the noisy channel. And obtaining the target parse, through arc argmax based computation. These argmax based computation is expressed through this equations here.

T star is the target parse and this is obtained by argmaxing over, T for the probability values of P T given S. So, S is the sentence T is the tree. So, P T given S is the conditioned probability and for the different values of T it has different values. So, choose that T which as the best possible probability. Now in this case again we apply a bayes theorem and we converted into P T the prior probability, into P S given T, that likelihood.

Now here the application of bayes theorem is eminently suitable, for a very interesting fact. P S given T, what is this? Is the probability of the sentence S given the tree T. Now, one could see that, if the tree is known. Then there is no reason any uncertainty about the sentence, because, the one simply teachers together the words. At the leaf levels and produces the sentence. So, P has given T is 1. Therefore, T star is equal to static argmax

P T. So, choose the that, particular tree, a parse that, the probability. We discuss this continue the, this discussion, in the next lecture.