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

Lecture - 52
Optimization Based Approaches for Summarization

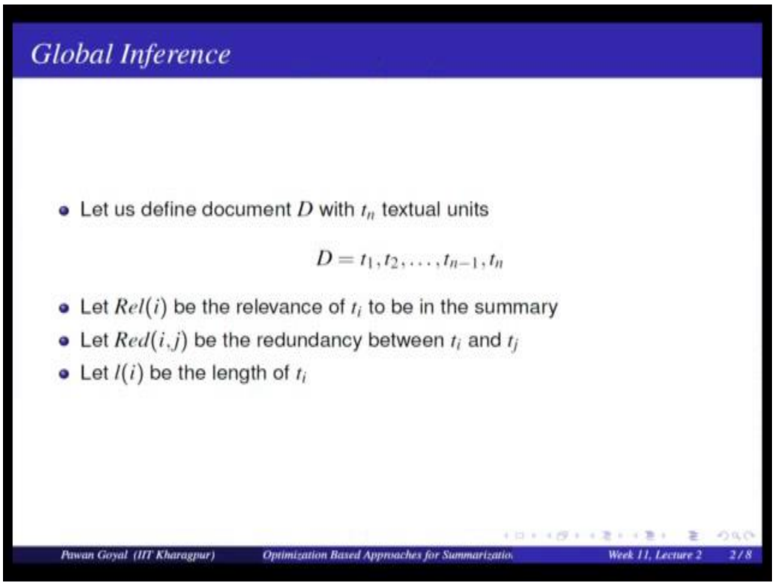
Welcome back for the lecture 2 of this week. So we had started our discussions on text summarization, and in the last lecture we discussed one of the very important approaches for text summarization that is based on a graph based method, that is called also lexrank or page rank method for summarization, what was the idea there? So we were taking all the different sentences in the document and then finding out the similarity between them and trying to choose and then once we run the page ring algorithm over this graph, we take the sentences that have the high page rank.

So idea is that my document corresponds to one particular theme and the sentences that are important to this document should have many other sentences similar to that. And that was the idea that was captured by using the page rank formulation. So the sentences should have high incoming edges from other important sentences.

And then we said this will give you the important sentences, but further if you want to find out sentences that are diverse enough with respect to each other then you will use some sort of mmr maximum marginal relevance algorithm. So you will say that sentence if it is already similar to some sentence in the summary I will reduce it is overall score. And then I will sort the remaining sentences again and choose the sentences that are coming out to be having highest score. So lexrank you can always choose as one of the very simple and effective base length for any of the summarization tasks. Although that it does not give you much flexibility in having you decide what is your criteria for importance. So how do you want to choose your important sentences?

So in this lecture today what we will see that there are some approaches that that can help you that can give you a generic framework for optimization. And then it is up to you to define your own optimization function your own constraints and you can keep on adding your constraints depending your on your task. So you will probably discuss about 1 or 2 scenarios that how do you modify it for different tasks. So in this lecture today. So we are talking about the optimization based approaches for summarization.

(Refer Slide Time: 02:30)



Global Inference

- Let us define document D with t_n textual units

$$D = t_1, t_2, \ldots, t_{n-1}, t_n$$

- Let $Rel(i)$ be the relevance of t_i to be in the summary
- Let $Red(i,j)$ be the redundancy between t_i and t_j
- Let $l(i)$ be the length of t_i

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarizatio Week 11, Lecture 2 2 / 8

So for doing that, let us define it formally define a document as containing some different textual units and because we are talking about extractive summarization. So our textual units will be sentences. So I have some n number of sentences. So I am denoting them by t 1 to t n. Now let us say I can define what is the relevance of a textual unit t i to be in the summary by relevance of i rel i. Now this is a generic formulation. So I am saying I will define the relevance of each unit by rel i, but how do I compute rel i, can you are free to choose your own method.

So what are the different methods of computing relevance that we have discussed? One is simply by taking the weights or topic signatures in the sentence and adding or and doing the average over all the weights in the sentence. That is one particular method of choose choosing the relevance of ith textual unit. Then you can also run your lexrank algorithm and find out the relevance as the page rank value of that sentence, but suppose you want to use some different criteria for relevance. So you are free to do that. So your relevance criteria might be how many named entities are involved in this sentence. That can be your criteria then you can accordingly choose your define your relevance.

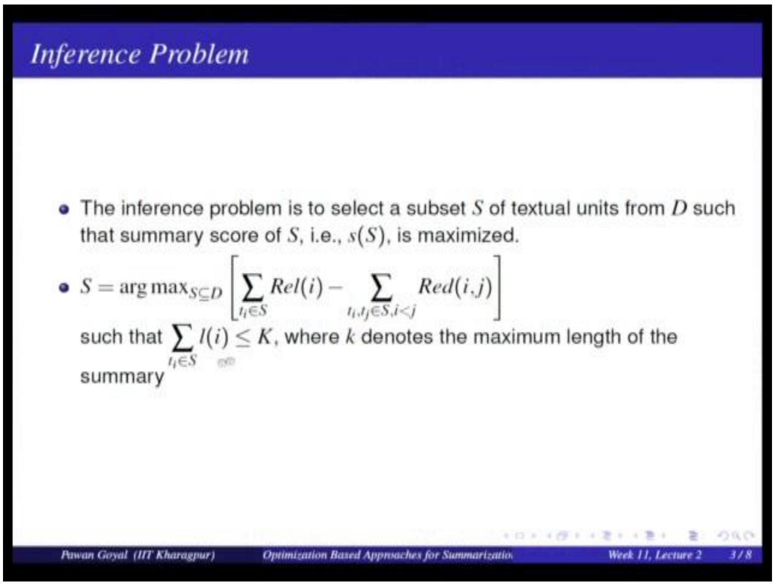
So you can define your relevance then you have to also define what is the redundancy between n into textual units. So that is how much they are redundant to each other. And why we are doing that? Because in the final summary that we obtain we do not want multiple sentences that are conveying the same information, remember in summarization the idea is that I want to get as much information as possible within the minimum possible space. So I always have a space crunch. So I want to fit as many information as possible. So I want to remove redundant sentences. So that is another function that you have to obtain.

Now the redundancy between 2 textual units again you can define it by simply taking the cosine similarity of them, how similar they are or you can also define it by some other methods like how much apart they are in the document are the very near are they very far apart. You can also depend on many different criteria like do they talk about the same sort of entities and many other things. So once you have done that, and let us say we define what is the length of the textual unit that is simply the number of words that are there in that textual unit.

Now, once we have all these definitions what will be my optimization criteria? So for choosing the summary, when I talk of summary, so in all these systems I will generally have an upper limit on the summary. So I will say I want a summary of length k and k will be roughly 100 or 200 at most depending on your input document. So you want a summary of length k. So what should be some optimization criteria?

Now, optimization criteria should be from all these t n units select a subset such that the score of the summary is maximized. And how do you define the score of the summary? The summary will have some set of textual units. So score could be summation over the relevance scores of all these units that you have taken in the summary minus what is the redundancy between any 2 units. So that will be all paired redundancy score. So relevancy score of all the units in the summary minus redundancy score of all the pairs in the summary, that will be your optimization function and you want to select the subset such that it maximizes this score.

(Refer Slide Time: 06:16)



Inference Problem

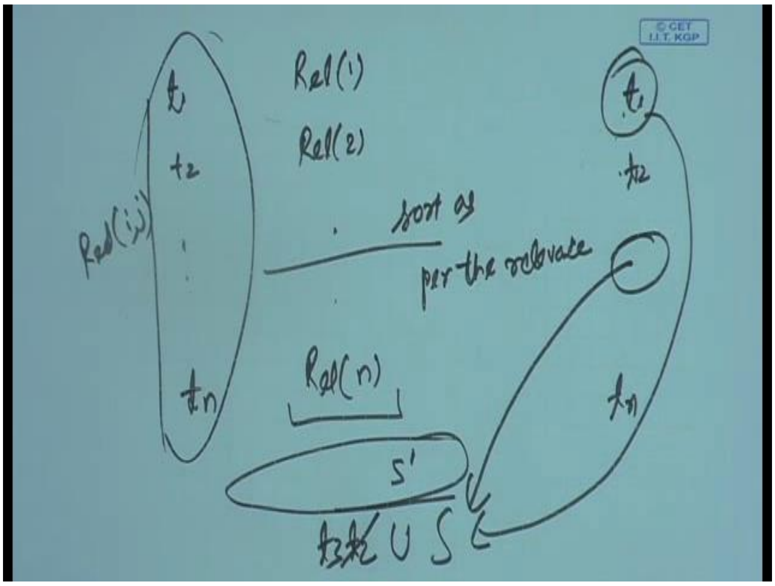
- The inference problem is to select a subset S of textual units from D such that summary score of S, i.e., $s(S)$, is maximized.
- $S = \arg\max_{S\subseteq D}\left[\sum_{t_i\in S} Rel(i) - \sum_{t_i,t_j\in S,i<j} Red(i,j)\right]$ such that $\sum_{t_i\in S} l(i) \leq K$, where k denotes the maximum length of the summary

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarizatio Week 11, Lecture 2 3 / 8

So I can say that what is my inference problem. I want to select a subset as of textual units from d. So d is my whole document containing t 1 to t n. Such that summary score of s we can define it as small s of the set capital S is maximized. And how do I define the summary score of the subset, as I said I will take all the textual units there get their relevancy scores sum over that, summation over relevance of i for all the units ti in s minus summation over redundancy score of any 2 units, i and j that I am taking in the summary and that I will sum over all the pairs, but I do not want to sum over I do not want to repeat a pair. So I will put some ordering say I have to be less than j, and they can be in the order in which they occur in the document. So I have this t i, t j in s i is less than j and I am getting the redundancy score. So what is the objective function trying to achieve maximum relevance within the set with the minimum redundancy? So total score is defined like that.

Now, and then what is the constraint? For doing that, the length of all the textual units that you are doing the summary l i in all the t i in s should be less than equal to k, that is your constraint and so capital k denotes the maximum length of the summary. So you have a constraint and you have an optimization function. Now question is that how do you achieve this optimization, how do you maximize this particular objective function. Now in the case of MMR we were using a sort of greedy approach for doing that. So a greedy approach is always a solution. So how would you use a greedy approach?

(Refer Slide Time: 08:13)



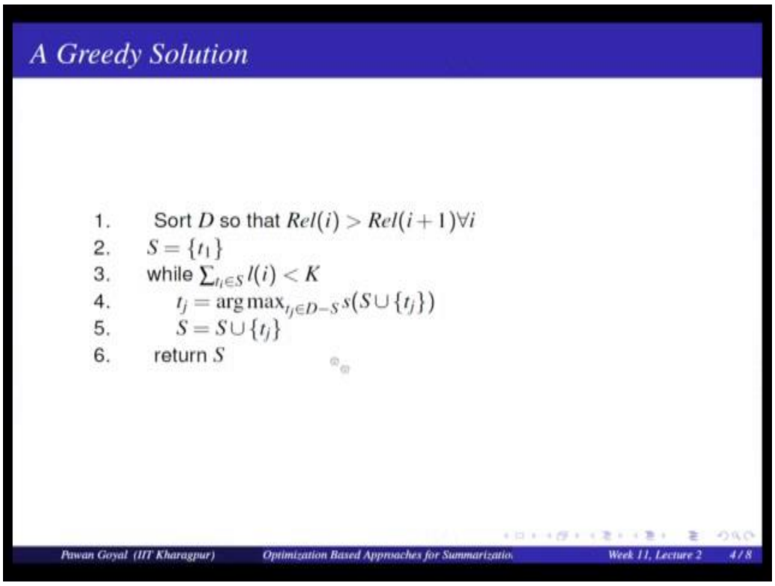
So you have some n textual units and you have also completed the relevancy scores say units t 1, t 2 up to t n and you have completed the relevancy score rel 1, rel 2 up to rel n. And as we have discussed this can be completed by many different manner ways. So let us not worry about that.

Now, once you have completed all that you want to maximize your objective function that is and yeah and yeah you have also completed your redundancy score of i j between every 2 units. So your objective function is I want to; somebody said that summation of relevance is minus redundancy between all the pairs is highest. So what you will do in greedy approach? You will first sort these, sort as per the relevance and you get some final sorting order. And let us say that order is t 1 to t n – t 1 has the highest relevance and t n has the lowest relevance. So what would be the greedy approach? I will say take t 1 in my summary already. So now, t 1 is there in the summary because it has the highest relevance.

Now the next unit will not be directly t 2. So we will say among all the possible units which unit I can add to my summary such that the score is maximized. So I am doing it one by one. So what is the score? Summation over relevance of t 1 plus t 2 minus redundancy of t 1 t 2, so that will the score for adding t 2, you will add t 2 to the summary. So this is your summary you will add t 2. And now this you call it your s prime and then compute your objective function for s prime with t 2. Then you will instead of t 2 you will put t 3 and compute it and so on you will do for all the t n.

And finally, you will have all different summaries all the scores take the one that is having the maximum score. So suppose some t 3 gets the maximum score. So this gets into my summary. Now again from the remaining units again keep on adding one by one to the at a time to the summary and see which one is giving you the highest score. Take the one and you will stop whenever you we have achieved the desired length of your summary that will be your final summary. So it is a greedy method.

(Refer Slide Time: 10:41)



A Greedy Solution

```
Sort $D$ so that $Rel(i) > Rel(i+1) \forall i$
$S = \{t_1\}$
while $\sum_{t_i \in S} l(i) < K$
    $t_j = \arg\max_{t_j \in D-S} s(S \cup \{t_j\})$
    $S = S \cup \{t_j\}$
return $S$
```

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarizatio Week 11, Lecture 2 4 / 8

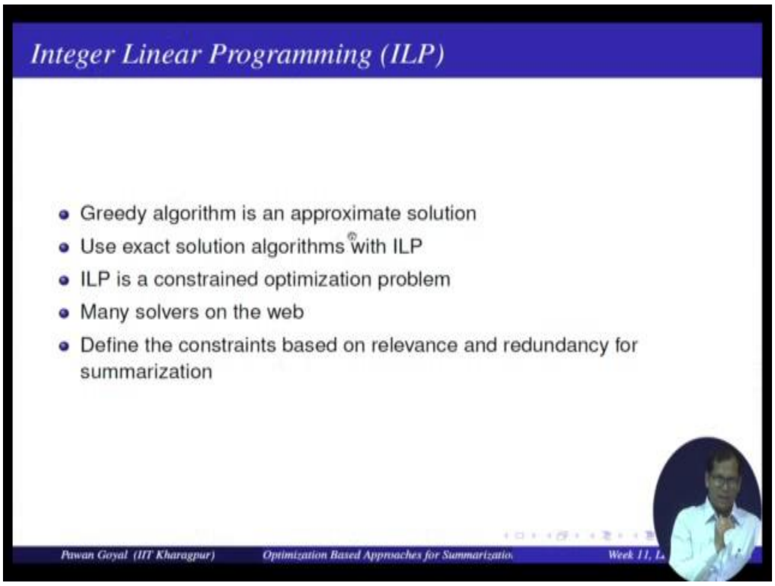
So what we are doing? So I am sorting my all the textual units my document such that relevance of ith unit is getting the relevance of i plus oneth unit for all t i. So you will get the score t 1 to t n, they are in the sorted order. Then what do you? Do you take the t the first unit in your summary? Then while you have not achieved the real length of the summary, for this loop you will take each sentence that is remaining each unit that is remaining put that in your summary as union t j and compute the score and get the r max over all t j’s. And that will be the one that you will include in your summary whichever gets the highest score. And then you keep on doing that until you have achieved the desired length of your summary.

And this is a very generic approach you have to define your own relevance score redundancy score and then you have to apply this algorithm to obtain a summary such that it has the highest relevance, but minimum redundancy. So you achieve both relevance and diversity at the same time by using this approach.

Now, we are saying that this is a greedy approach. Why is this greedy approach? You are picking you are not going to an optimal solution you taking the greedy approach that say it is near optimal solution it may not be an optimal solution. I am always choosing t 1, but it might happen that in my optimal solution I do not have t 1 because t 1 is similar to many other sentences. This can always happen. So this does not guarantee in optimal solution, but this is good enough if you want a decent solution. There are other approaches where you can use dynamic programming algorithm to find out what is the best way I can fill up my length k such that this is maximized of course, if you try out all possibilities, it might be exponential in nature, but you can always in dynamic programming. So that you are storing somewhere else what is the best way of achieving the summary of length 5, 10, 15 and so on and then use that for any higher length. So that is another approach.

So what we will discuss instead is an approach that is again very generic. And there although we have started with some optimization function some constraint you are free to choose your own optimization function and also your constraint and still this will give you an optimal solution. And suppose is called integer linear programming. So there you are trying to optimize some of objective function with respect to certain constraints and you are getting some integer solutions for your different parameters such that they maximize the objective function.

(Refer Slide Time: 13:25)



Integer Linear Programming (ILP)

- Greedy algorithm is an approximate solution
- Use exact solution algorithms with ILP
- ILP is a constrained optimization problem
- Many solvers on the web
- Define the constraints based on relevance and redundancy for summarization

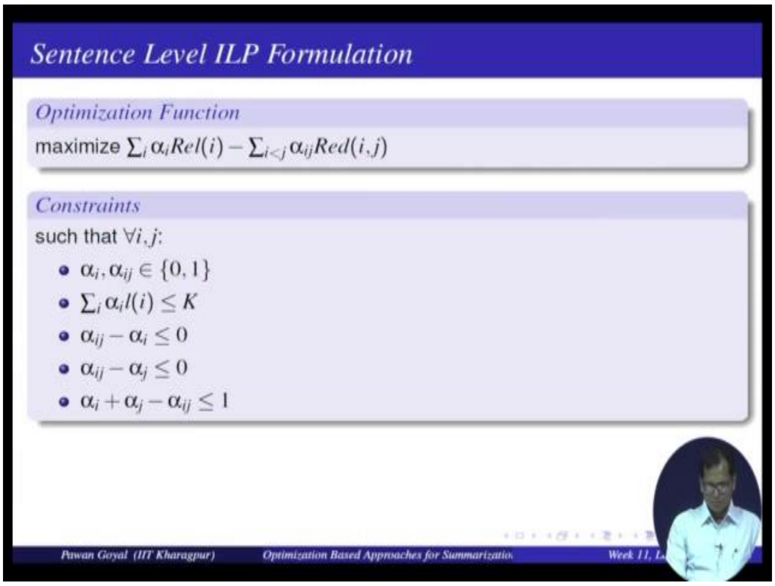
Pawan Goyal (IIT Kharagpur)
Optimization Based Approaches for Summarizatio
Week 11, L

So let us first see what is that. So we will talk about how to use the integer linear programming for summarization. So this is like a greedy algorithm. So greedy algorithm sorry; greedy algorithm that we discussed in the previous slide it gives me an approximate solution; so ILP will help you an exact solution that is the optimal value for this objective function.

Now ILP is like a constraint satisfaction problem. So you define the various constraints and based on those constraints you are trying to optimize your objective function. So important thing is that all your constraints should be satisfied now your constraints will be such that that are needed for your summarization for example, what is one very simple constraint that you will always put that summation over all the units that you have selected has to be less than or equal to k where k is the desired length, that is one constraint that you will always put, plus you might have some other constraints based on your task, we will discuss that once we will give the basic approach.

So one good thing about ILP is that there are many different solvers on the web and you can take any of those. And in the format that they have described you will put your objective function and constraints and they will give you the solution based on that. So initially to understand ILP we can what we can do? We can define our constraints and based on only the relevancy and redundancy. So they are 2 things that we talked about in the previous slide also. Let us define our constraint objective function based on that and then we will see what will be the form of the ILP.

(Refer Slide Time: 15:04)



Sentence Level ILP Formulation

Optimization Function

maximize $\sum_i \alpha_i Rel(i) - \sum_{i<j} \alpha_{ij} Red(i,j)$

Constraints

such that $\forall i,j$:

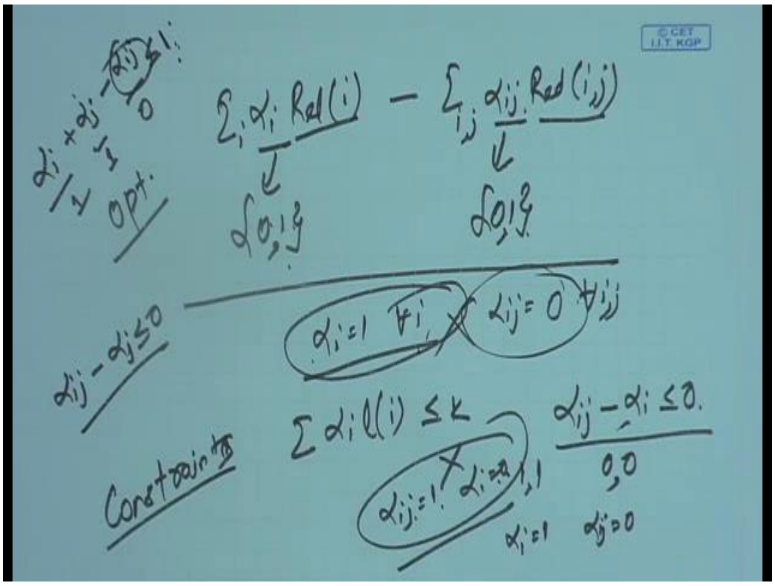
- $\alpha_i, \alpha_{ij} \in \{0,1\}$
- $\sum_i \alpha_i l(i) \leq K$
- $\alpha_{ij} - \alpha_i \leq 0$
- $\alpha_{ij} - \alpha_j \leq 0$
- $\alpha_i + \alpha_j - \alpha_{ij} \leq 1$

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarization Week 11, L

So here this is my objective function. So now, what we are doing? So again suppose that we have different textual units t 1 to t n. Now I also have defined the relevance and redundancy. So for now again like in the previous case assume that you know how to compute the relevance and how to compute the redundancy between 2 different units. So once you have defined all that now you are saying I want to maximize some function summation alpha i rel i minus summation i. So this is an ordering to ensure that you are doing it only once alpha i j redundancy of i j.

Now one thing you will notice that we are not in the earlier case we are taking a subset you were saying I will take a subset and for that subset I will maximize. This here we are not taking the subset, but instead what do we have here? We have some different parameters alpha i and alpha i j; now what are these parameters? So alpha i is a simple indicator that tells whether the ith unit is included in the summary or not, so I am saying alpha i summation alpha i rel i; so alpha i will indicate whether the ith unit in the summary or not.

(Refer Slide Time: 16:14)



So alpha i should have a value of either 0 and 1, 0 or 1. It can take only one of these values. And similarly alpha i j redundancy of i j alpha i j can again take a value only between 0 and only 0 or 1. Now what do these parameters alpha i and alpha i j indicate? 0 means it is not in the summary 1 means it is in the summary. So whenever it is 1 you are taking the relevance 0 it is not being counted. Alpha i j will be one whenever i and j both are in the summary and so that; that means, if i and j both are in the summary you have reduce you have to subtract the redundancy score. So you are trying to find a solution such that this is maximized. So now, you should you should put certain constraints.

So for example, if you do not put a constraint I can always find a solution where alpha is equal to 1 for all i alpha i j is equal to 0, for all i j i can all always put some solution like that, but you will immediately say this does not make sense because you are not subtracting the redundancy of the sentences, one thing. Second you are taking all the sentences in the summary. So what is the point of the summary you are taking the whole document.

So this optimization function is not sufficient until you add the constraints. So what are the constraints you will add? Since this is the optimization function and what will be the constraints. So constraints the first one that will come to your mind is that the length of the summary has to be less than equal to k. So how will you put that constraint? You see what is the; what is by this approach you want find the optimal solution for all this alpha i’s in alpha i j's. They can take value between 0 values 0 or 1. What is the optimal values. So you will put some constraints one constraint will be summation alpha i li less than or equal to k, li is the length of the ith textual element. This is less than equal to k. That will be the constraint that will immediately say this kind of solution is not acceptable, but again you can find a solution where some alpha i's are 1, but all the alpha i j's are 0. So you do not want that. You want that whenever alpha i and alpha j are 1 alpha i j has to be 1. So you need some more constraint for that. So what are the constraints we will put here? So we will say. So alpha and also you do not want alpha i j to be one and alpha i to be 0. It cannot happen.

So all these things how do we put together in a constraint let us see. So we are saying all the alpha i alpha j are either 0 or 1 that is the first thing. We are saying and summation i alpha i li is less than equal to k. Then the next 2 constraints are alpha i j minus alpha i is less than equal to 0. So alpha i j minus alpha i less than equal to 0 what is that mean.

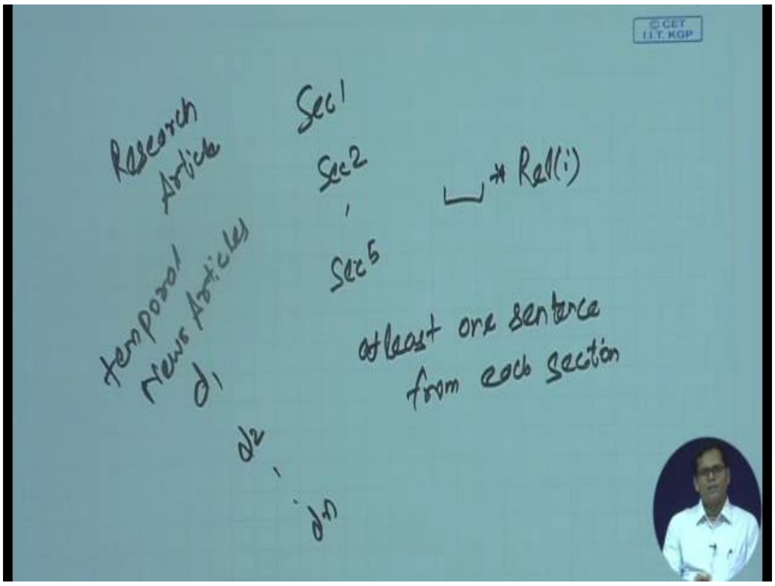
They can take values 0 or 1. So what are the possible values? Both can be one this is allowed right alpha i is one alpha i j is 1 is allowed both are 0 that is allowed, but what is not allowed. So and what is and also if alpha i is equal to 1 alpha j is equal to 0 sorry alpha i j is equal to 0. That is also fine this will be minus 1 what is not allowed. Alpha i j is equal to one alpha i is equal to 0, this is not allowed. Because then this will be 1. And why it is not allowed? What is it saying is that you are not including the ith unit, but the redundancy between i and j and this is not possible this you can pick only when alpha i is equal to 1. So this is a constraint like that similarly you can state now alpha i j minus alpha j less than equal to 0 and that you can understand in the same manner.

Now, what is the third constraint you are putting? That is alpha i plus alpha j minus alpha i j is less than equal to 1. Now what is this saying? So this can take all the values except a value of 2. Now when will this take a value of 2? So you can see that it cannot take a value of larger than 2 because it can be at most 1 at most 1 it can be at least 0. So maximum value is 2 and it is allowed to take values up to 1.

So what happens when it takes value of 2? That means, it is 1, it is 1, it is 0, but it ensures that whenever they are 1 you put it as 1. You are you are taking it as 1. So this constraint helps you in that you cannot put alpha i j equal to 0 whenever alpha i and alpha j are both 1, together these 5 constraints will actually give you the same sort of objective function that you were doing earlier, but now in the terms of integer linear programming. So now, your integers alpha i's all the alpha i's and alpha i j's you have to choose such that this objective function is maximized and the constraints are satisfied. And then you can run your ILP solver to actually obtain the solutions.

So what is good about this approach is that it is very generic enough. So depending on your task you can define what is your objective function and what are your constraints. So for example, here we took very simple constraints independent of your task, but suppose in your task.

(Refer Slide Time: 22:40)



So for example, you are doing a summary from a say research article. And research article has various sections. Section 1, section 2, and section up to section 5, and now suppose you want to put a constraint that in my summary I want at least one sentence from each section. So you can easily put that constraint in your ILP.

So you can define what is the section information for each of the unit and then say for each section from it you will have at least one unit in your final solution. So that is a constraint that you can add. Similarly suppose you have some sort of data where that is temporal. So suppose news articles and you have over various days, so day 1, day 2, day n. So this generally happens when you are doing some sort of time line summarization. So you have about an event you have various articles in a timeline and you are trying to summarize that. So there you want to put a constraint that I want, if a similar kind of information is there in 2 2 different time points. I want the later time point information or you can say that I want more information from the later time point than the previous time point.

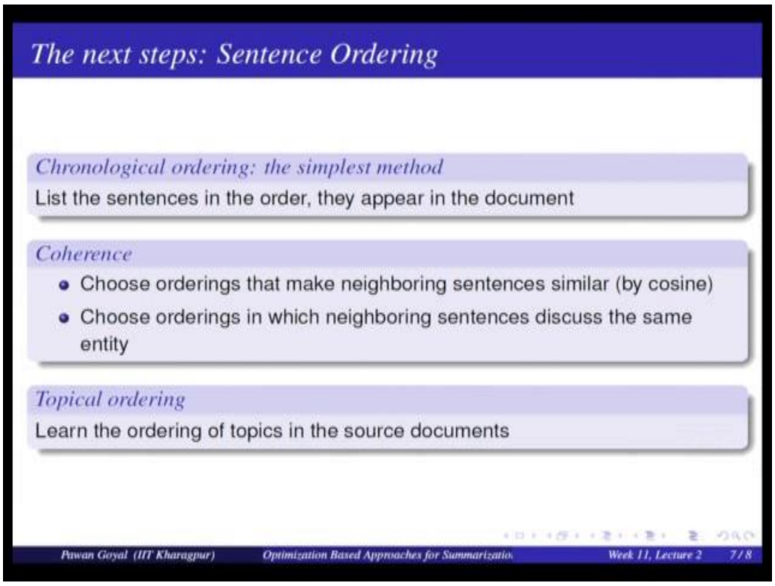
Again you can put them as your constraints you define like you defined your length you can define the time period and you can say certain time period I want more than the previous time period. This can also add as your constraint. Then you can also change your objective function. So here we said we will sum over all the relevancy score of the textual units, but suppose you define, I want the I want to put some weight to the importance to the importance to the different series like sections or days.

So we can multiply the relevance score by certain weight. So this can be the importance weight of different days. So this can also come in your objective function. So like that this is very generic approach where you can keep on changing your objective function. So you can also maximize multiple things together and you can keep on adding your constraints depending on your task and you will still be able to find the solution using ILP.

So once we have done this. So we have talked about different methods for doing summarization. A graph based approach and an optimization based approach. So what were we discussing in summarization? Once we have selected the important sentences the next step will be how do I order these sentences. Now ordering might also be some sort of a heuristic approach. So what is the simplest approach that you will take. So once you have found the sentences from the document, you will order them in the same way in which they occur in the in the document. So although their relevance might be that the sentence at the bottom has the highest relevance, but you will take the sentence that is coming first in the in the document as the first sentence and so on. So the order in which they occur in the document you will provide that set.

But there are other approaches also. And they are called like you want to optimize the coherence of the summary. So that it becomes more readable. So what is one possible way? So you do not give anywhere drastically different sentences together. So you will say I will try to put them together in an order such that the 2 sentences that are close enough are similar to each other. So that is one possibility. Or you can say that they talk about the same sort of main entities or events. So that is another criteria for ordering.

(Refer Slide Time: 26:42)



The next steps: Sentence Ordering

Chronological ordering: the simplest method

List the sentences in the order, they appear in the document

Coherence

- Choose orderings that make neighboring sentences similar (by cosine)
- Choose orderings in which neighboring sentences discuss the same entity

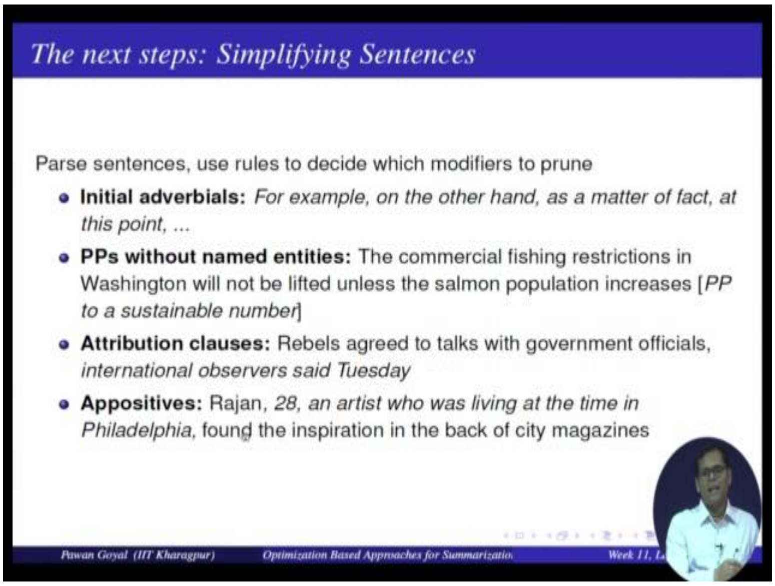
Topical ordering

Learn the ordering of topics in the source documents

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarizatio Week 11, Lecture 2 7 / 8

So either you can list the sentences in the order they appear in the document. It is also called the chronological ordering. And you can optimize the coherence that is choose ordering that make the neighboring sentences similar by cosine similarity. Or choose ordering in which the neighboring sentences discuss the same sort of entity. And you can also do topical ordering. So you find out in document what topics are occur in what order and accordingly you put the sentences belonging to first topic will come first and so on. That is another possibility. So once we do that another approach another step could that I want to further reduce some sort of redundancy from there and that can be in the in the way we write.

(Refer Slide Time: 27:35)



The next steps: Simplifying Sentences

Parse sentences, use rules to decide which modifiers to prune

- **Initial adverbials:** *For example, on the other hand, as a matter of fact, at this point, ...*
- **PPs without named entities:** The commercial fishing restrictions in Washington will not be lifted unless the salmon population increases [*PP to a sustainable number*]
- **Attribution clauses:** Rebels agreed to talks with government officials, *international observers said Tuesday*
- **Appositives:** Rajan, *28, an artist who was living at the time in Philadelphia,* found the inspiration in the back of city magazines

Pawan Goyal (IIT Kharagpur) Optimization Based Approaches for Summarizatio Week 11, L

So let us see what are the different ways in which you can simplify. So for simplification of sentences. So simplification is important because by simplifying sentences you can get more space into your final summary. So some sentences that contain some information that is not important you can remove that. And that is like not a very easy step. So what you will do? You will parse the sentences and remove certain specific clauses and phrases. Like for example, initial adverbials. So many sentences start with for example, on the other hand, as a matter of fact, at this point etcetera. All these are not important to convey the information and you can remove these initial adverbials.

Then some propositional phases that do not contain the named entities can also be removed. So like here the commercial fishing restriction in Washington will not be lifted unless the salmon population increases to a sustainable number. To a sustainable number is a prepositional phrase and this may not be very important to convey the information here. Because it does not contain any named entity. So you can also remove this. Then you can also remove the attribution clauses like rebels agree to talk with government officials' international observers said Tuesday.

So this attribution to something this may not be important. So you can remove this also from your summary. Then certain appositives can also be removed like Rajan 28, an artist who was living at the time in Philadelphia found the inspiration in the back of city magazines. So the appositive here is an artist who was living at the time in Philadelphia and this may not be important to convey the information. So like that you can also define many such rules. These rules can be again given manually to the system. And you can also learn these rules. If you have some sort of data that says what is the original sentence and how did some human simplify that. So from there you try to learn the rules that what kind of nodes in my parse tree can be removed and how I can simplify my sentences that is also possible.

So we discussed 2 3 different approaches of summarization. And how can you do some final polishing and post processing once you get the sentences to fit in a much smaller space. And as such there are many different approaches for summarization. You can always use these as your baselines, but you can also explore the other methods.

So in the next lecture we will briefly discuss that once you have done the summarization how do you go about evaluating your system summary. So you are getting some summary and how good is that. So what is the different methods we will talk about one specific method for that.

Thank you.