

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

**Lecture - 47**  
**Entity Linking – II**

Welcome back for the second lecture of this week. So, we have started talking about entity linking and we talked about two different approach; so one approach where we use the keyphraseness in commonness to find out what are the appropriate mentions and how do you link to them their corresponding reference in the knowledge base. And we were considering Wikipedia as our knowledge base. And we found out one particular problem with using simple keyphraseness and commonness.

(Refer Slide Time: 00:50)

*Keyphraseness and Commonness: Always the best decision?*

**Depth-first search**  
From Wikipedia, the free encyclopedia

**Depth first search (DFS)** is an **algorithm** for traversing or searching a **tree** **tree structure** or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking.

Formally, DFS is an **informed search** that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hadn't finished exploring. In a non-recursive implementation, all freshly expanded nodes are added to a **LIFO stack** for exploration.

sense	commonness	relatedness
Tree	92.82%	15.97%
Tree (graph theory)	2.94%	59.91%
<b>Tree (data structure)</b>	<b>2.57%</b>	<b>63.28%</b>
Tree (set theory)	0.19%	34.04%
Phylogenetic tree	0.07%	20.33%
Christmas tree	0.07%	0.0%
Binary tree	0.04%	62.43%
Family tree	0.04%	16.31%
...		

*Using Relatedness: Basic Idea*

- In a sufficiently long text, one finds terms that do not require disambiguation at all.
- Use every unambiguous link in the document as context to disambiguate ambiguous ones.

Pawan Goyal (IIT Kharagpur)      Entity Linking - Part II      Week 10, Lecture 2    2/9

So, what is the problem? So, in commonness what we were doing, we were always taking the page that is having the highest commonness. So, what will happen? Suppose I have word like tree, and the commonness to the sense tree is 92.82 percent, but the other concepts because they occur rarely commonness is like 2.94 percent, 2.57 percent and so on. So, what you are seeing wherever the word tree occurs you will by default assign it to the first sense of tree, and you will not look at the context at all.

So, in this case, so you have it article about depth first search and you have a sentence where the word tree occurs, and because of using commonness, you will always assign it

to the sense tree. But the correct sense says here is tree data structure, but it has a very small commonness, so only 2.57 percent. So, what you need to do for assigning it to the correct data structure correct sense there is a tree data structure. For that you should be able to use the context here that is what are the referent words that I am seeing in the context.

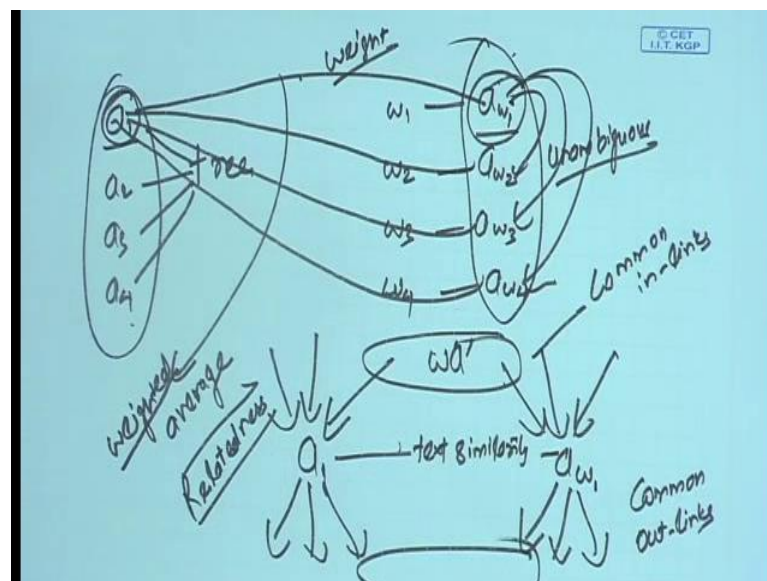
Now, question is how best we can use the context. Now, I do not want to use some random words in the context, I want to use the words in the context that have actual correspondence to a Wikipedia page, so that I can find out something about the Wikipedia page how common this Wikipedia page is to one of it is reference. Now, there we fall into the same problem that how do I use a Wikipedia reference to any of its page when the disambiguation has not yet happened, so I am only at the stage of disambiguation. So, how do I use the actual entities, entity page?

And for that, a nice tree can be used. And the idea is ok, so with this word there are many other words that are coming in this article or this piece of text, and some of these will be appropriate mentions, and they will into one or many Wikipedia pages. Now, some of out of these at least some will be there that have a unique disambiguation page in Wikipedia. So, there is a unique page in Wikipedia where they link to. And there I do not need to do any disambiguation. So, why do not I use only those pages which have a unique page in Wikipedia to find out what should be a good of a Wikipedia page for this entry tree, so that is what is done. So, this is the hypothesis that if you have a sufficiently long text, you can find out terms that do not require disambiguation at all; there will be some terms that have only one mention or one referent in Wikipedia.

Now, use this unambiguous link in the document that context to disambiguate the ambiguous ones. So, what is the idea here, so you are given this article and you want to find out what is the appropriate sense for this word is. So, in a sense, I mean what is the appropriate link in Wikipedia, is it tree, tree graph theory, tree data structures, set theory etcetera. So, what I will do I will see what are the other mentions in this page. So, what are the other things you have seen algorithm, tree structure, graph, backtracking, uninformed search, tree backtracks, LIFO stack and so on. Now, among these there will be some that have only one Wikipedia page as the referent, so these are shown as box search.

So, algorithm, tree structure, uninformed search and LIFO stack have only one Wikipedia page. So, I will take these unambiguous links and try to find out how close these four links are to my all of these possible senses. So, how close are these four links to this possible sense and the one sense that is having the closest to these four will be called by (Refer Time: 05:11) sense, this will be the link to which I will link my corresponding mention.

(Refer Slide Time: 05:19)



So, how do we compute the relatedness score, and this can be very simple. So, you can initially start by representing each candidate sense and context term by a single Wikipedia article. So, for example, what is happening now, so a word like tree, and tree corresponds to many different senses and call them your article 1, article 2 article 3, article 4. And if you remember your word sense disambiguation this, it is like constructing various sense backs, there are four different senses they are like sense back. Now, you are having a context back kind of context track, where you are saying ok in this context of tree I am finding four different words word 1, word 2, word 3, word 4. Now, what is the property of each word, they link to one page in Wikipedia, so like a w 1, a w 2, a w 3, a w 4. So, now, these are Wikipedia articles and they are also Wikipedia articles.

(Refer Slide Time: 06:43)

*Computing Relatedness*

- Each candidate sense and context term is represented by a single Wikipedia article.
- Thus the problem is reduced to selecting the sense article that has most in common with all of the context articles.
- Comparison of articles is facilitated by the Wikipedia Link-based measure, which measures the semantic similarity of two Wikipedia pages by comparing their incoming and outgoing links.
- The relatedness of a candidate sense is the weighted average of its relatedness to each context article.

*How to give different weights to the context terms?*

Pawan Goyal (IIT Kharagpur) Entity Linking - Part II Week 10, Lecture 2 3/9

Now, the problem is select the sense article that is the most in common with all of the context articles, so that is among the four articles, which is most common with all these four articles. And you will see what is the argmax that is having the more similarity with these four articles. And this can be captured in many different ways. So, we will talk about one particular method. So, one particular method is you just take the Wikipedia link based method that is two pages are similar if they are having many incoming and outgoing links common.

So, what is the idea so how do you find out how similar a  $w_1$  is to a  $w_2$ . So, I will say I have two articles  $w_1$ ,  $w_2$  in Wikipedia, I find out what are the incoming links to this article, and what are the outgoing links from here; same I will do for this article. And now once I have found this out, I can see ok, what are the common links. So, how many articles in Wikipedia article  $w_3$  are linking to both of these. Similarly, what are the articles to which both of these links to? And these are very good measures for finding out how similar they are. You see we can always do it by seeing how similar they are by measuring their text similarity. How much text similarities is there, you can capture cosine similarity and what or something else.

But this is a nice link based measure that says ok, how many pages linked to both of these, so what are the common in links, and how many pages they both linked to that is a common out links. And this is again a nice measure in that it says ok; this article refers to

both of these that means they need to have something common similar they both mentioned the same article again that means, they need to have something in common. And you will find out how many what fraction of incoming links are common, what fraction of outgoing links are common and that you will take it as a measure for computing how similar these two articles are. And this is also called by relatedness. So, we talked about keyphraseness, commonness and this is relatedness.

And then you can find out the relatedness of a candidate size sense by taking a weighted average of its relatedness with all of the context articles. So, that is to find out the relatedness of this sense a 1, you will say ok, it is relatedness with a  $w_1$  with a  $w_2$ , a  $w_3$ , a  $w_4$  by using this measure, then you take a average or a weighted average - computed weighted average. And this will be what is the relatedness of this sense a 1, a 2, a 3, a 4 whichever as the highest, you take that. Like, if you see the previous slide, so here you were capturing showing relatedness of various senses. And this tree data structure had the highest relatedness 63.26 percent that uses the average of relatedness with all these the four different context articles.

Now, so there is one term here we are taking a weighted average. Now, what should this weight depend on, why should I weight one of this higher than the other ones. So, again if you think about it, it can depend on which context term is more important than another. So, what we have done we have taken the context, we have found out to all the words that are mentions; and from there whichever are so these were unambiguous, whichever were an unambiguous we are taking them as my context to find out the relatedness. But some of these might be more important for this topic other documents than others. So, can I give a weight depending on how important they are to the topic?

Now, the question again comes how do I know which one are more important to the topic or the theme of this document. And there you see you can again use the idea of relatedness that means one among these four context senses or articles the one that is having the highest relatedness with the others which is more appropriate to the theme of the document. Yes, because there is a theme of the document and the words that are appropriate should also be connected to each other that means, a word that is having a high relatedness with other words is it should be given a high values, and that is a nice method to also give a weight here. Weight to different of these relatedness that is how related these this context article is to the other context articles.

(Refer Slide Time: 12:22)

*Weighting the Context Terms*

- **link probability:** Use the ones that are almost always used as a link within the articles where they are found, and always link to the same destination
- **relatedness:** We can determine how closely a term relates to the central document by calculating its average semantic relatedness to all other context terms

*These two variables - link probability and relatedness - are averaged to provide a weight for each context.*

Pawan Goyal (IIT Kharagpur) Entity Linking - Part II Week 10, Lecture 2 4/9

So, in general, there are so what are the things that are used to give a weight to the context term, so one is called link probability. So, what is link probability, so again in your context, you are finding say four or five articles where the link is unambiguous, there is only one link. Now, you can use the link probability itself that is among the four which one is like a keyword that is it always links to something. Some words may not link may not always link some words always link. So, the words it always link should be given a high weightage, because I know this is a more a specific term if a word is sometimes links sometimes not linked it may not be a very important keyword, so that can be one measure. What is the link probability, probability that it will be given a link in general Wikipedia that is same as my keyphraseness measure? And certain thing we have already discussed that is relatedness.

So find out how closely it relates to the central document by computing it is average relatedness to all other context terms. So, I take for each word what is the link probability or keyphraseness and relatedness. Now, I have two different measures, so how do I take these together to compute the weight of this word, you can simply take an average to provide the weight for each context, so is that clear now. So, you have some words in the context they are unambiguous. For each word, for each context word you find out the relatedness of this mention in your mention sense one of the sense, taken weighted average and this weight depends on what is the link probability, and what is the

relatedness with all the context terms. And by doing this method you can find out relatedness of each of the four senses.

Now, so we have discussed, we can take some mentions by some method, and then once with the mentions we can also give a link by finding out which of the candidates are similar in from with the context mentions. But here there is an interesting question that by using all this when we are doing all this approach, can I go back and also improve my mention detection part, so how I was detecting the mentions.

(Refer Slide Time: 15:03)

*Can we improve mention detection with this approach?*

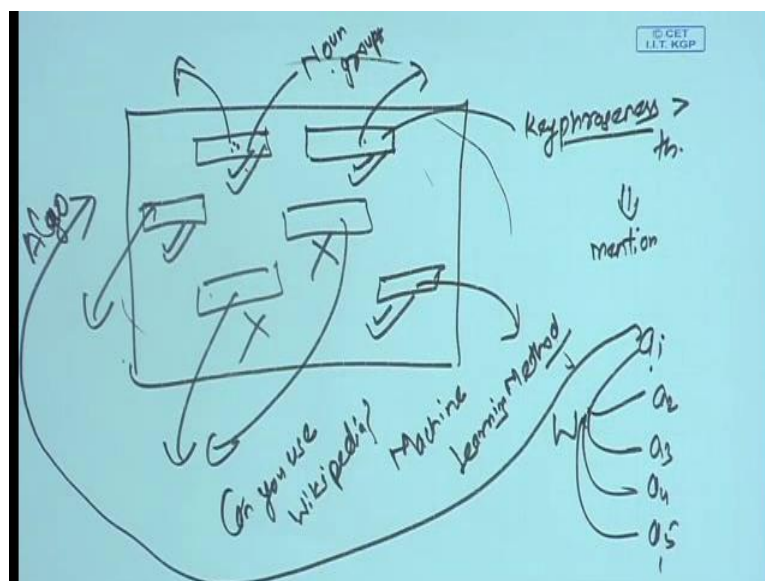
- The link detection process starts by gathering all n-grams in the document, and retaining those whose probability exceeds a very low threshold. *Is it the best method?*
- All the remaining phrases are disambiguated using the approach mentioned earlier.
- This results in a set of associations between terms in the document and the Wikipedia articles that describe them.

*Can you use this to learn – which concepts should be linked?*

Pawan Goyal (IIT Kharagpur) Entity Linking - Part II Week 10, Lecture 2 5 / 9

So, I was gathering all the n-grams in the document, and retaining only those whose probability exceeds a very low threshold.

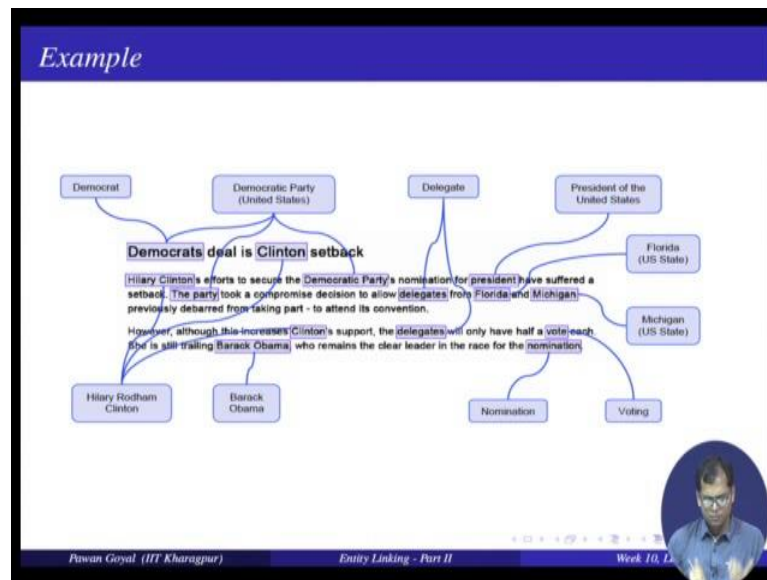
(Refer Slide Time: 15:14)



So, that is I start with a text document there I take various n-grams, it can be I take certain pattern they are noun groups or something. I take some patterns n-grams, I can say 1, 2, 3 whatever and I take some n-grams and then see what is the keyphraseness of each of these and whichever has if this is above a threshold, when I take it as a mention. And then it is a mention and then I go to my link disambiguation part, but see are you seeing that when I am finding out the appropriate entities as mentions, I am not using the context at all. I am just seeing is it a good word for, is it a good key phrase or not, does it have a good keyphraseness or not overall, so independent of a context is it a good mention or not.

So, question is can I also use the context to find out what are good mention and what are not so good mentions and that is what we will see. So, is this the best method? So, all the remaining phrases are disambiguated using the approach mentioned earlier, yes, so we have whatever mentions we have found or whatever phrases we have found we do disambiguation using an approach that we have. So, now by doing this approach you get to find a lot of different things like what are the links, what are the appropriate mentions here, what are the Wikipedia pages they link to? So you are getting some new Wikipedia pages also, now can you use this additional information to find out are they good candidates for mention at all, so that is can you use that to find out which concepts should be linked.

(Refer Slide Time: 17:14)



So, here is one example. So, you are having a Wikipedia page, so it is like a news article, so democrats deal is Clinton setback. And you are having lot of so various sentences are here. Now, what is your approach, in your approach, you take a various mentions like Hilary Clinton occurs at various locations and try to find out what is the entity in Wikipedia it will link to. Similarly, here Barack Obama, nomination, vote, Michigan all these are link to their various Wikipedia entities. So, you get all this by your link disambiguation phrase.

Now, my question is can I use that together to find out what are good mentions also from my text. And for that we have to convert that to some sort of a learning problem. Learning problem where I run my algorithm on a data, and see what are the mentions I am detecting what are the links, I am connecting to. Now, once I have all this information, someone gives me gold standard that what are the good links here, what are not the good links, using that can I learn what are my what are the good candidates to be mentioned, what are not so good candidates to be mention. So, like coming back to my example so I start with the text data, and I find out ok, there are some mentions, they have a key phrase above a threshold.

Now, I also link them to their Wikipedia articles some of these might be linked to the same article, so that I do for this whole document. Now, suppose someone tells me that actually this is a good mention, this is a good mention, this is a good mention, but this is

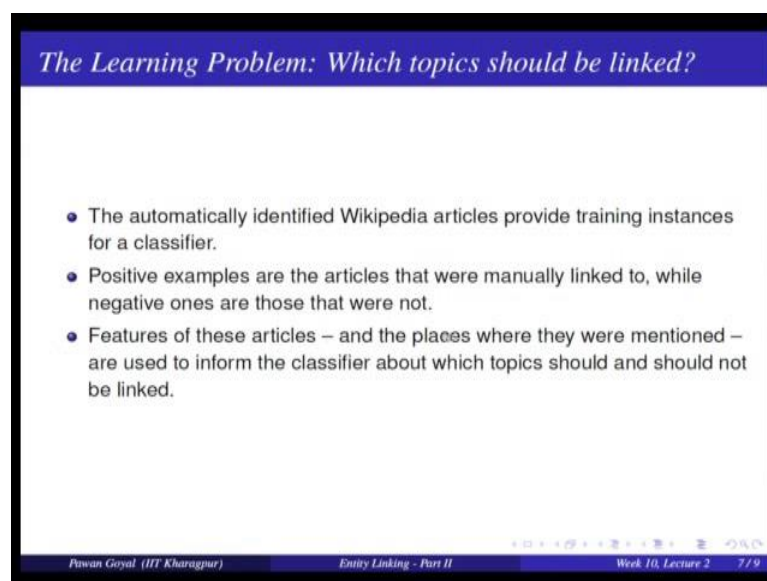
not so good mention, this is good mention, this is not so good mention. So, once I have all this information, can I develop a machine learning method to detect ok, given an article, given in mention and it is approved Wikipedia page all the context, is it a good mention at all. So, given a phrase with all these attributes is it a good mention for this document or not.

So, now so you can say that once you given me the text, all the all the steps that I have taken are deterministic. So, I can apply keyphraseness, I can find out the mentions, I can link them to their Wikipedia pages, so all this I can easily do. But how would I get these gold standards that this is in appropriate mention, this is not in appropriate mention and this is one of the bat bottlenecks, so how do I get this actual links and not so good links and good mentions and not so good mentions.

For that, so now what is interesting idea here, can you use Wikipedia again, so can use Wikipedia again. So, how would you use Wikipedia for this? So if you think a bit, so how you can use Wikipedia and that is actually very, very easy. So, you take Wikipedia and take some Wikipedia articles say a 1, a 2, a 3, a 4, a 5 so on. Now, each of the article now forget the hyperlink structure here, so take it as a plain text and feed it to your algorithm. So, algorithm takes a 1 as input plain text and runs this. So, your algorithm will run this it will tell you what are the mentions, what do they link to and so on.

Now, because a 1 is already, so it is already in Wikipedia, you know what mentions are good, what mentions are not good. So, from there you can automatically construct your gold standard. And once you have the gold standard, you can apply a machine learning method to say which given a feature around this phrase, is it a good mention in this context or not, and that will solve your problem. So, this is like you are learning to link using Wikipedia. So, using the Wikipedia data and so very nicely you are taking it as a training data and also constructing your gold standard from this without having some manual efforts of labeling, because otherwise you will see this labeling will take a huge amount of time and this will help you do that automatically

(Refer Slide Time: 22:33)



*The Learning Problem: Which topics should be linked?*

- The automatically identified Wikipedia articles provide training instances for a classifier.
- Positive examples are the articles that were manually linked to, while negative ones are those that were not.
- Features of these articles – and the places where they were mentioned – are used to inform the classifier about which topics should and should not be linked.

Pawan Goyal (IIT Kharagpur) Entity Linking - Part II Week 10, Lecture 2 7/9

So, what will I do? So, now once I have taken the Wikipedia as input I know ok, whichever phrases gave me a Wikipedia article that was actually there in the original article, they are possible examples and whatever was not there becomes a negative example. And so you got the possible, negative examples and you feed it to your classifier. And then you use various features around these various mentions and articles to detect whether it is a good mention or not. So, you use various features like the phrases where they were mentioned to inform the classifier about which topic should and should not be linked.

So, now, what can be these possible features that you can use from a given n-gram phrase. So, let us see some features, some of these features are what you have already seen, and some other features can depend on how do people actually write a Wikipedia article. So, what are the good features? So, one feature that we can use is link probability, so that is for a given mention what is the link probability. Now, if it is occurring at multiple places like Hilary Clinton, Clinton, so they are occurring at different, different variations. So, what are they their link probabilities at each of these phrases? Take either the average or the maximum of this link probability, and that can be one feature. So, this you are doing jointly so Hilary Clinton and Clinton together, should they will linked or not. And here you are trying to use what is a link probability it at different places taking average or also taking maximum, both can be a features.

Then you can use the relatedness so how related these phrases are to the central theme of the document. So, again you will find out, what is the relatedness of these mentions with different unambiguous links in the entire article. So, this can be another feature. If they are very highly related then only you will take them as your mentions; if they are not related to the entire theme of the document; that means, they are not probably not good candidates for mentions.

Then you can also use the disambiguation confidence that is when you are trying to do a disambiguation over this mention how confident your classifier is. If your classifier is not very confident that means, you do not have sufficient context in those document and it may not be a good mention at all. So, this confidence can also be one of the features.

Then you can use the generality that is when you are trying to link some phrases in your text, so what is the idea you do not want to link something that is very, very generic that all people already know about. So, you want to link the phrases that are very specific, so how can you know about the how generic or specific a particular phrases further you can use the category tree of Wikipedia. And there you can see at what depth in the tree this particular mention comes in. So, if it comes at a very top label itself that means, it is a it is a very generic term, but this coming very low in the tree that means, say a specific term. So, specific terms might be given a high preference. So, this can also be like your feature what is the depth in the Wikipedia hierarchy tree.

And then you can also see how the documents are written that is where all this entities mentioned. So, for example, if it is a good entity, it will be mentioned in the introduction; similarly it will be mentioned in the conclusion of section of article. So, if it is mentioned in the initial few lines or the last few lines, it might be important. So, you can simply measure the offset from the beginning and the end. Then you can also see the spread that is what is the distance between then you shall mention in the last mention, so that is how far does the response across the document. If the spread is high that means, it might be a good mention; if the spread is low that means, very to only cover very small topic of this document this has been used. So, this can again be a feature for this task.

And you can think of many of other features combine these features in your classifier, and then you are learning whether given this phrase with all these features is it a good

candidate for mention or not. And this is like you are learning to link using the Wikipedia structure. So, as such you take many different methods, but this is the basic conceptual idea about entity linking that how do you detect mentions different methods, once you detect mentions how do you link them to their appropriate entries in the Wikipedia or any other database, and can you use this task to also improve your mentions. And you can take it in different, different applications, take different databases, and you can try out various variations for this task. So, this so that is where we finish our discussions on entity linking.

So, in the next lecture onwards, we will start talking about information extraction that is from a document where there is a lot of unstructured data, text data, how can you identify various entities and the relations between them.

Thank you.