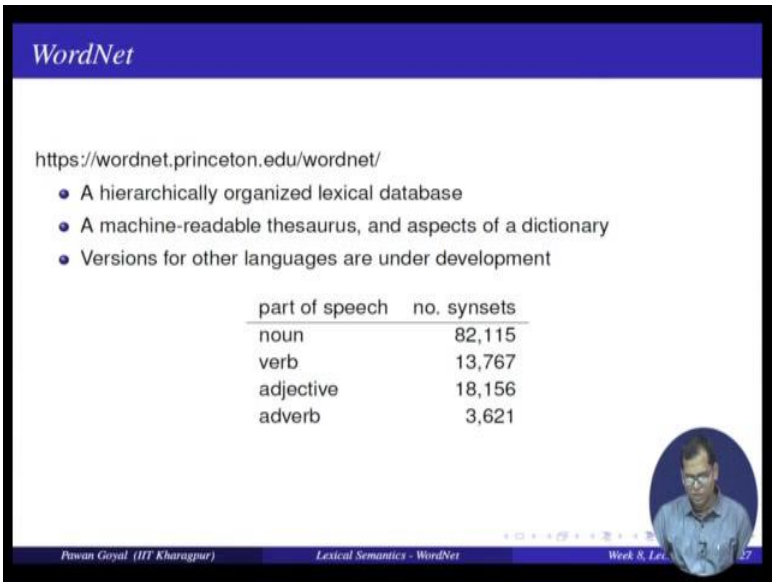
Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 37
Lexical Semantics – Wordnet

Yes, welcome back to the second lecture of this week. So, in the last lecture, we started with lexical semantics and we defined various relations between lexical entities, and we talked about polysemy, hyponym, hypernym, meronymy and so on. And in this lecture, we will see a particular resource wordnet and how it captures all these relations.

(Refer Slide Time: 00:48)



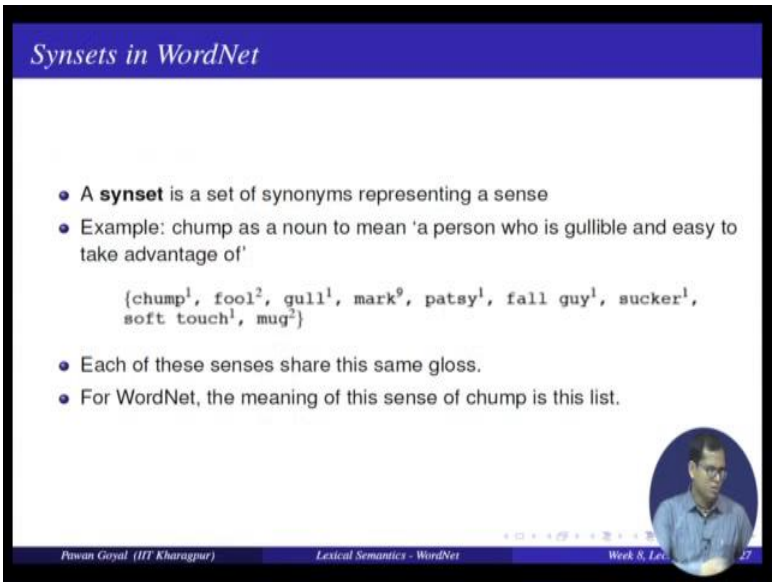
So, wordnet, you can get a lot of information of wordnet on this website. So, it is an effort started at Princeton; and there are lot of versions that have come up. And you can find out the latest version of wordnet and also download that and you can use that also, once you download you can use that in your command terminal also. So, what is wordnet? Wordnet as such is a hierarchically organized lexical database and it is completely machine readable.

So, you can use that in different applications, you can call wordnet get the recent information from there. So, as such on this website, you will find the wordnet for English, but there are other so there are many, many other versions for other languages also built. So, there are lot of wordnets available for European languages, and a lot of

effort has happened in the last decade for building wordnets for Indian languages and that you can also download many of those versions are also available for download for free. So, for that you can look at indo wordnet website. So, there you will also find out what sort of synsets or concepts in one language are relate to some other concepts in other language. So, this information would be available in this indo wordnet and euro wordnet websites.

So, now we will focus only on the English wordnet part in this lecture, but the methods would be easily applicable to other wordnets that are having a very similar structure. So, if we talk about English wordnet, so there are mainly four different part of speech words that are present there. And what we are seeing here; how many different synsets are there for each part of speech. So, there are roughly 18,000 synsets for nouns, 13,000 plus synsets for verb, and 18,000 plus synsets for adjective, and 3,000 plus for adverbs now. So, one important thing is that we when we talk about wordnet we talk in terms of synset. Now, what is the idea of a synset in wordnet. So, in synset what will happen, so various different words that have similar meanings will be stored, but a single word might have multiple meanings also. So, how are both of these things taken care together?

(Refer Slide Time: 03:18)

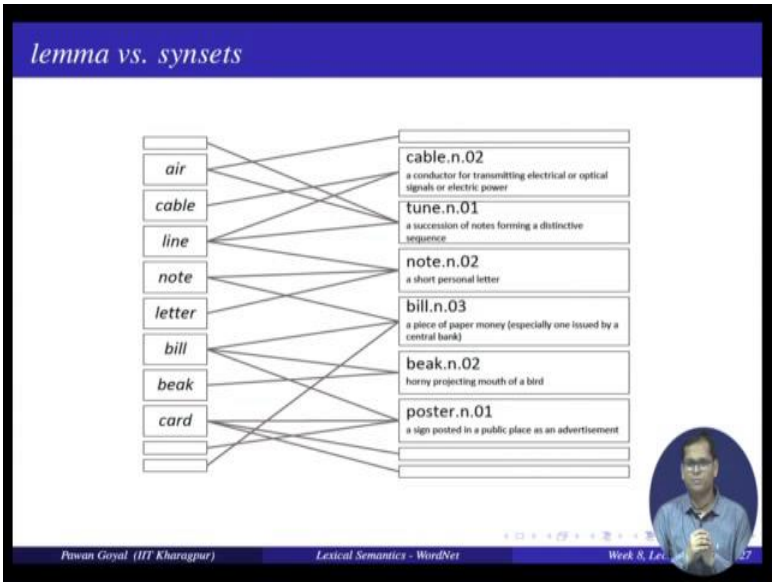


So, let us take an example. So, example is particular synset corresponding to chump that is a noun that means a person who is a gullible and easy to take advantage of. And suppose chump the first synset has this particular meaning. So, word has have might

have multiple synsets. So, chump first synsets is this meaning, but there might be other words also that have the sense. So, this word, its 9 sense denotes this particular meaning. So, I have to say that the word mark its 9th sense is this meaning; the word fool, its second sense is this meaning; the word gull the first sense is this meaning, and similarly fall guy and so on.

So, what I will do here? Each word might be represented by multiple synsets. So, a word might have sense 1, sense 2, sense 3 and so on. And what I will do, if suppose the third sense of word 1 and second sense of word 2 shared the same meaning I will put them in a single synset and that is the idea of synset, different words that share a same meaning. And by words here I will mean the lexical or a particular sense. And for wordnet this list might denote what is the meaning of the word chump or you will also find some gloss information of the wordnet.

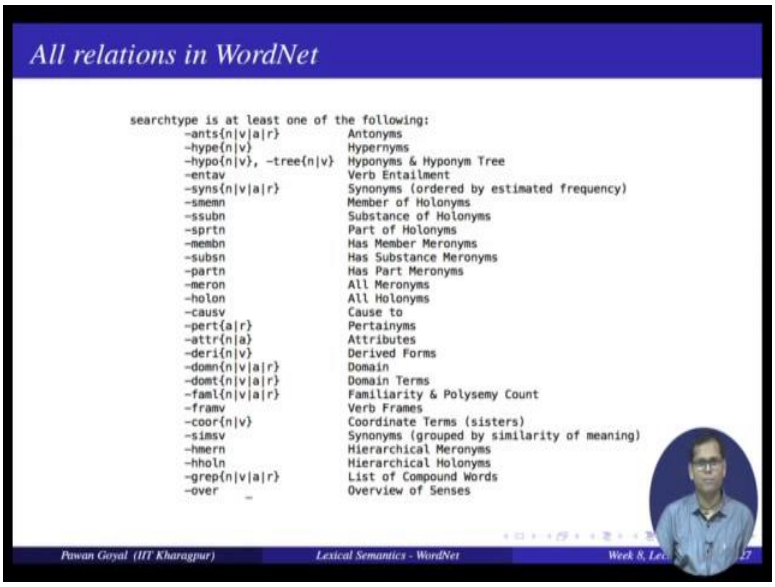
(Refer Slide Time: 04:50)



So, now this also this figures some sort of explains what is the relation between the word form or the lemma and the synsets. So, what you are seeing here? So, there is many to many relation. So, that is for example, the word note, it can go to note n 2 and bill n 3. So, denotes two different synsets in one sense, it goes to note synsets another sense it goes to bill synset. But this bill synset will contain note and also contains bill and some other words. So, what you are seeing here? Many different lemmas that share the same sense can come together in a single synset; and the same lemma for different meanings

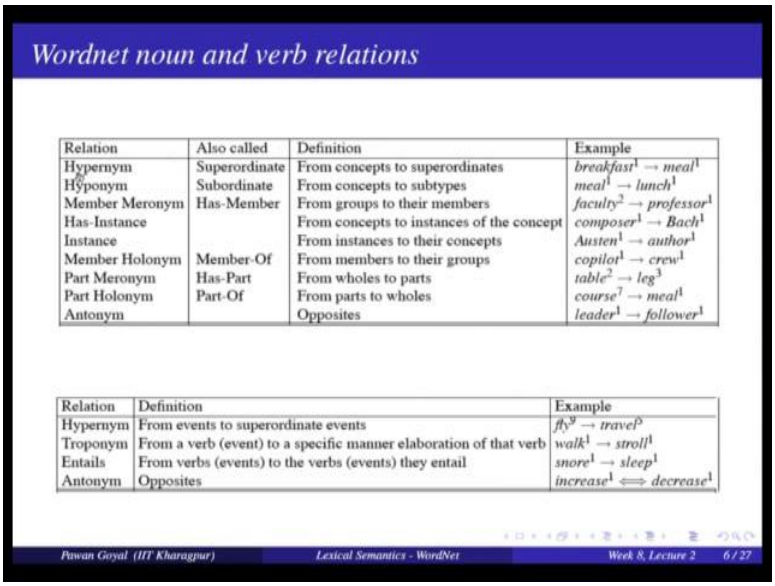
can go to different synsets, and that is captured by giving some unique identifies to different two different synsets of the same lemma. So, I will start talking about now line 1, line 2, line 3 instead of just saying line; and each of these three will be in different synsets.

(Refer Slide Time: 06:02)



So, what are all the possible relations that you see in wordnet here is some example. So, in wordnet, you have relations like antonyms, hypernyms, hyponyms, entailment relations, synonyms and so on, many of these we have already seen in the previous lecture.

(Refer Slide Time: 06:26)

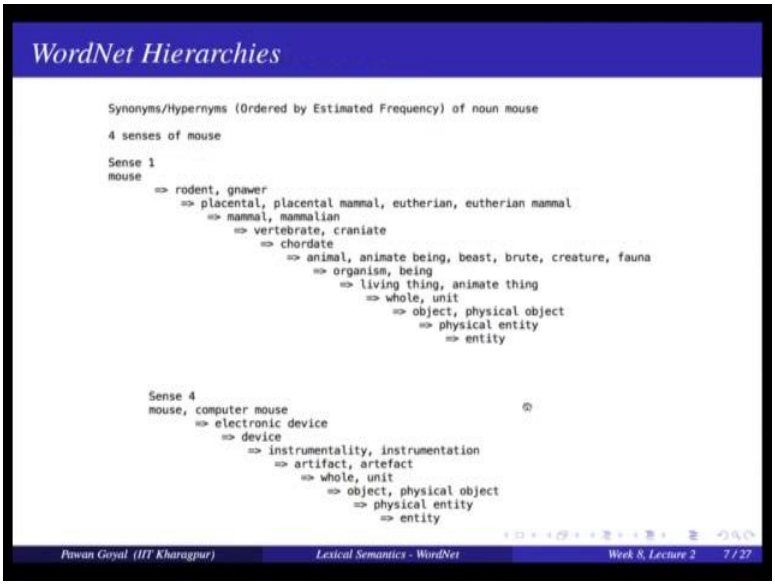


Let us see some example types of these relations. So, like what are the relations between various nouns. So, the relation hyponym that we discussed, so hyponym is relation between a concept and its super concept. So, like breakfast is a kind of meal. So, as was a meal is a hyponym of breakfast. So, this relation will be there in the wordnet where the breakfast 1 synset or the corresponding sense it is connected to meal one by the relation of breakfast is a hyponym of meal and meal is a hyponym of breakfast. Similarly, the converse relation of hyponym meal and lunch, member, so professor is a member of faculty; has- instance, is composer and instance is Bach; Austen is a instance of author; member holonym like capital is a member of crew; part meronym from whole two parts. So, table to leg. Part holonym the other way round from parts to wholes, course to meal, yes, and antonym that is a opposite leader and follower.

So, what would happen? All these relations are defined in wordnet, you know what particular relation means and then you can find out for a particular relation, what are the different pairs or given pair in the wordnet is there any relation that is defined. Similarly, there are relations between verb elements also like hyponymy, fly and travel. So, fly is a kind of travel. So, travel would be the hyponym for fly. Troponymy from a verb to a specific manner for that verb like, walk and stroll, so stroll is a typical manner of or a particular manner of walking. There can be entailment relation also from verbs to the words that they entail like snoring and sleeping. So, I will say snore and tell sleeping. So,

this is also relation that is captured in wordnet, and then there can be antonyms for the opposites, so increase and decrease, they are antonyms for each other.

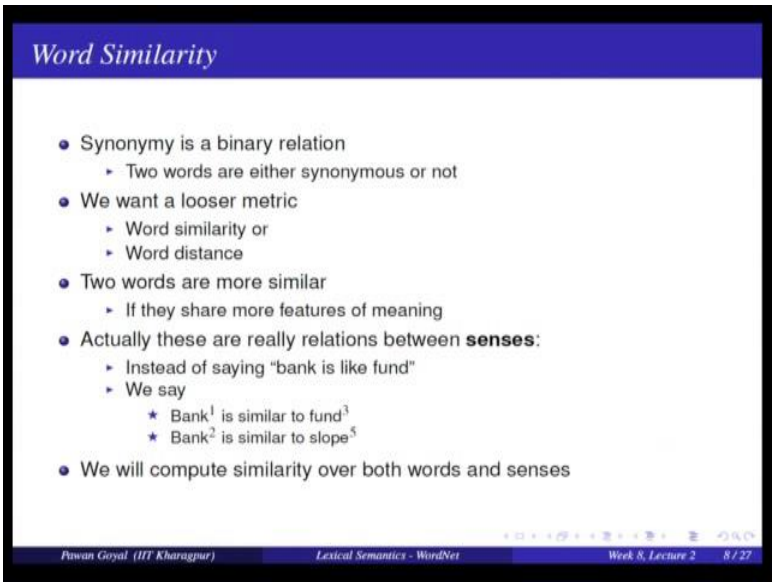
(Refer Slide Time: 08:51)



Further, in wordnet, we can also capture what is the complete hierarchy of a given concept. And what do I mean by hierarchy that is starting from the root word and wordnet how do you go down to that particular word. What are the different concepts that you need to pass through, and that you can find for all the different synsets for a given word, so that is why we said that wordnet is very, very hierarchically organized database. So, you can find out or you can locate each concept or a subset each synset in that complete hierarchical tree.

So, for example, if I see the word mouse, so I will find it has four sense. So, for sense one – mouse, you can find out the complete hierarchy. So, mouse is a kind of rodent, kind of placental, kind of mammal, vertebrate and so on up to animal, living thing, object physical entity, and entity you can go to the root of the wordnet entity, it starting from this particular sense of mouse. Here is another sense of mouse that is a computer mouse. Again you can keep on going up the hierarchy electronic device, device, artefact, object, physical object, physical entity, entity. And you can see where do they depart. So, mouse comes under whole then an artefact, and this mouse comes under living thing that is where they depart in that wordnet hierarchical tree. And this we can do for any sense any word in the wordnet, you can find out the complete hierarchy.

(Refer Slide Time: 10:32)



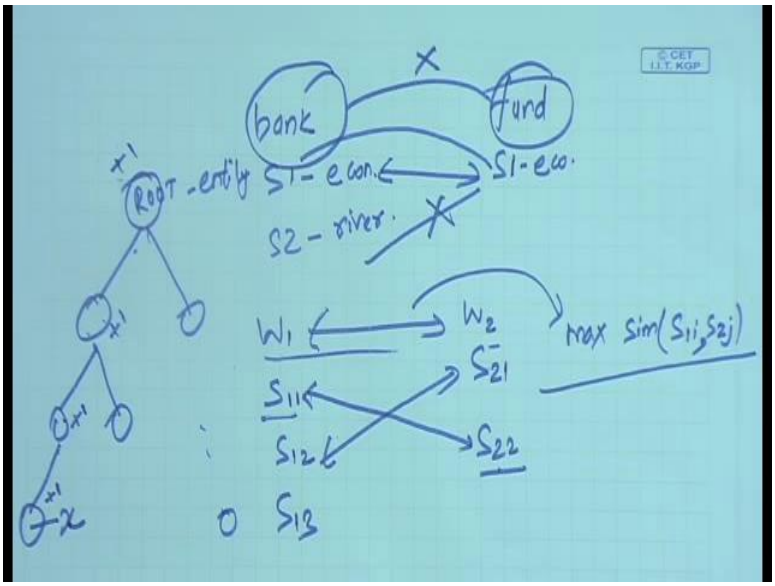
So, as such wordnet will always store which two concepts are related by word relations. And suppose car and automobile occur in the same synset, if you query in a wordnet, there is a very easy way you can query in wordnet, and you can find out if they are part of the same synsets then you can say ok they are similar. But suppose two words are not part of any sense any particular synset in wordnet, and I want to give it a number like to what degree are they similar, similar to what we did in the case of distribution semantics, we found out how much two words are similar to each other, by seeing how much their patterns are similar. So, can I do something in the case of wordnet that is given two concepts or two words how similar they are even if they are not part of the same synsets.

So, by using synonymy I can only say whether that the two words are synonymous or not, but suppose I want to lose matrix for word similarity or word distance. So, what are the different things in wordnet I can use? So, in wordnet, there are many other relations also defined. So, I can also use the fact that whether there is any other kind of relation between these two words or do the words, do these words share a common hypernym, hyponym and so on also you might have a have a look at the gloss of these two entries whether their glosses are also very similar.

So, I will say two words are similar if they share many different features of meaning and it features can be captured in terms of how many relations are common, how many words are common in their gloss and so on. Now, one thing that we must keep in mind

when we are talking about establishing relation between different entities in wordnet, we are talking about relation between different synsets, and not the words. Because a word might have multiple synsets, and the relation that is there in one sense of word 1 and one sense of another word may not be there in the other synsets.

(Refer Slide Time: 12:59)

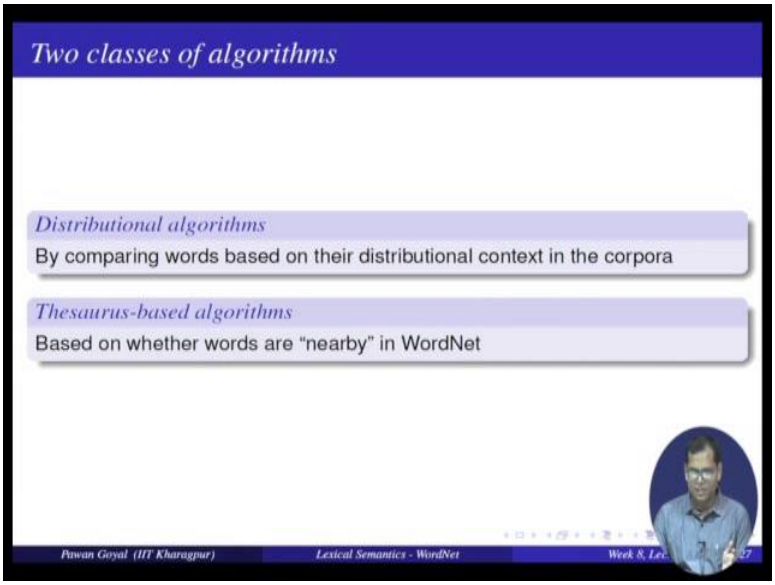


So, for example, is a word bank, this is sense one this is in the sense of you can say in the economics; and sense two is like river bank. And then I have another word like fund and there is s 1 that is for economy. So, I will say that this sense one of bank is connected to this particular sense of fund, but I cannot say that this is connected to fund, this is not true. So, I cannot say that bank is connected to fund saying that will not be correct, what I will say sense one of bank is connected to sense one of fund. So, we will talk about relations between synset of words and not word directly, all though we will try to extend it later to the words. So, instead of saying bank is like fund, I will say something like bank 1 is similar to fund 3, suppose that is the third sense of fund; and bank 2 is similar to slope 5 - the fifth sense of slope and so on.

Now, we will also compute similarity over words and synsets both. So, let us see how do we do that by using the wordnet hierarchy and any other information that we have. So, now this is something that that we have talked about earlier that if I want to find out similarity between words there are two very popular methods one is by using distribution algorithms and that we discussed in detail in the last week. There I can find out the

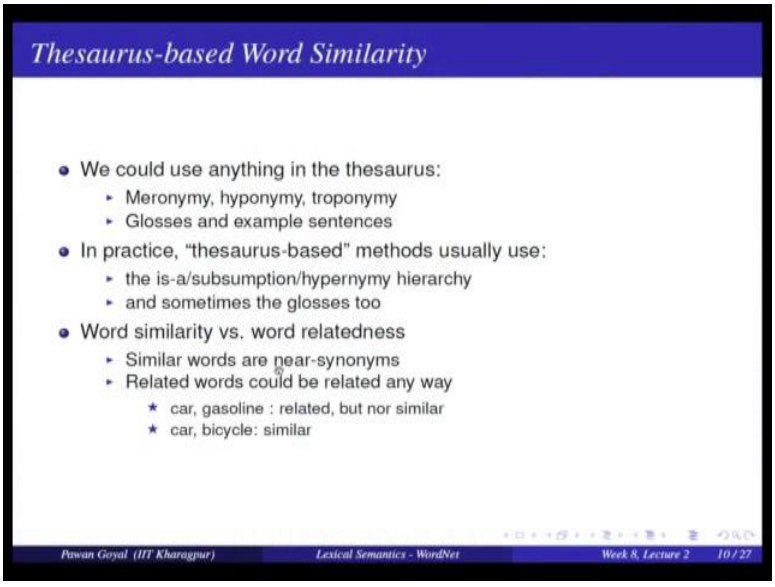
distribution patterns of two words and compare those, but if I want to use a lexical resource like wordnet.

(Refer Slide Time: 14:52)



So, here can I use the idea that some words are more near in the wordnet hierarchy than others. So, if two words are near in the hierarchy, I might say they are similar; if they are very far apart in the hierarchy, they might be different. So, can I use this idea to establish if two words are similar by using the wordnet resource? So, we will now see lot of such methods of doing that.

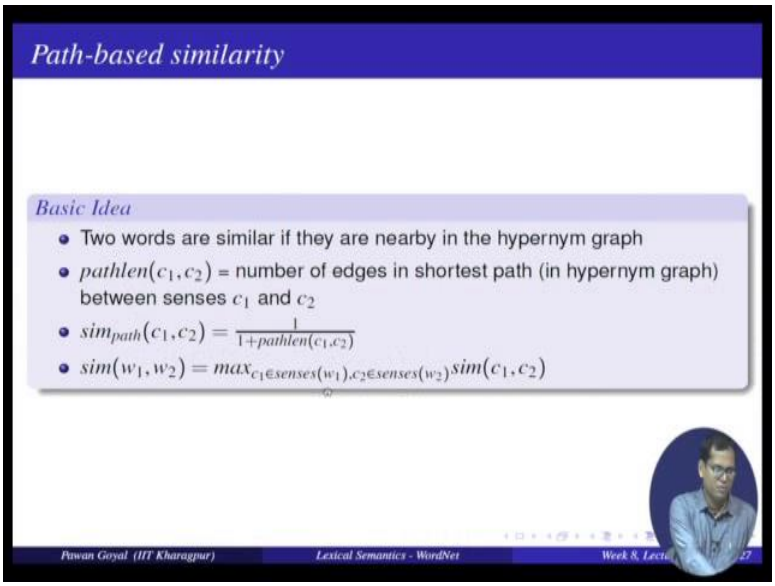
(Refer Slide Time: 15:21)



So, as such I can use any relations like meronymy, hyponymy, troponymy glosses examples, yes, but in particular the thesaurus-based methods that we have they mainly use the EJ hierarchy tree the hyponymy, hyponymy relation tree. Sometimes we will also use the glosses of the words. So, we will start by seeing some examples or some particular methods that try to use the hierarchy - wordnet hierarchy to capture the similarity between the words and then we will also see a particular method that uses the glosses of the different synsets to capture their similarity.

Now, so one thing that you might have seen or have understood by now that by using all these methods we are not capturing the synonymy as such we are not seeing that that we are finding two words that are very, very similar. What you finding is that two words that are related or are used in similar sort of context and topic. So, like if I take car and bicycle, they are quite similar, but car and gasoline they might be related, but not similar. So, by these methods car and gasoline might come closer, but all that means is that they are related they may not be exactly similar.

(Refer Slide Time: 16:47)



So, now coming to the methods to capturing similarity across words, what is the first idea? First idea is to use the path between two words in the hypernym graph. And what can be a simple measure, I will say that two words are similar if the path that connects the two words in the in that hierarchy is small. So, two words are similar if they are nearby in the hypernym graph and to give a formal measure or quantity to that, I can

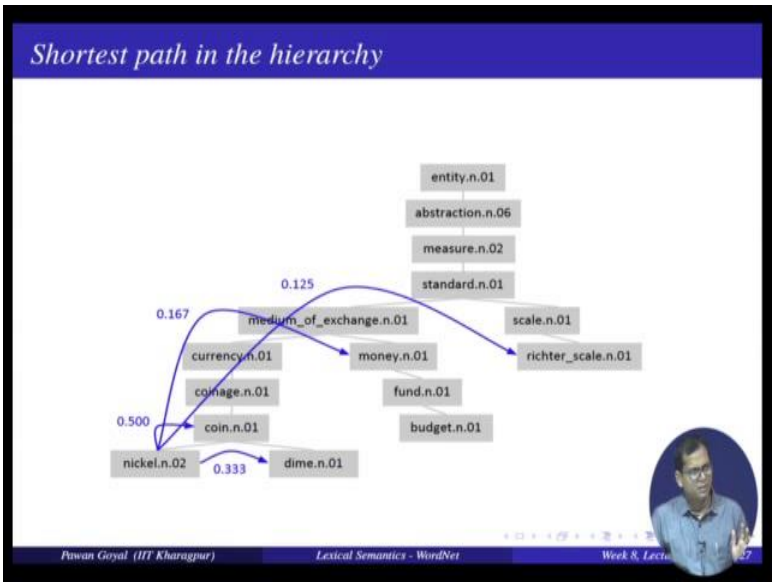
define the path length between two concepts. So, path length between two concepts what will be that that is the number of edges in the shortest path in my graph between synsets c 1 and c 2.

So, what is the length of the shortest path that connects c 1 and c 2 in my whole graph? Now, once have found this path length how do I use that to compare the similarity between these two concepts, so that is path length is large they are less similar; if path length is small they are very, very similar. So, my similarity should be inversely proportion to the path length. So, one measure can be 1 divided by 1 plus path length and this is one simple measure that is used. The path similarity between two concept is 1 divided by 1 plus path length of c 1 and c 2, so that is how we can find out the relation between two synsets.

So, now suppose I want to extend that to find the similarity between two words. So, one way is to find out similarity between all the synsets and take an average, but a more commonly accepted measure is find out similarity between all the possible pairs of synsets between the two words and take the maximum. So, what you mean by that? Suppose I have word 1, word 2 that is sense s 1 1, s 1 2, s 1 3 and this a synsets s 2 1, s 2 2. So, by using this measure, I can find similarity between any two pair s 1 1, s 2 2, what is the similarity. Similarly, I can do for all these pairs.

Now, how do I establish similarity between w 1 and w 2, so that is similarity is the maximum value of similarity between s 1 i and s 2 j. You take any sense for the first word; any sense for the second word whatever the maximum similarity between a pair gives me the maximum similarity or the similarity between these two words that is a very simple way in which I can extend all these ideas to word similarity. So, in whatever we will see the next methods, we will also always talk about similarity between synsets and extended for similarity between words. So, I will say similarity between words w 1 and w 2 is nothing but the maximum similarity between any of the synset of the word 1 and word 2.

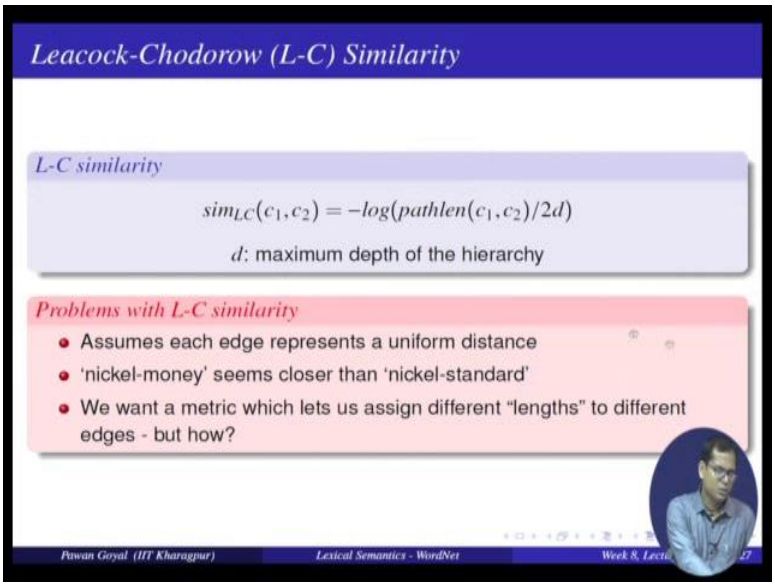
(Refer Slide Time: 20:20)



So, let us take an example that how what will this pathway similarity look like. So, this is my wordnet hyponym graph. So, not all the notes are shown only if very few notes are shown. So, we are starting with entity, abstraction, measure, standard although with entity there will other concepts here. Then you are you are coming to a particular branch with where you have medium of exchange, currency, coinage, coin, nickel and so on.

Now, I want to find out similarity across two different concepts. So, what is similarity between nickel and coin? So, I will say what is the path length; path that connects nickel and coin, what is the length of this path. So, here the length is only 1. So, similarity between these two concepts will be 1 divided by 1 plus 1, so 0.5. And what is the similarity between nickel and dime, it will be 1 divided by 1 plus path length and path length is 2, so it will be 0.33. And if you see nickel in a very different concept like Richter scale, so here you will find similarities 0.125, because the path length is 7. So, like that I can capture the similarity between two concepts by using the path length.

(Refer Slide Time: 21:29)



Now, there is another similarity measure along the same lines. So, this is called L-C similarity. So, what it says, the similarity between two concepts is minus log of path length divided by 2 d. So, this is just a different function over path length. So, earlier we had a function 1 divided by 1 plus x, now they have function minus log x divided by 2 d. And what is d here, d is the maximum depth in my hierarchy. So, starting from the root node, what is the maximum depth of a hierarchy for any of the leaf note? And this helps in that this path length will always be less than or equal to two d and this will give me a similarity between these two concepts.

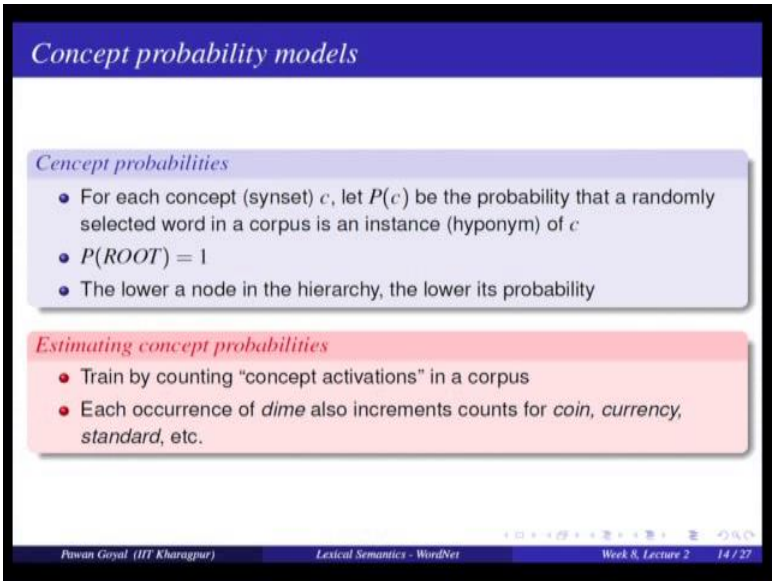
So, what is the problem with this L-C similarity or the previous similarity that we have seen? So, what they are saying for any two concepts find out the path length and the similarity is nothing but a function of path length if path length increases the similarity decreases. But one problem with these approaches is that any two pairs of concepts, if the path length is same, the similarity will be the same irrespective of wherever they occur in the tree.

So, let us just go back to the previous tree. So, what these approaches will say. So, what is similarity between coin and nickel? The path length is 1, similarity is 1 divided by 1 plus 1 that is 0.5. But what would be the similarity between entity and abstraction their path length is also 1, so their similarity will also become 1 divided by 1 plus 1, so that is 0.5, but ideally what would we want do we want the similarity of entity of entity

abstraction to be the same as between coin and nickel. So, if you think about it as we are going down in the hierarchy, we are moving to very, very specific concepts. So, while we are moving this specific concepts, the same path length should amount to a higher similarity noun that it was doing earlier.

So, entity and abstraction this similarity should be much lower than the similarity, this similarity should be very very high, but these methods as of now as of now do not capture this idea. So, they will have this path length to contribute same way as this path length. So, now can we do something different? So that here the similarity becomes high, but here similarity becomes low. So, you want a matrix that let us assign different lengths to different average. So, what is the idea that we will be using?

(Refer Slide Time: 24:35)



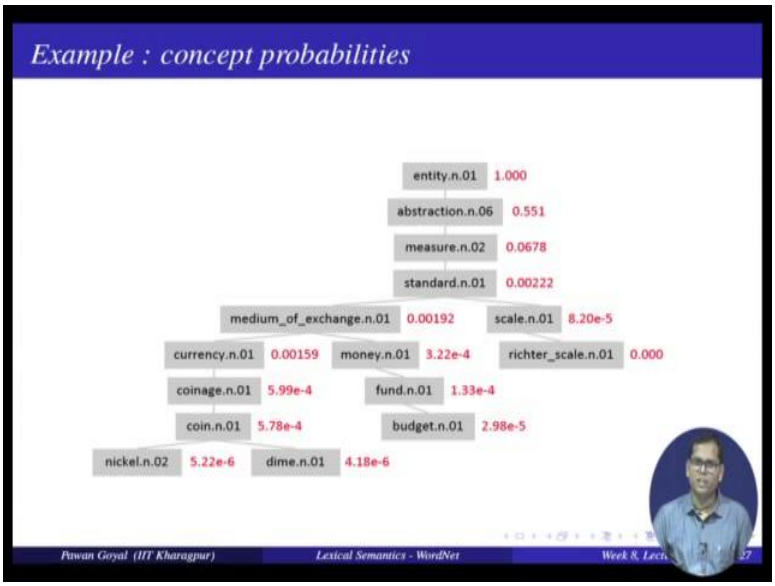
And for that we use the idea of concept probability model. Now, what is this? So, in the wordnet, whatever concepts we are seeing, we will assign them into probability. So, what is the probability with which I see this concept in a corpus? Now, idea would be whatever I am seeing in my corpus is an entity, because it is part of the tree where entity is the root. So, whatever word is in the tree is an entity. So, in the other word, whatever word I am saying in the corpus is an entity, but it may not be an abstraction. So, there will be some words that are abstraction and some words that are not. So, what I will do whenever I encounter a word, I will find out what are all the concepts to which it

contributes, and I will add a count to all these concepts. And finally, I will convert them to probability values.

So, what would happen? The root node will get a probability of 1, because everything I see is an entity, but as you go down these values will keep on decreasing. I will use this idea to convert them into log values and then taking the difference between the two values as the path length and that can converted to finding the similarity between two synsets. So, let us say P c is the probability that are randomly selected word in the corpus is an instance of c. So, what would happen? Probability of root is 1, and a lower a node in the hierarchy, the lower is its probability. And how it can be estimated these probabilities, we count something called concept activations in the corpus. So, what would happen? Whenever encounter a dime, I will also increment a call for coin, currency, standard etcetera.

So, what is the idea? So, I have a wordnet hierarchy tree is starting from root that is my entity and going down. So, what would happen? Suppose this is my word x, whenever I encounter this word x in my corpus, I increment its count by 1, and all its parents, because whenever I am encounting, I am also encounting this concept and so on. So, what would happen? Whatever I encounter I always add 1 to the root, but only to its parents. So, when I do that root will have, so all the counts will be added to root, and I can find the probability by dividing everything by the count of root.

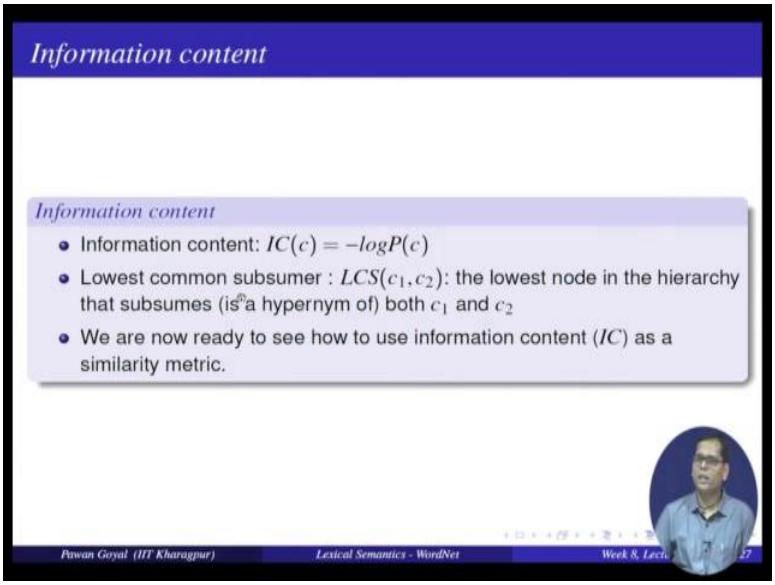
(Refer Slide Time: 27:34)



So, suppose I do that on my corpus. So, this is one example. So, here you can see there are 1.9 million roughly instances overall because entity has been always appended with one, but there the numbers are different. So, diamond and nickel has only 8 and 10; coin has 1,108. So, you see, we are not showing the whole tree that is why there will some other branches of coin also that are not here. Now, once I have got these counts, I know opt in the probability values by dividing everything by this number. So, this will be my concept probability.

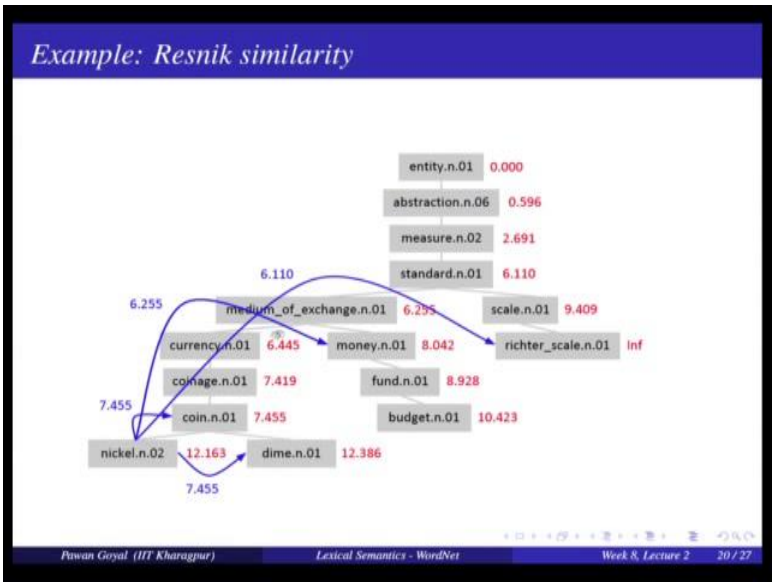
Now, how do I use this concept probability to define or define my edge weights? So, what I will do? I will convert them in some information value. What is the information content of each concept and the information content can be directly obtained by the probability values by using minus log probability. Idea is that if the probability of something is very high, it does not have much information, but the probability is low it contains a lot of information.

(Refer Slide Time: 28:47)



So, I use the information content of a concept as a minus logarithm of the probability of that concept. So, I can do that for all the concepts, and I can also define what is my lowest common subsumer that is if I take two nodes or two concepts c 1 and c 2, the lowest common subsumer is the lowest node in the hierarchy that subsumes both the nodes or what is the lowest node in the in the tree where these two concepts meet.

(Refer Slide Time: 29:31)

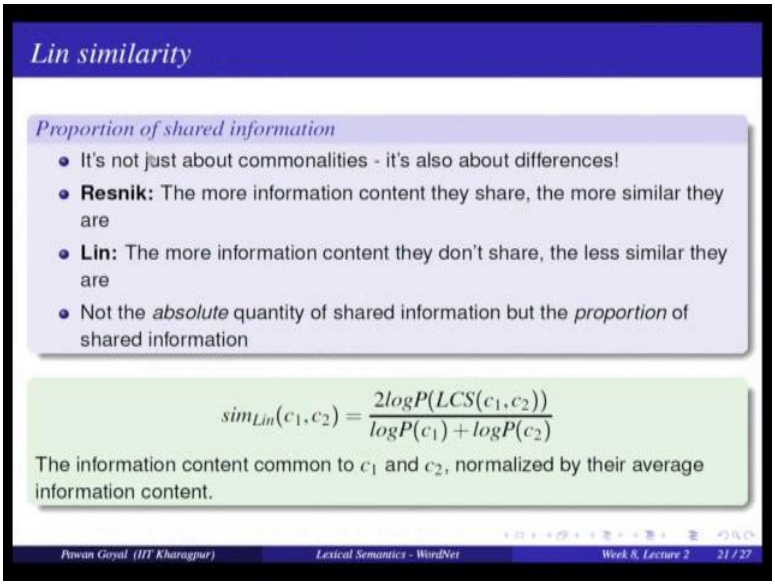


And now we will see how you can use that computing similarity between concepts. So, these by using from the probability, if I take minus log probability, the other different numbers I will get. So, minus log 1 becomes 0, if entity is 0 then 0.5 minus 5 and so on. Now, what is something that you are seeing here? So, as you are going down these numbers are increasing. So, one simple way of capturing similarity between two words is by seeing what is the lowest common subsumer and what is the information content of that. So, nickel and dime, what is the similarity, path length is 2, but the lowest common subsumer that is coin has information content of 7.455. On the other hand, if I take these two concepts medium of exchange and scale, their common lowest subsumer standard has an information content of 6.11.

So, immediately you can see this can be said to have a higher similarity than this pair. So, what are the formal method by which we can capture this? So, one is Resnik similarity. So, it says how similar two words are depends on how much they have in common, so that is find out their lowest common subsumer and find out their information content of that and that will denote the similarity between these two concepts. So, it measures the commonality by the information content of the lowest common subsumer. So, like here nickel and dime, the similarity would be the information content of the L-C s that is coin. So, similarity between them is 7.455. Now, nickel and money similarity would be the information content of their L-C s that is medium of exchange, so that would be 6.255 and so on.

So, now, immediately you can see that as you keep on going up in the hierarchy, the similarity will be decreasing. So, it will be highest when you have took single leaf nodes, but if you keep on going up the similarity will decrease. So, this capturing what you wanted to do, but still there is one problem here. So, can you find out what is the problem? So, one problem here is if I find the similarity between coinage and money, this would be same as similarity between coinage and budget, yes, because their L-C s is the same. So, here what is being captured is that how much information they share, but what is not being captured is how much information they do not share or how much they are different. So, we have a different measure for capturing how much information they do not share.

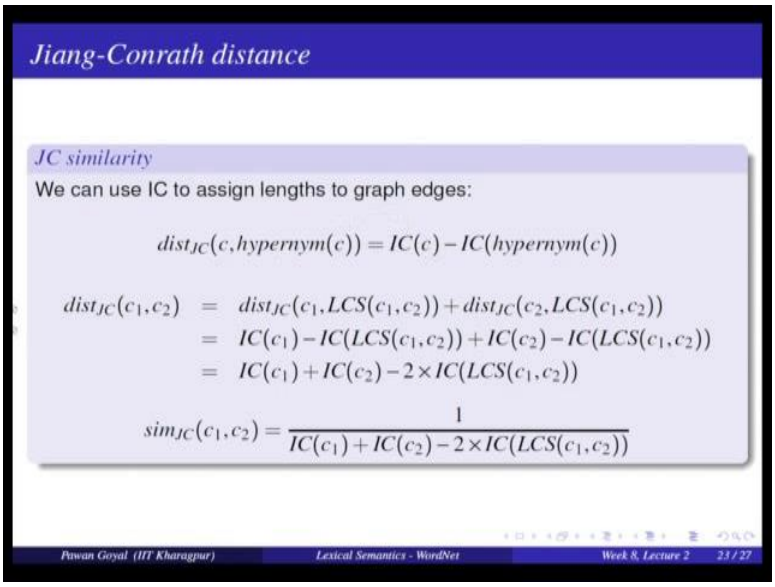
(Refer Slide Time: 32:23)



And this is called Lin-similarity. So, it says that this measure is not about just commonalities we also have to capture the differences. So, the more information they share, the more similar they are, but the more information they do not share the less similar they should be. So, accordingly the Lin-similarity between two concepts is defined as two times logarithm of or two times information content of the L-Cs divide by information content of the concept plus information content of the concepts. So, while doing that what would happen? Now if I take the similarity between coinage and money this would be 6.255 times to divide by 7.419 plus 8.042.

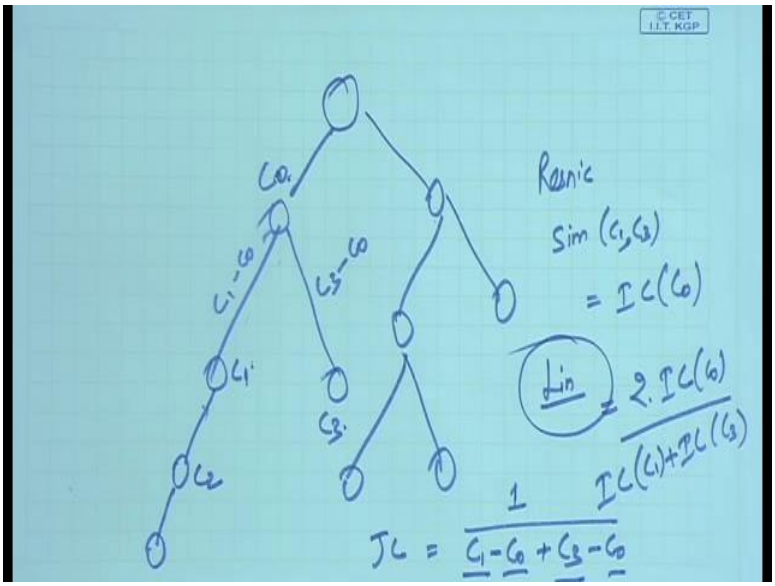
And if I take the similarity between the coinage and budget it will have a term of 10.423 in the denominator instead of 8.0242. So, immediately this similarity will become lower than this similarity. So, Lin-similarity is a much more well accepted measure than the previous measure that we have seen the Resnik similarity.

(Refer Slide Time: 33:41)



There are some variations here. So, for example, the JC similarity, what they say in JC similarity find out or give a value to each h that is a distance between two concepts. And this would be the information content of the concept minus the information content of the hyponym, and I can define the distance between two concepts as their distance. So, how do I go from one concept to its L-C s, and second concept to its L-C s and I just add the distances. And I compute similarity by taking the inverse of this distance.

(Refer Slide Time: 34:27)



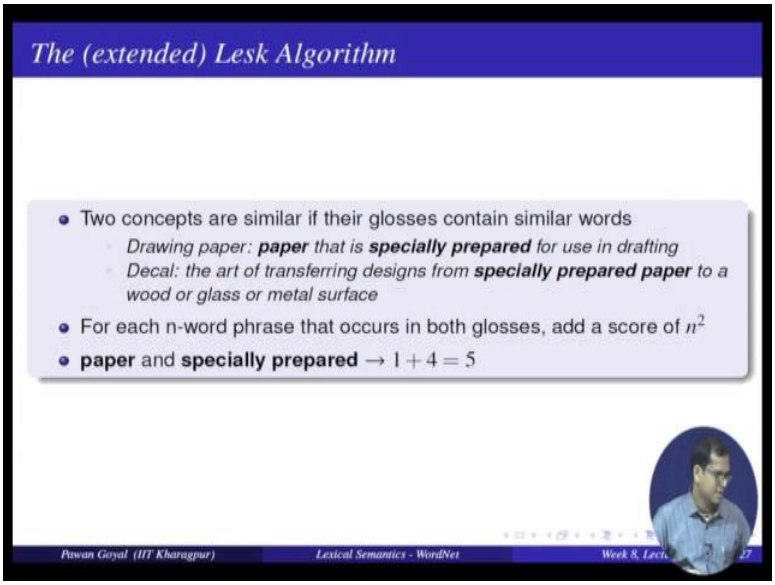
So, what am I doing here I am having different nodes in my hierarchy yes and suppose that you have already found, what is their information content. So, what do you do in the case of Resnik similarity, in Resnik similarity the c 1, c 2, c 3 and this is c 0; in Resnik similarity the similarity of c 1 and c 3 is nothing but the information content of c 0. So, as per Resnik similarity of c 1, c 3 is IC c 0. As per Lin-similarity, this would be two times information content of c 0 divide by information content of c 1 plus information content of c 3.

Now, in JC similarity, what you would do? You would count you would find out the distance this is distance is nothing but c 1 minus c 0, and this distance is c 3 minus c 0. So, for JC similarity it will be c 1 minus c 0 plus c 3 minus c 0 is the distance between these two concepts. And the similarity between the 1 divided by that. And that is what is written here? Information content of c 1 plus information content of c 2 minus 2 times information content of the L-C f of the 2 and that is what you can say here IC of c 1, IC of c 3 minus two times IC of LCs. So, I hope by this example you understand what is the difference between Resnik similarity, Lin similarity and JC similarity; and among these Lin-similarity is very, very popular.

So, I hope it is clear that how do you apply these three different similarity measures. So, here you have example that if we use the JC similarity, what is the different values that you will obtain among different concepts. So, I will encourage that you will you try and

find out that say suppose between nickel and Richter scale or between nickel and coin can you obtain the same values by applying the formulas.

(Refer Slide Time: 37:03)

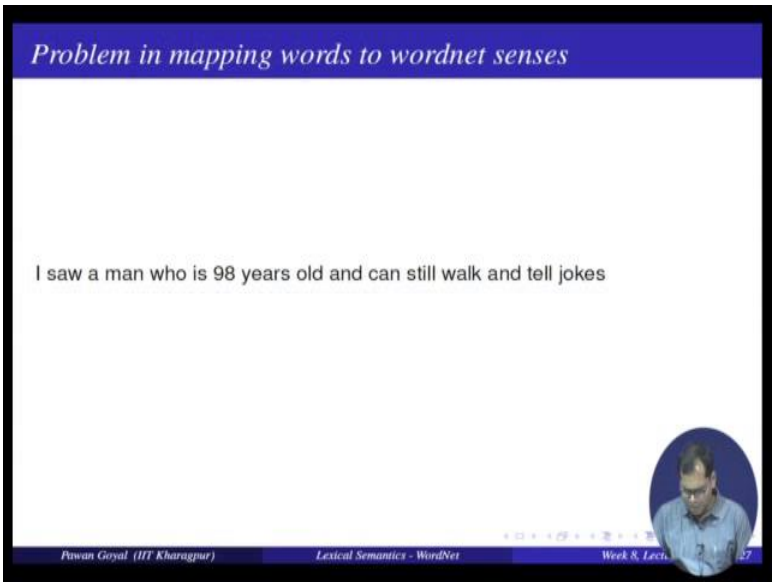


So, now, I will also talk about briefly the other approach for computing similarity between two different concepts in wordnet. So, till now we have only used the hierarchy tree in wordnet, and I am saying two words are similar, if they are nearby in the hierarchy tree or some other formulation and that we have seen some examples. Now, suppose I want to use their glosses, the way the different concepts are defined in wordnet for comparing similarity. So, these are very simple algorithm called Lesk algorithm, also have a extended Lesk version that is used for that. And what is the idea two concepts are similar if their glosses contain similar word. So I have the word drawing paper defined as paper that is specially prepared for using drafting; and decal - the art of transferring designs from specially prepared paper to a wood or glass or metal surface.

I want to find out how similar they are. So, what I will do? I will see how many words are common there. So, as such you are seeing that three words that are common paper, specially and prepared that occur in both the glosses. So, what algorithm does is that it counts how many n-grams are common, so n-gram in the sense of one unigram, bigrams, and trigrams and so on. So, you will see that this bigram is specially prepared is common to both the glosses and the unigram page is common. So, whenever an n-gram is common, it adds a score of n square, so that you have is given for a commonality of

bigram trigram and so on. So, in this case, what would be the similarity one bigram is common, so two square and one unigram is common and one square. So, similarity would be 2 square plus 1 square – 5, 1 plus 4 - 5 that is the similarity of using Lesk algorithm.

(Refer Slide Time: 39:04)



So, we have talked about what are the different relations we can capture using wordnet, and we have also seen how we can find out similarity of across two words, and that looks like very simple method once the wordnet tree is given. And you might also wonder this might be a better method of capturing similarity than distribution similarity, because I have a manually created this orders, I know which of the words occur where in the tree I can simply use the distance or measure to find out how similar they ar. But there is one inherent problem in using wordnet for any of the task. So, can you think of what is the problem?

So, let me give you the hint if you know if you want to capture similarity across synsets wordnet is very good, it can capture the similarity between the synsets very nicely but if you want to catch similarity between words that is very difficult. Now, when we encounter natural language, we will only encounter the words; and when we see the words, we do not know what are the synsets that are being used. So, I cannot directly apply wordnet there, because I do not know the sense, I can only do an approximation where I can find out ok, I am assuming this word corresponds to or I am applying the

methods I used for synsets for the words. This would be an approximation. And it works sometimes, but does not work some other times.

So, to be able to apply wordnet, to be able to use wordnet, one important problem that we have to deal with is I need to find out if a word is used in a particular sentence what is the wordnet sense that has been used for that and that is difficult problem that we will try to address. But this problem is very, very common and we will just take a very simple example for that. So, you see very easy sentence, I saw a man who is 98 years old and can still walk and tell jokes. Yes, this is a very simple sentence. Now, for the simple sentence, what do you think, are there many different synsets of this word, or this whole sentence, or there is one interpretation. So, when we hear this term this sentence we have only one interpretation in mind, but in wordnet what are different synsets of individual words.

(Refer Slide Time: 41:44)

So, let us see. So, if I go to wordnet, the word saw has 25 senses, man has 11 senses, years has 4, jokes has 4, tell has 8 and so on. So, now, if you combine all these together, they are has a 67 million plus senses that are possible sentence this might look like a very extreme case, but you might have examples where they are multiple divisions for the same sentence, and different words can have can occur in multiple synsets. So, my problem is if there are so many synsets, how do I find out what exact synset wordnet is being used, and that is where we talk about the problem of word sense disambiguation

among the many possibilities disambiguate the particular sense of the word, and that we will start in the next lecture.

Thank you.