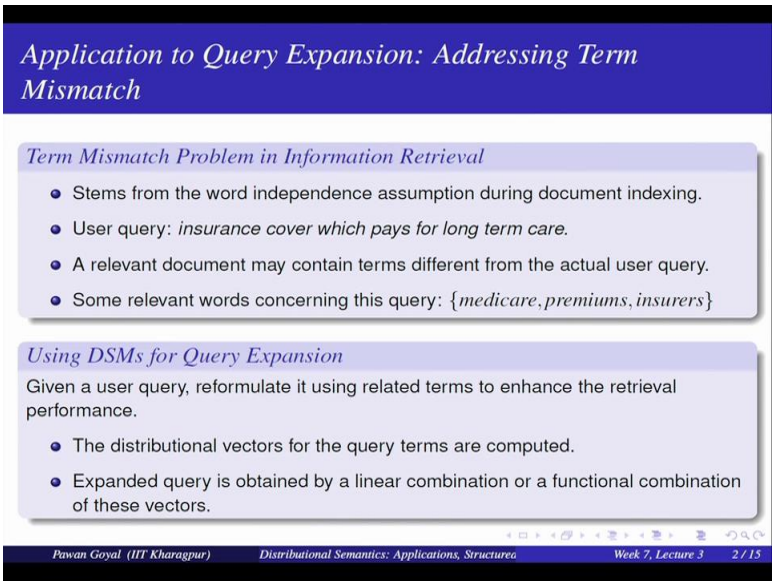
Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 33
Distributional Semantics - Applications, Structured Models

Hello everyone, welcome to the 3rd lecture of this week. So, we are talking about distribution semantics and we had already discussed what is a generic framework of distribution semantics, how do you build models by taking different sorts of targets and context and how do you fill up all the antigen of matrix. Now in this lecture, we will talk about some of the some applications for distribution semantic models, also how do I compare similarity across two words and then we will move towards some a structured model of distribution semantics.

(Refer Slide Time: 00:56)



Starting with some application, so let us talk about this problem of term mismatch that we see in the case of information retrieval so, what is the problem? So information retrieval, the user is giving a query that is in terms of certain words that he feels are describing his information in it and so the system is trying to match this set of keywords to all the documents that are there in the depositary and by doing this matching they are trying to find out what are the potential document that can be returned to the user.

Now, what is one problem here? So, it might happen that the particular concept that the user has in mind; the user expressing by using certain words, but the documents have the same concept using a very very different word. So, they are, the words are similar on the semantic label, but on the surface label they are 2 different keywords or 2 different words. So, this is called a term mismatch problem and one of the key issues in information retrieval is how do we solve this term mismatch problem? How do I find out that the user typed this particular term? But he might also be interested in the other term that is semantically similar, but on the surface form it is different.

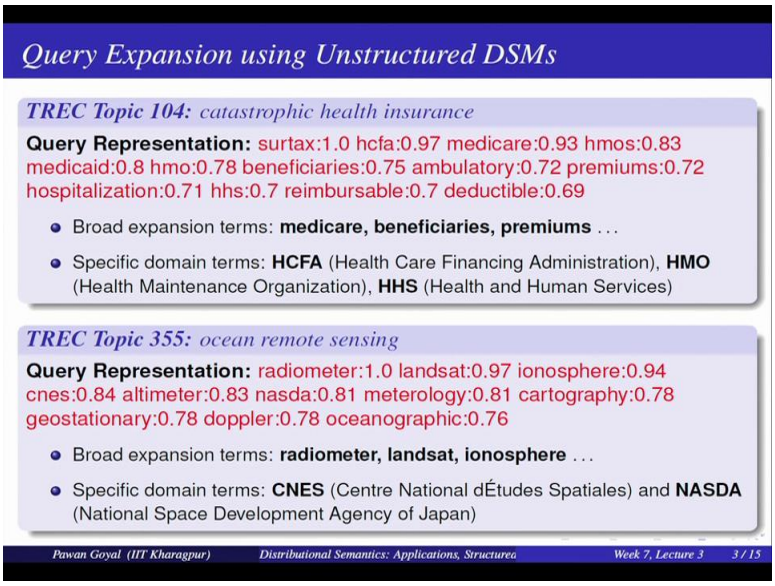
Let us see an example. So, suppose there is a user query insurance cover which pays for long term care and it might happen that the relevant document that contains the answer to the user query may have words like Medicare, Premiums, Insurers, etcetera that are not directly provided in the user query. Now, one question is; how do I find out that these documents that contain some similar words are also relevant to this user query.

So, for this task, one can use issue semantic models for query expansion. So, query expansion is one of the techniques that is used for this term mismatch problem. So, idea is that I the user has given a query with certain words can I try to find out some other words that are semantically similar to the user query words? If I find out some words then I append those words to the user query and this is called the expended query expansion process and now I give this expended query to my search engine and then retrieve the documents.

Now, what we will see in brief, can we use our distribution semantic models for this problem of query expansion. So what is the idea? So, once user gives a query, we reformulate it using the terms is a relevant or that are related to the terms that are already there in the query, find out some related terms and try to use that to improve the retrieval performance. So, how do we find out the related terms? So firstly, I have the query terms that the user is giving, find out what are the distributional vectors and take a function combination of all the distributional vectors and obtaine the expanded query.

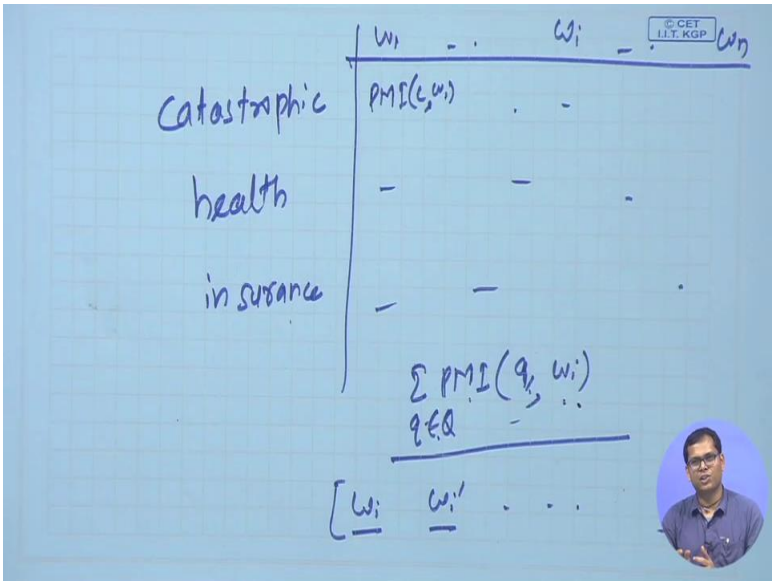
Suppose that the query has 3 terms find out what are the related terms to all this 3 terms in the distributional sense and make a linear combination of all such possibilities and give the expanded query to the user.

(Refer Slide Time: 04:14)



How will it look like? So, suppose the user gives a query like catastrophic health insurance. So, what you are doing? So, in your distribution vector, suppose you are finding out, what are the terms that are related to each of the 3 terms catastrophic health and insurance?

(Refer Slide Time: 04:34)



What will happen? So, user has given these 3 terms and suppose, you are using words at con as context. So, the different words were there w i w n and for each word. So, you are computing, what is the co occurrence? So, it can be say PMI value, what is the PMI

between catastrophic and w 1. So, like that, you are computing for all the words. So, now, what you might do? You might say, what are the words that are associated with all the 3 words? So that are having high PMI values. So, one way is you just add all the PMI values PMI q w i for all words q in my query.

For each word w, why I complete a score like that what is the PMI of that query, one of the query word with this word w i and add your all the query words and then I can sort it and maybe I can normalize it with respect to the highest term. So, what will happen at the end of that if I can sort these course and find out what are the words w i, w i prime etcetera that are very much related to the query terms and all these we have already covered how do we computed this course, once we have this course I can compute this symbol formula and find out what are the terms in a decreasing order.

This one part way, but you can have any other sort of functional combination. So, this is a simple addition, but you can have multiplication and other sort of function combinations also. So, suppose we do that. So, what happens to the actual query? So, this is my ac actual query catastrophic health insurance and if I try to use this technique to find out some other related terms I get terms like this. So, I have terms like surtax, h c f a, Medicare and hmos, Medicaid, hmo, beneficiaries, premiums and so on. Now so these terms are taken by from a news corpus that is US news corpus. So, they are certain terms that are relevant to the US medical system. So, what are the; so what you are seeing in the expected terms? So some terms are like broad explanation terms so; that means, if you look at may be some (Refer Time: 07:03) or some other thesaurus you can find out such kind of similar terms.

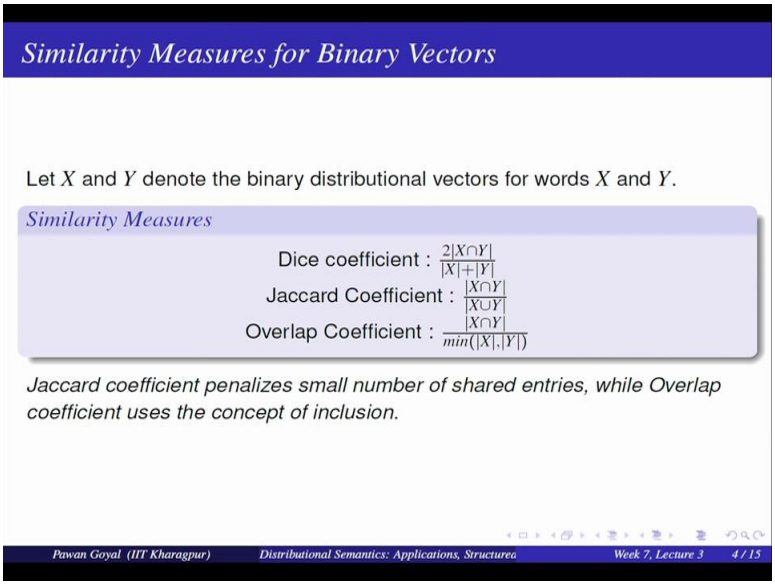
You will say Medicare, beneficiaries, premiums, are some broad explanation terms. So, they are ready to the query. In addition, you also get some very very specific domain term like here HCFA that is US domain term for health care financing administration, similarly HMO for health maintenance organization, HHS. So, these are very very domain specific terms. So, if you know that you query is coming from a certain domain, this approach can be very very helpful to also find out terms that can be used to explain the query and not also a domain specific.

Let us take another example. So, these are so, I am showing here some TREC topics. So, TREC is a text t o conference. So, they organize various competitions for information

retrieval. So, they are actual queries from the TREC data. So, you have a query like ocean remote sensing and when you use this method, you can find some other terms like radiometer, landsat, ionosphere, cnes, altimeter, nasda, meteorology and so on and again as we seen saw in the earlier query, there are some broad explanation terms like radiometer, landsat and ionosphere that are connected to the query and they are some domain specific terms like cnes and nasda here.

This is one very nice application of distribution semantic models you have some existing query you want to match it to some document, why do not you expand this query by using some related terms and this is the idea. And this we can use in general for any rid matching task you are having do different text data and you are trying to match these. So, try to find out if there are somehow related on the semantic label using distribution semantic models and use this idea to find out if they are similar or not.

(Refer Slide Time: 09:04)

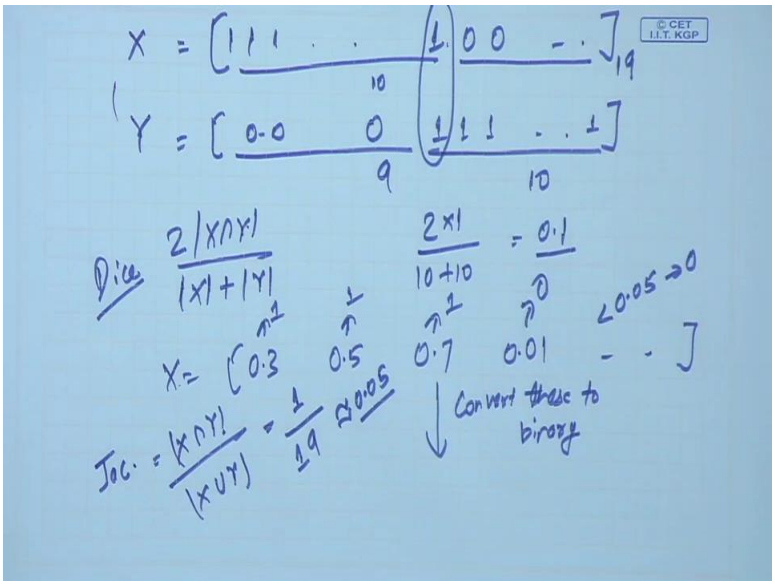


Now, so once we have computed the distributional semantic model so, I have the vectors for different words. So, different target words, I have the vectors. Now, how do I compute the similarity between different words? So, it depends on what is your representation that you are using for computing the semantics or the what is the vector representation if it is the binary vector you will use different sort of similarity methods than if it is a real valued vectors or if it is a probability distribution you will use a different sort of similarity matrix.

So, as is the framework allows you to use any of these vector representations. So, let us see if you use, if you are using any of these what kind of similarity methods you can use to compute similarity between 2 words. So, let us say my. So, have words X and Y and they are denoted using some binary vectors. So, as such any sort of distribution you can also convert into binary vectors.

How do I compare, how do I find similarity between 2 word 2 words where the representation is binary vectors? So, for binary vectors, we have some standard methods like using dice coefficient. So, what is that? 2 times intersection of X and Y divided by length of X plus length of Y and what do I mean by this. So, X and Y are binary vectors.

(Refer Slide Time: 10:32)



Suppose X and Y are binary vectors and let us say the size of the my vector; the dimension are say since it is a 19 dimensions and my X is 1 1 1 1 in the first 10 dimension and 0 0 in the rest 9 dimensions. On the other hand Y is 0 in the first 9 dimension and 1 in the last 10 dimensions.

Now, I have 2 vectors - X and Y and I want to compute the similarity between these 2 vectors, suppose I am using dice coefficient. So, the formula is 2 times X intersection Y divided by length X plus length Y. So, this is the length of X intersection Y, now what is X intersection Y? So, that is only 1 element. So, X intersection Y has 1 element that is 1. So, this will be simply 1, 2 times 1 length of X, how many 1s are there.

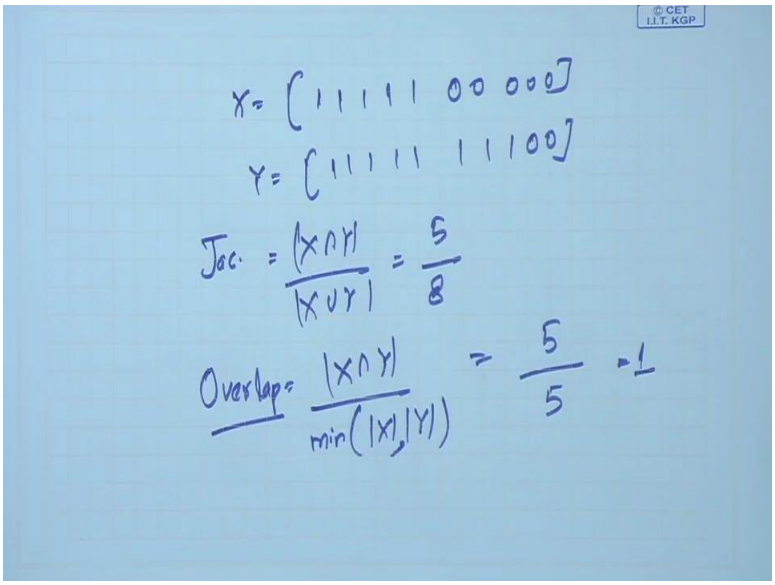
We will not take the length of the vector because otherwise it will be the same for everything, everything will have the same size. So, length of X means how many entries are 1. So, here it will be 10 for Y, also it is 10 10 plus 10. So, this will be 0.1. So, now, this is if you have the binary values, suppose you do not have the binary values so, suppose my values are like my X could be 0.3, 0.5, 0.7, 0.01 and so on. So, if you want to use these measures. So, 1 simple way could be convert them to binary and now converting would mean you would put a threshold that if the value is below this threshold then you put it to 0 above this threshold you will put it to 1. So, suppose here threshold is 0.05. So, what you will do? This will go to 0, this will go to 1, this will go to 1, this will go to 1 and here you will check whether less than 0.05 then go to 0 if greater than then go to 1 like that. So, you can convert any such vector into binary representation and then use this dice coefficient, now are there some other methods of computing similarity also?

So, we have Jaccard coefficient and overlap coefficient. So, Jaccard coefficient is X intersection Y divided by X union Y. Now what is the difference between Jaccard and dice, in what scenario Jaccard will gave give a different value than dice coefficient. So, let us try out the same example. So, I am using the Jaccard as X intersection Y divided by X union y. So, in this case what is X intersection Y this will remain as one only one entry is one and what is X union Y now X union Y will contain all the elements this will be 19. So, what you are seeing for the same 2 vectors dice give the value of point one and Jaccard gives 1 by 19 that is closely 0.05. So, these give much smaller value.

Why is that? So, you are seeing here if there are very small number of shared entries Jaccard further penalizes. So, that is what you are seeing here there are very only 1 entry common among 20. So, Jaccard is giving further penalty.

Now, what will overlap do? So, overlap is X intersection Y divided by minimum of X Y. So, can you think of a scenario where overlap it can becomes 1, but say Jaccard will not become 1. So, this will happen when 1 of X and Y is completely included in the other.

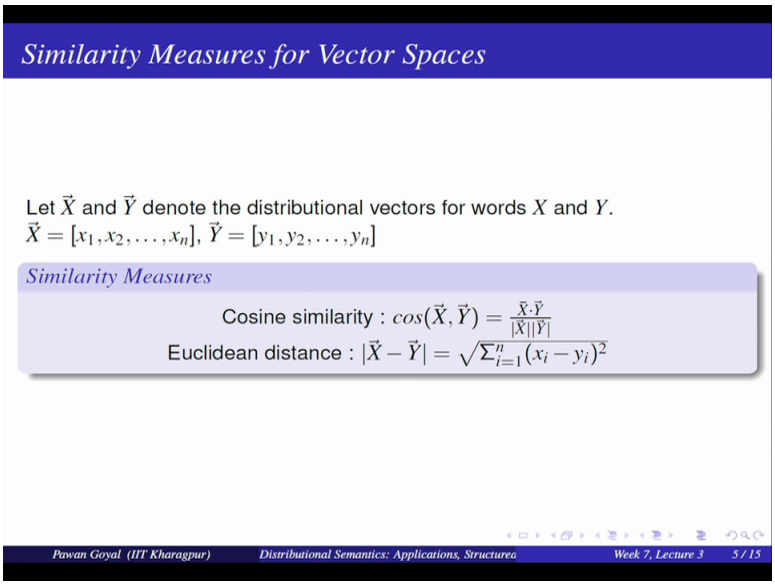
(Refer Slide Time: 14:54)



Let us say, let us take a different case suppose my X is 1 1 1 1 1 0 0 0 0 0 and Y is 1 1 1 1 1 1 1 1 0 0, so in this case, what would be the Jaccard? It will be X intersection Y divided by X union Y. So, this will be 5 divided by 8 and what will be overlap? This will be X intersection Y divided by minimum of X and Y. So, this is again 5 and what you mean of X and Y again 5. This will become 1. So, that is if 1 of the vector is completely subsumed by another vector overlap is 1 and that will not happen in Jaccard dice coefficient and similarly if there are a small number of shared entries, Jaccard will give us smaller value. That means, depending on what kind of similarity want to use, you can have either dice Jaccard overlap, suppose in your task, you want to find out if 1 of the words is completely subsumed by another, you will use overlap and not Jaccard, but if you want to see some other criteria you can choose 1 of the 3 methods.

Here so what you have seen? Jaccard coefficient penalizes small number of shared entries while overlap coefficient uses the concept of inclusion where the one of the entries completely included in the other one, now this is a (Refer Time:16:29) vectors of binary vectors.

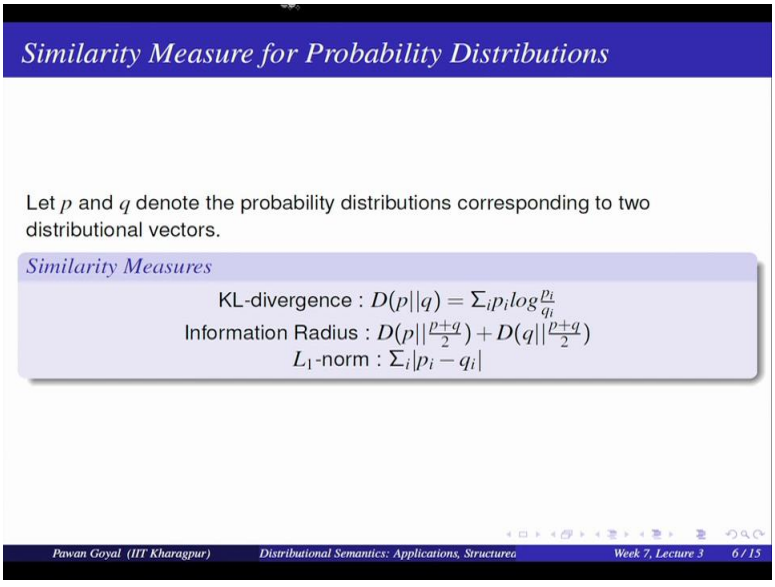
(Refer Slide Time: 16:31)



Suppose they are real number values, so like X and Y so, say the n number of real number values. So, then you can use simple cosine similarity or Euclidean distance.

You can use cosine similarity and Euclidean distance, now what is the difference between the two? If my vectors are not normalized cosine similarity Euclidean distance are different, but if they are normalized, they will be the same and they will give the same sort of ranking and that you can do a very simple exercise, if they are normalized, they will give me the same sort of ranking between the similarity of vectors.

(Refer Slide Time: 17:14)

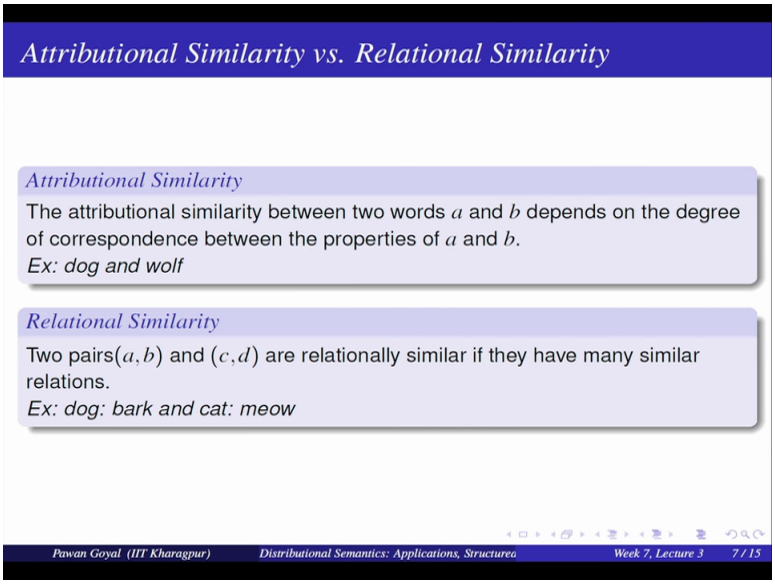


Now, on the other hand, suppose my distributions are probability distributions. So, I am denoting different vector size probability distribution, my space of the context vectors.

Then how do I compute the similarity? I will use different measures that are used for computing similarity or distance in the case of probability distribution. So, what are the common measures? So, you can use KL. KL divergence is so that is sigma i p i log p i by q i. So, this so KL divergence is asymmetric. So, if you use divergence between p and q and q and q and p; they will come out, they may come out to be different. So, that is why there is a symmetric divergence also that is information radius. So, p and p plus q by 2 and q and q plus p; p by 2, find the KL divergence between these and add this that is information radius and also you can use is a very simple formula like L 1 norm. So, you have p i q i find out the LL 1 norm.

What is the difference between p i minus q i sum over all i. So, that is you have different sort of representation and you can u use different similarity value similarity matrix.

(Refer Slide Time: 18:24)

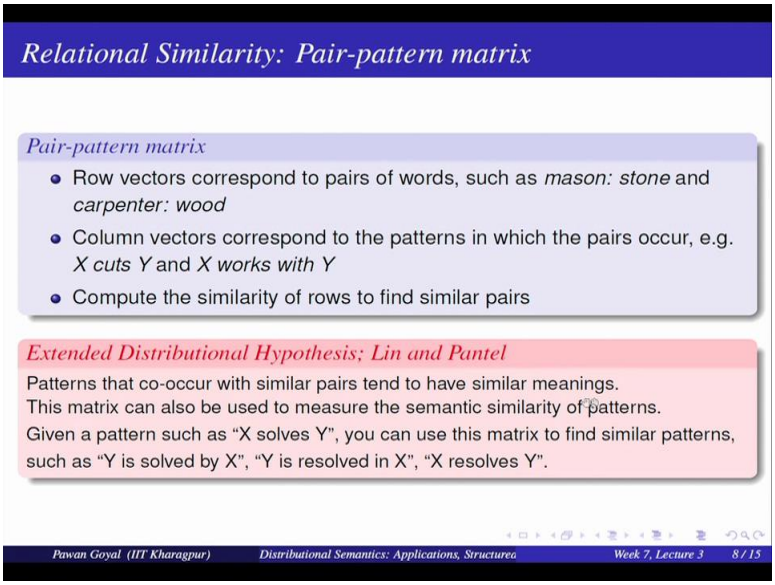


Now, let us talk about what are the different; what are some other sorts of distribution semantic models that all that also try to use some specific instructions in the sentences and we try to motivate them using this example. So, that is, what is the difference between an attributional similarity task and a relational similarity task? So, what is attributional similarity? So, that is I am given 2 words like dog and wolf and I want to find out how similar they are. So, similarity between dog and wolf will depend on how

much their attributes are similar that that is why it is called attributional similarity and by using distributional semantics how do you capture that? What are the other words they are co occurring with? Are they co occurring with similar sort of words? If they are co occurring with similar sort of words, they will have a high attribution similarity and what is relation similarity? So, that is slightly different.

That is now I am talking about pairs. So, that is I have 2 pairs a b and c d are they relationally similar that is how many co similar relations that they have. So, example would be like 1 pair is dog and bark second is cat and meow. So, now, dog and bark do they share similar sort of relation as cat and meow. So, this is simple different type of task and so this that is why first one is called attribution similarity. How much the attribution similar among the 2 words? Second is called relation similarity. I have the pair the relation between this pair does that hold also for the other pair and this also gives tries to many analogy testing task. So, a is to b as c is to what? So, we will see how do we extend our distributional seman similarity or semantics models to also capture all these cases.

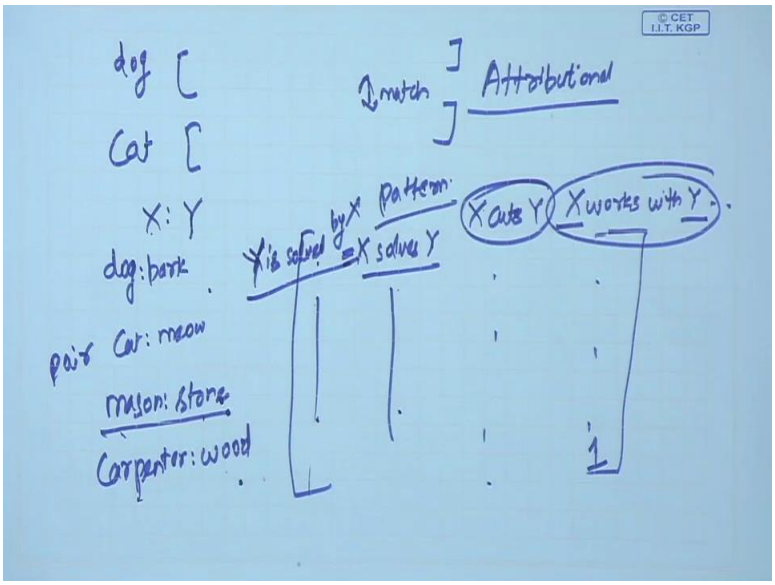
(Refer Slide Time: 20:15)



Now, for that we will talk about a different sort of matrix and this will be called pair pattern matrix. So, till now, you are talking about target context matrix. So, let us see if you can use a similar idea for building a pair pattern matrix. So, what do I mean by this?

Here the row vectors will correspond to various pairs of words like mason stone and carpenter wood and the column would be various patterns in that in the sentences these words occur with. So, like here X cuts Y, X works with Y etcetera and then you compute the similarity of rows to find similar pairs of words. So, now so, what do I mean by this?

(Refer Slide Time: 21:01)



Till Now, what we were doing we had words like dog cat and I was finding out these representation in various contexts. So, I am finding out vectors for dog vector for cat and trying to match these and this was my simple attributional similarity. So, now, what am I doing I am trying to compute a similarity between pairs. So, now, my rows are different pairs. So, like the pairs can be say dog bark cat meow or it can be like here mason stone carpenter wood. So, like that they can be various pairs now these are my rows now what do the columns denote now columns would be various patterns. So, pattern could be like X cuts Y X works with Y and so on, these are various patterns now what are X and Y you can think of this as my patterns X Y sorry pairs X Y.

Now, what ha how will I fill this matrix simple way would be I have X and Y X can take value like mason and Y can take stone for a given pair now I will go through my coppers and see what are the various patterns in which these words co occur with. So, suppose there is a sentence mason works with or mason works with stone or carpenter works with wood let us take a sentence carpenter works with wood. So, here, I have pattern X works with wood X fits for carpenter Y fits for wood. So, I will say there is a plus 1 here,

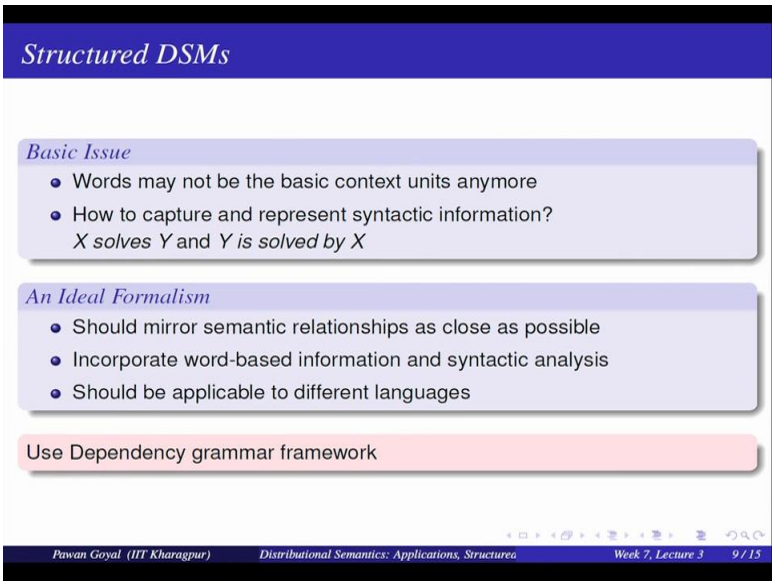
similarly there will be all sorts of pattern here in which X and Y can occur and I will fill which pair occurs with what patterns. So, now, this is my pair pattern matrix pair pattern matrix and then I can once I have this matrix I can compute which pairs occur in similar patterns and then they are called relationally similar they have similar relation at the other pair.

Then, so we can talk about the extended distributional hypothesis this that much given by Lin and Pantel. So, what is the idea patterns that co occur with similar pairs tend to have similar meanings. So, here we are talking about in terms of our columns. So, here suppose what they are saying if a pattern p one and pattern P 2 if they co occur with similar sort of pairs then they are giving similar sort of meanings.

And therefore, I can use this matrix to compute semantic similarity of patterns also. So, I can find out the semantic similarity of the pairs also the patterns. So, suppose I am given a pattern like X solves Y and I want to find out what are other similar patterns as X solves Y, how will I do that? I will first enumerate all the possible patterns that can occur with X and Y then I find out all the possible pairs how many times they co occur with various patterns. So, I will fill this matrix and then I will find out for this pattern like X solves Y what are some other patterns that us that are having similar pairs as X solves Y.

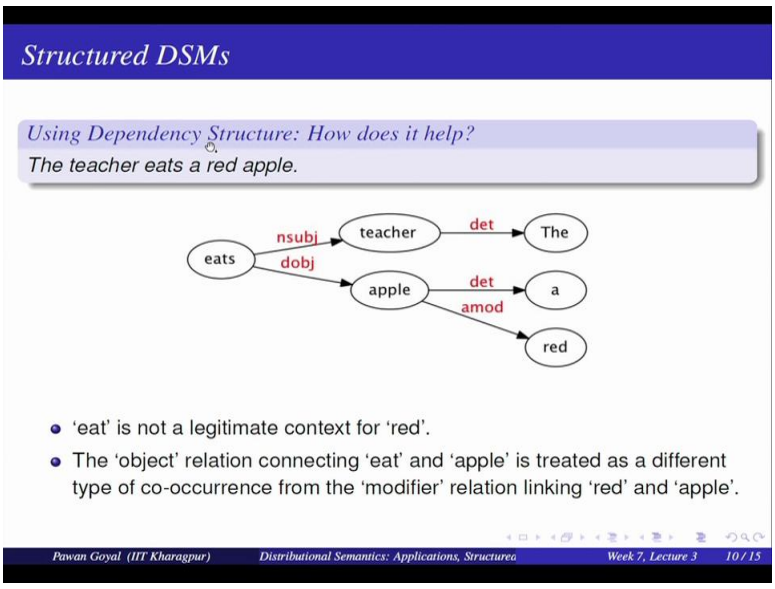
And what are patterns like X is solved by Y sorry, Y is solved by X, suppose it occurs with similar pairs as X solves Y, I will say that this and this are same and that will be the idea it is like here Y is solved by X Y is resolved in X and X resolves Y. So, what do you find all these patterns occur with similar pairs? So, they can be also called similar.

(Refer Slide Time: 25:17)



Now, what is 1 thing? So, for dealing with this pair pattern matrix words will not be my basic context units. So, how do I capture and represent this sort of information like X solves Y and Y resolved by Y X, how do I capture this information? So, for that I will need a formalism that can capture semantic relations and also various syntactic information can be captured and for that we will go back to our dependency based formalism to capture this kind of information what is the idea.

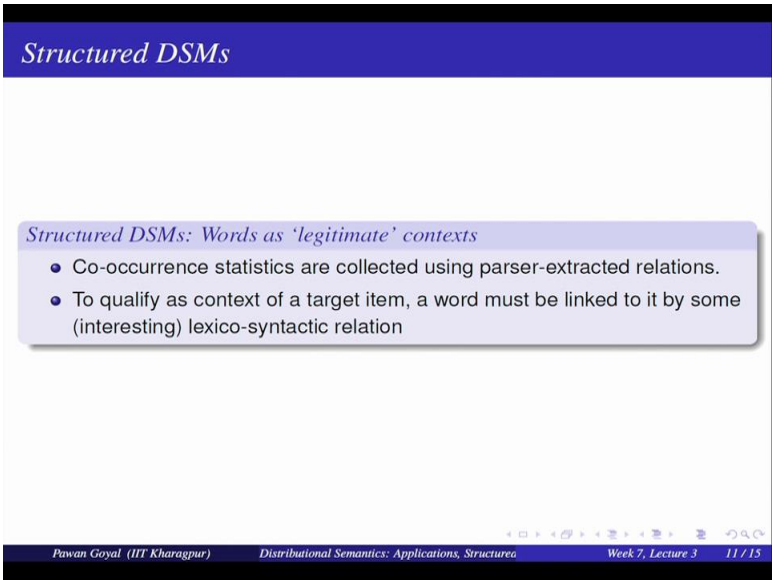
(Refer Slide Time: 25:59)



Let us say I have a sentence like the teacher eats a red apple. So, I can first formulate a dependency graph for this sentence, now I can use this dependency relations to say that only some sort of dependency relations are interesting and others are not interesting. So, for example, I can say that eat is not a legitimate context for red although eats and red co occur with the very small distance context window, I can say that the relation det object and a modifier together do not form a very nice context.

I will not use this co occurrence. So, I can be selective in choosing, what kind of co occurrence information, I will use and what kind of co occurrence information I will not use and I may also give them different sort of weights. So, for example, I can say that the object relation connecting eat an apple will be different than the modified relation connecting red and apple. So, this relation det object and a modifier at can be different relations. So, till now what was happening I was only seeing if this word co occurrence with this word or not.

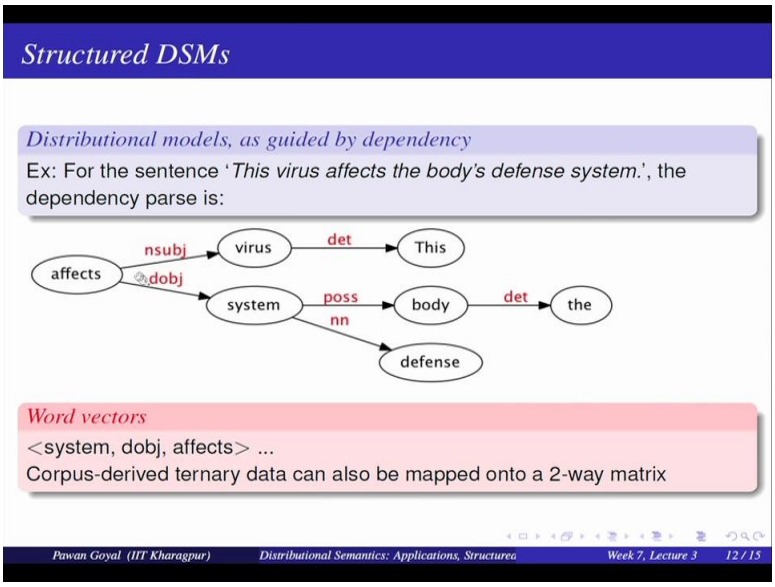
(Refer Slide Time: 27:36)



Now we have started talking about in different terms this word co occurs with another word in this context. So, with using a det object relation with using an adjective modifier relation and so on, now so from the parser I can get all these relations. So, how do I further use those? So we will say that to qualify as a context a word must be linked by some interesting lexicon syntactic relation. So, what do I mean by lexicon syntactic relation a good dependency path should adjust between the 2 words. So, in simple terms

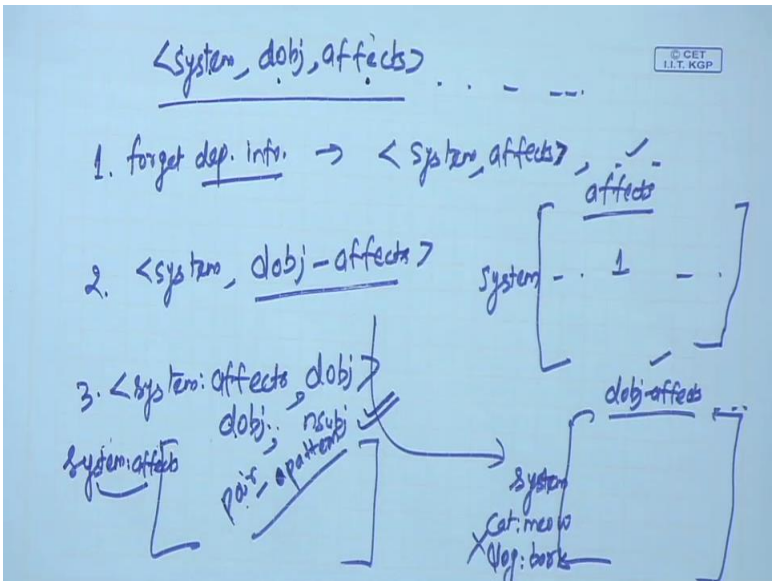
I can only use of use a single edge between 2 words, but in general you can also talk about larger path larger length of edge between the 2 words. So, let us take it simply for the simple paths of length 1 between 2 words.

(Refer Slide Time: 28:11)



What will I do? So, let us see I have a sentence the virus affects the body's defense system and you get the dependency parse, now from the dependency parse I can extract the various triplets like system det object affects body possessive system. So, these are the various topples I can extract from a dependency graph. So, I will have things like this system det object effects and so on, now how do I use those for my structured model? So, let us try to have a look.

(Refer Slide Time: 29:00)



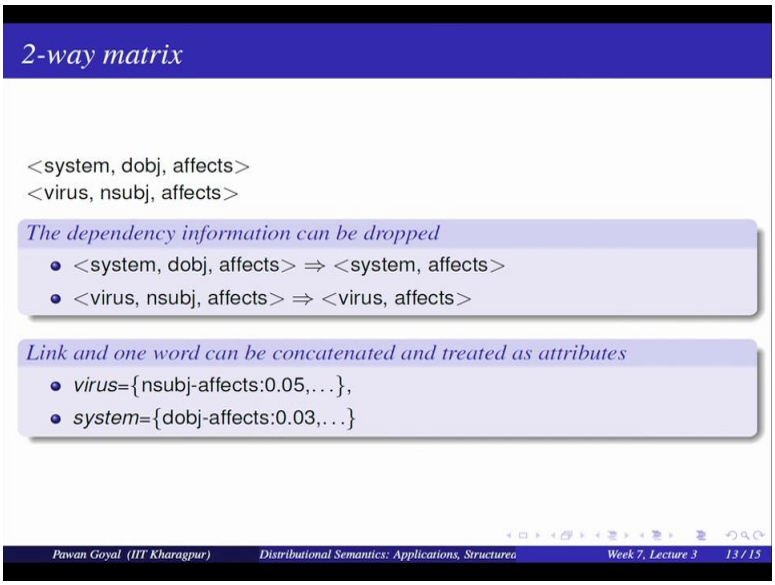
We will have words like system d o b j and affects yes and I have many such pairs many such topples now how do I use those for my distributional semantics. So, there are actually many ways you can do this. So, one simple way would be just forget dependency information so; that means, you will have system affects and so on. So, that is you going back to your earlier relation where you are not using the instruction. So, this you can again convert into that kind of model the word system is here and what is the co occurrence with the word affects and you will have 1 and so on there is one way. So, where you forget the dependency information, but that is not what you wanted right you wanted this information for some reason. So, another option could be I combine dependency information with the context. So, my context is now structured. So, that is I will have system and my context is det object of the verbs affects. So, these are my context now.

Then I can represent system in this context det object of affects det object of other verbs and subject of these verbs and so on. So, now, you see immediately my dimensions are different from here I had only words here, I had verbs and some relation together, but I am going back to the same sort of matrix format and still I cannot use this dog is to bark and cat is to meow, I cannot do it here, now what is some other sort of representation from here that you can gather?

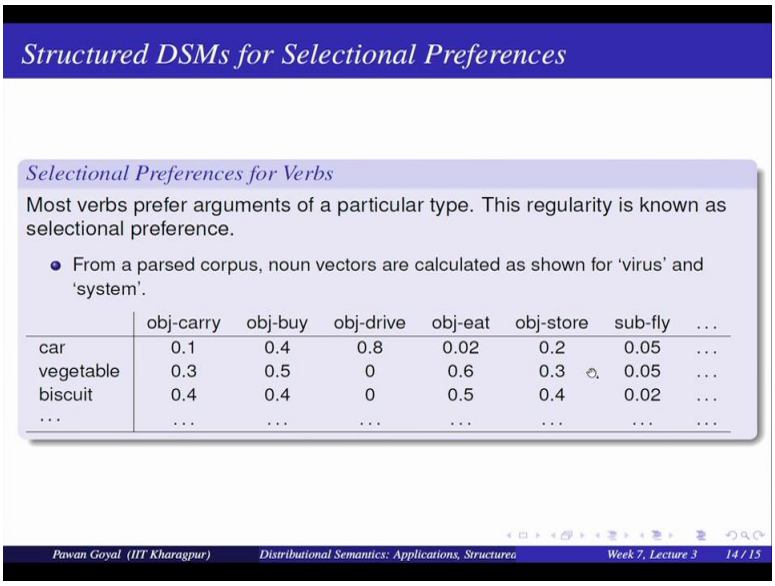
Other could be I combine these 2 in my rows, this become my target and this come become my context. So, that would be third would be. So, like system and affects coming my target and det object is my context. So, now, we can talk about my pair matrix. So, I have systems and affects these are my pairs and pattern right now only dependency relation det dependency relation det object and subject and so on and this is the general framework pair pattern now here we are talking about only very simple paths of length 1 what is the det relation between these 2 words you are free to choose a higher order path it can be there is a path of length det object n subject and so on and use that at your patterns here.

That can help you to capture all these things like X solves Y. So, you can also use what are the different words that occur in between these 2 words, if there are some other words occurring that can be your one of the pattern. So, here you have complete freedom of choosing, what are your patterns? So, these are simple dependency graph for simplicity, but you can choose any other patterns that you would like these patterns come from may be dependency graph come from the word co occurrence how what are the words that are occurring between these words and so on and then you can compute similarity between various pairs.

(Refer Slide Time: 33:02)



(Refer Slide Time: 33:13)



Now, quickly we have this data and I can use that to get this information or this, the other sort of information or the third sort of the information that we saw. Now let us see one simple application like how do we use that to find out selection preferences of the verbs, now what do I mean by this selection preferences? Now different verbs for their different argument like object, subject, they prefer a particular type of noun and this information is useful in many task like dependency parsing.

For eat I want to know that eat will prefer only certain sort of nouns as objects you can eat only very very specific things. So, like, so how do we compute these selection preferences for verbs? So, we have seen that that from a parsed corpus I can compute these vectors like virus and system I can put them in this space how many times car occurs as object of carry. So, you are carrying car, you are buying car, you are driving car, you are eating car and you are storing car and flying car and so on.

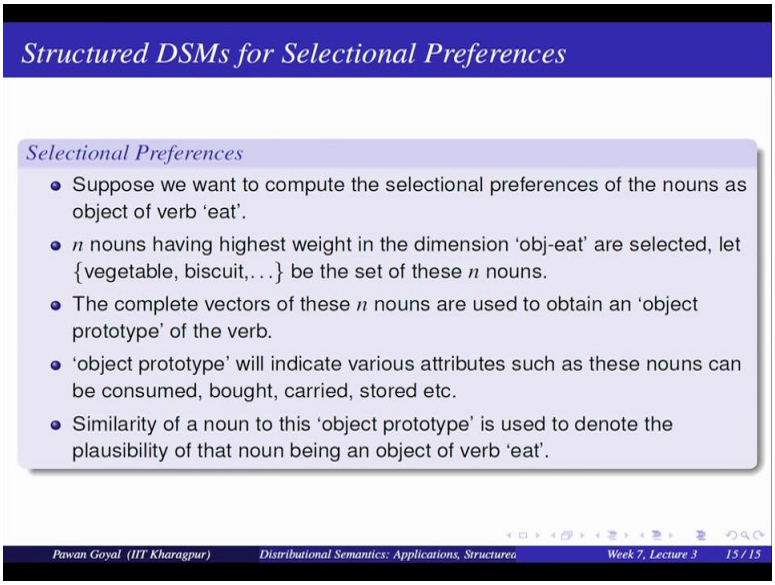
What are the number of times and this is some normalized values vegetables how many times are they carried bought eaten stored and so on. So, this you can compute from the corpus once you have the parse. So, this gives you some sort of representation that what kind of objects come into that can be used, what kind of words that can come as object of the verb buy, what kind of words will come as object of the word verb drive and so on.

But suppose you want to build a prototype that is in general what kind of words can come as object of the verb eat from the corpus here 1 simple thing, I can do, I can find

out which words has have a high value. So, you know vegetables can be eaten biscuits can be eaten. So, they have a high value, but suppose I want 200 prototypes for a new the words that is not occur in the corpus how likely it is to come as object of the verb eat. So, what will I do? So, what is the simple method? So, I would say let me find out what are the words that are having a high weight in this dimension. So, you know vegetables biscuits fruits etcetera will have a high weight in this dimension, now my hypothesis will be words that are similar to these words like vegetable biscuits can also be eat now how do I find out words that are similar to vegetables biscuits and so on.

For that I will build a prototype that is what are these words? So, these are words that can be carried bought stored right so; that means, any other words set can also be carried, bought, stored, might also be eaten. So, then I will find out all the other words that have high weights in these dimensions other than eat because whatever is having high dimension eat I can easily capture here, but what are the other words that are having high dimension in all these high weight in all these other dimensions that also can become my prototype.

(Refer Slide Time: 36:27)



What will we do? Suppose I want to compute the selection preferences of the nouns as object of the verb eat I will take some top n words like vegetables biscuit that have high weight in this dimension of object eat then I take the complete vectors. So, that is of all these top nouns. So, this will words like that can be consumed bought carried stored

etcetera now this becomes my object prototype now given any noun try to match it with this with this object prototype and you will see that how likely it is to come as a object of the verb eat and that is the generic method of finding selection preferences of an word noun for any word noun to come as a object or subject for verb.

So, we talked about, how do you extend your distribution semantic method for using structured models? A lot of work again a lot of research has gone to this domain. You have already touched the basic (Refer Time: 37:26) and I hope if you need this idea you on your own, you can think about how do you use different sort of interesting context, interesting targets and solve various problems.

So, in the next lecture, we will talk about another very import important interesting idea that is what are word vectors? So, and word embedding and how do we obtain them from the corpus and what are different tasks they can be used on? So, I will see you in the next lecture.

Thank you.