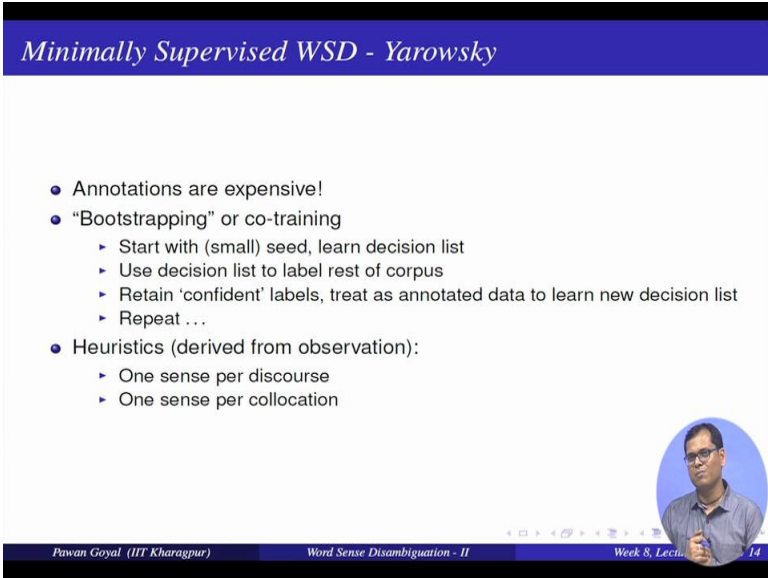


Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 39
Word Sense Disambiguation – II

Welcome back for the fourth lecture of this week. So, we had already started our discussions on word sense disambiguation in the last lecture; and we talked about some approach that we use I that are either knowledge based or use some machine learning algorithm. So in this lecture, we will talk about some approaches that are either semi-supervised or unsupervised.

(Refer Slide Time: 00:41)



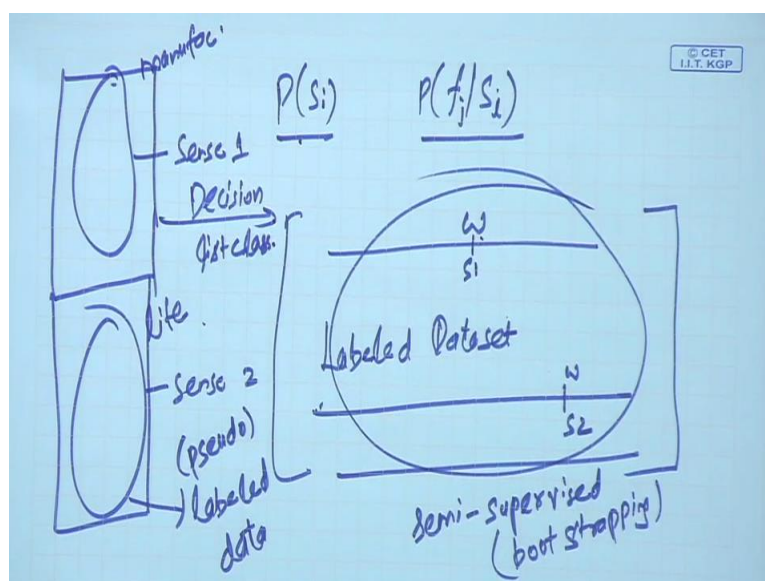
Minimally Supervised WSD - Yarowsky

- Annotations are expensive!
- “Bootstrapping” or co-training
 - ▶ Start with (small) seed, learn decision list
 - ▶ Use decision list to label rest of corpus
 - ▶ Retain ‘confident’ labels, treat as annotated data to learn new decision list
 - ▶ Repeat ...
- Heuristics (derived from observation):
 - ▶ One sense per discourse
 - ▶ One sense per collocation

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

So, why do we need to talk about the semi-supervised, unsupervised approaches? So, in all the machine learning approaches that we can use for this particular problem or any other problem, what is one particular bottleneck let us take the Naive Bayes algorithm. So, for applying Naïve Bayes algorithm, you want to compute the features at that are as such easy some sort of domain knowledge will be required.

(Refer Slide Time: 01:13)



But you have to find out all these numbers like what is the probability of a particular sense, what is the probability of a feature given a sense, yes, now how do you get all these values. For getting all these values, you need to have some sort of label data set. And what do I mean by labeled data set in this context, I will have various sentences where this is my ambiguous word would have occurred and somebody has labeled it that this is sense 1. In this sentence the word w occurs this is sense 2 and so on. So, in some good number of sentences somebody has labeled each word as belonging to one or the other senses; and from there, I can compute all these probabilities. And this has to be done for each individual word, and this can be really, really expensive getting all these annotations and that is true for any generic machine learning task.

So, need to have all these annotations first and that is quite expensive that is why if you can find some sort of semi-supervised approach where starting with very few labels you can generate a lot of different labels, and which may not be 100 percent accurate, but you can assume that they are mostly correct. So, that is also quite well appreciated that is you are trying to use some sort of semi-supervised approach, and that will save a lot of efforts in labeling, and also you use a lot of bootstrapping here.

So, you see one such approach for word sense disambiguation that was proposed by Yarowsky. So, this is also called minimally supervised word sense disambiguation. And what is the motivation that for using a supervised learning algorithm, I have to use

annotations and that are quite expensive. So, I use bootstrapping, so what is the idea I start with some small seed set, and I am learning my decision is classifier. From the small seed set, use my decision list label, the rest of the corpus again from there find out what are the labels that are confident, modify your decision list algorithm, again run it on the corpus, find out more labels and so on. So, you try to go in some sort of a loop of starting from some seed set, labeling some sentences, extending your seed sets of our patterns or rules, again applying it getting more labels done and so on until you have somehow converging.

So, and you can use some sort of heuristics for doing this semi-automatic labeling. And two heuristics are very common in the case of words disambiguation, one is one sense per discourse and another is one sense for collocation. One sense for collocation we have already seen in the previous example of decision list classifier. What is one sense for discourse? So, idea is that suppose you are taking one particular document or one particular long context; if the same word has been repeated multiple times most probably it is use in the same sense. So, for example, you are seeing a cricket commentary or some news article about sports and you find the word bat has been used at one point and you know this category is sense one.

Now, if the same word bat occurs in some nearby places, you can probably assume that this is also being used in the same sense, and this is the idea. So, in the same discourse, if your word appears multiple times, I tag it with the same sense.

(Refer Slide Time: 05:12)

More about heuristics

One Sense per Discourse

- A word tends to preserve its meaning across all its occurrences in a given discourse

One Sense per Collocation

- A word tends to preserve its meaning when used in the same collocation
 - Strong for adjacent collocations
 - Weaker as the distance between the words increases

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 3 / 14

So, these are my two different heuristics one sense for discourse that is a word tends to preserve its meaning across all its occurrences in a given discourse. And one sense for collocation we have already seen that. If it is used with the same collocation, it will have one meaning, and generally it is strong for nearby collocations, and if the distance becomes higher it becomes weaker and weaker. So now, how do we use these ideas to construct a semi-supervised approach for words disambiguation?

(Refer Slide Time: 05:46)

Yarowsky's Method

Example

- Disambiguating plant (industrial sense) vs. plant (living thing sense)
- Think of seed features for each sense
 - Industrial sense: co-occurring with 'manufacturing'
 - Living thing sense: co-occurring with 'life'
- Use 'one sense per collocation' to build initial decision list classifier
- Treat results (having high probability) as annotated data, train new decision list classifier, iterate

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 14

So, idea would be let us take the same example that is I am having the word – plant, it has two senses, one is industrial sense, another is living thing sense. So, I will start with some sort of seed labels or some seed collocation. So, suppose I say that with the first sense that is industrial. If the word plant occurs with manufacturing, it will have only the first sense. Similarly, if the word plant occurs with life, it will have the living thing sense. So, this becomes my decision list classifier I check simply if the word plant is occurring with manufacturing or life; manufacturing, I say it is sense 1; if life, sense 2.

So, what will I do, I will take my whole corpus see wherever with the word plant one of these words either manufacturing or life occurs; wherever the word manufacture occurs I say this is labeled as this is my pseudo label sentence. So, now I have some set of pseudo label sentences both with manufacturing and using manufacturing and life. So, what I would have now, from my corpus, I would have labeled some sentences with sense 1, with sense 2. So, these are all the sentences where the word plant occurs with manufacturing and these are all the sentences where the plant occurs with life.

So, now the idea would be. So, now, I am doing going iterations. So, now I will treat them as my labeled data. Although to be very clear, we can also call it my pseudo labeled data. It is like I am only making an assumption that they are labeled with sense 1 and sense 2. Now, treat it as your label data and then apply your decision list classical algorithm to find out what are the good collocations.

Suppose, you find some three, four collocations here, so it can be different things about growth and whatever or the word car like what we saw in the previous case. So, we find some collocations here. Now, use this collocations and your algorithm to build a new decision list classifier - new classifier. Again you will use that to label new sentences. And you will have new data, you will get new collocations from there, and you keep on repeating the system.

(Refer Slide Time: 08:32)

Yarowsky's Method: Example

used to strain microscopic **plant life** from the zonal distribution of **plant life** . close-up studies of **plant life** and natural too rapid growth of aquatic **plant life** in water the proliferation of **plant** and **animal life** establishment phase of the **plant virus life cycle** that divide **life** into **plant** and **animal kingdom** many dangers to **plant** and **animal life** mammals . Animal and **plant life** are delicately automated **manufacturing plant** in Fremont vast **manufacturing plant** and distribution chemical **manufacturing plant** , producing viscose keep a **manufacturing plant** profitable without computer **manufacturing plant** and adjacent discovered at a St. Louis **plant manufacturing** copper **manufacturing plant** found that they copper wire **manufacturing plant** , for example s cement **manufacturing plant** in Alpena

vinyl chloride monomer **plant** , which is molecules found in **plant** and **animal tissue** Nissan car and truck **plant** in Japan is and Golgi apparatus of **plant** and **animal cells** union responses to **plant** closures . cell types found in the **plant kingdom** are company said the **plant** is still operating Although thousands of **plant** and **animal species** **Animal** rather than **plant** tissues can be

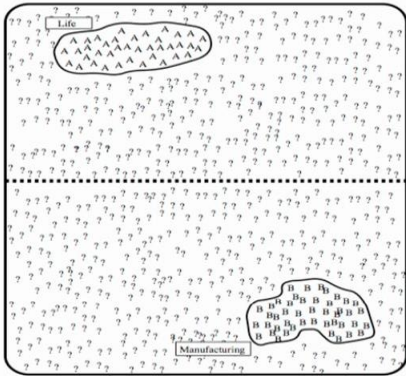
Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 5 / 14

So, let us see for this particular example. So, suppose I use that and I label some sentences with sense of living thing, some with sense of industrial plant. So, you see here everywhere life is occurring, here everywhere the word manufacturing is occurring. Now, from these labeled or by pseudo labeled data, I will try to extract some more collocations like here suppose I see that the word like animal and kingdom occur a lot with this sense, but they do not occur with this sense.

So, this becomes my collocation now. Now, use your decision list classifier algorithms previously that we have talked about to make my decision list. Now, again run it over the corpus, and then you can find suppose the word where plant and animal occurs. So, here you can again label it with sense of sense of living thing and from once you have done that you can extract some more collocations from here and you keep on doing that.

(Refer Slide Time: 09:35)

Yarowsky's Method: Example



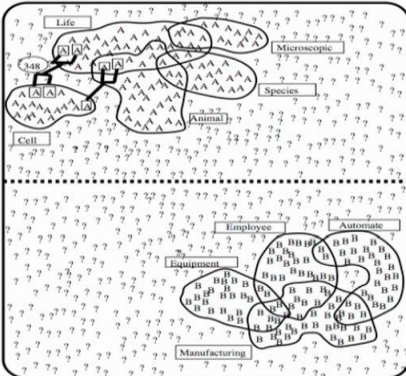
Initial state after use of seed rules

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

So, idea would be something like that. So, you have this whole corpus where the word plant occurs somewhere by using this initial seed set of life; and manufacturing you have label sense a and sense b. Now, what you will do, you will treat them as label data from their capture some more collocations.

(Refer Slide Time: 09:58)

Yarowsky's Method: Example



Intermediate state

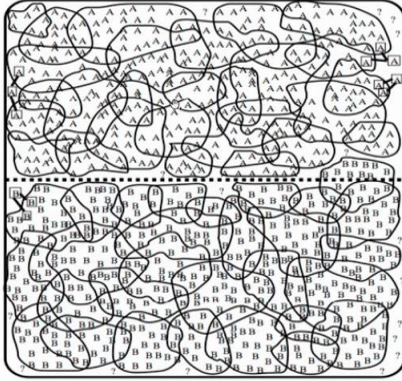
Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

So, something like this. So, you capture collocation like cell, animal, species, microscopic for the sense of life; and equipment, employee, automated, manufacturing, for the sense of industrial plant. Now, use that to label more sentences and it is a more

sentences that you have labeled. So, all these content this microscopic species, animal and so on. Now, you have a larger data, again extract collocations from where build your entries and try to label the rest of the corpus, so you keep on iterating this. And ideally at some point you will have you whole corpus covered

(Refer Slide Time: 10:32)

Yarowsky's Method: Example



Final state

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

(Refer Slide Time: 10:40)

Yarowsky's Method

Termination

- Stop when
 - Error on training data is less than a threshold
 - No more training data is covered
- Use final decision list for WSD

Advantages

- Accuracy is about as good as a supervised algorithm
- Bootstrapping: far less manual effort

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

So, when do you stop, when the error on the training data is less than a threshold? So, we can have some small training data to check; how could this bootstrapping approach is performing or when no more training data is covered by the algorithm. When you see

that all the examples have been covered, and whatever final decision list you get by this algorithm use that for word sense disambiguation. Now, what is the advantage of this approach? So, you can see that the advantage here are that it would have to put the efforts into manual labeling of the corpus. So, you can start with some very small seed set keep on applying this algorithm retroactively and obtain the final classifier.

Now so accuracy also turns out to be quite good, but what are the different what maybe one problem with this approach. So, one problem with this approach might be that the whole accuracy that you get by this approach depends on how good are your initial seed set. If your initial seed set is not good, then you may not go very far in this approach. So, the good thing is if you choose very good initial seeds that have lot of represent the corpus and are also very, very distinctive. You see this gives starts giving you very good precision and good recall from the very beginning. So, now this is a semi-supervised approach, and I hope the idea is clear.

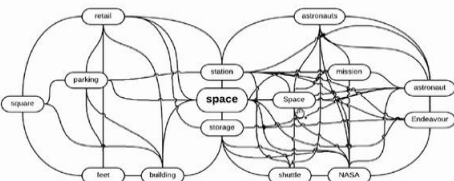
Now, we will see one unsupervised approach for word sense disambiguation. By unsupervised I would mean that the senses for the words are not defined a priori, but they are sort of learned from the data. So, from the data, you try to find out what are the different senses a word is used in, and this becomes your sense definition for each word. And this field is also called word sense induction as such. So, you are trying to induce the sense of the word by its usage in the corpus.

(Refer Slide Time: 12:52)

HyperLex

Key Idea: Word Sense Induction

- Instead of using "dictionary defined senses", extract the "senses from the corpus" itself
- These "corpus senses" or "uses" correspond to clusters of similar contexts for a word.



Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 14

Now what is the basic idea? So, we will talk about the algorithm the HyperLex. So, key I idea here is word sense induction that is instead of taking some sense that are defined by a dictionary, try to extract the senses from a corpus itself. So, the way the word is used in a corpus, use that for extracting senses of the word. And these corpus senses or uses will correspond to clusters of similar context for a word.

Now, let us take one example to get the intuition. So, here you are seeing the word space and there are many other words that are coming along with this space. So, the connections show here that two different words occur together above some threshold number of times in a corpus. So, as denotes that they are probably similar in the sense that they are co occurring a lot. So, now, try to look at this simple picture. So, what do you seeing the words like retail-parking are connected, retail-square are connected, feet and parking connected here in the left hand side; right hand side you are seeing words like astronauts, mission and NASA, shuttle all these are connected.

Now, by looking at these connections, so what is one thing that you are seeing is that you are finding out two different clusters here; one in the left hand side one in the right hand side. One is denoting one sense of space in the sense of parking space, retail space. In secondary space as such space where astronauts will be going in and that involves NASA and all. Now, from the corpus I can find out that astronauts - NASA, and shuttle-NASA connected, retail-parking, retail-square all these are connected. But how will I find out that the word space has two senses, and these two senses correspond to some particular words.

So, if you look at the figure again, you can see that for a particular sense of the word space, the words will be highly connected to each other; for another sense again these words are highly connected to each other, but there will not be much connections between the words in this sense and words in that sense. So, that is if I construct this as a graph and use some sort of clustering algorithm to find out what are the different portions of partitions of this graph.

So, one partition will belong to one sense another partition will correspond to another sense. And this is the idea try to exploit how much that the difference for the same sense the words will co-occur a lot together, but it will not happen for the words across two

senses. And there are many, many different algorithms that are developed based on this idea for word sense induction.

(Refer Slide Time: 16:07)

HyperLex

Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G .
- **Step 2:** Arrange nodes in G in decreasing order of degree.
- **Step 3:** Select the node from G which has the highest degree. This node will be the hub of the first high density component.
- **Step 4:** Delete this hub and all its neighbors from G .
- **Step 5:** Repeat Step 3 and 4 to detect the hubs of other high density components

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 11 / 14

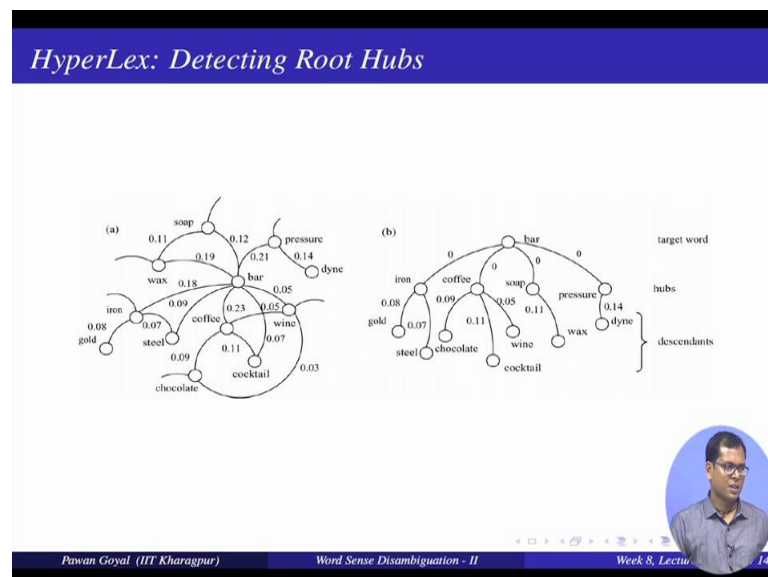
So, now we will talk about a very simple approach HyperLex that uses this idea. Now, what is this algorithm HyperLex. So, what it does is that for a given word that is ambiguous by using the data in my corpus, it tries to identify what are the main hubs. So, each hub will correspond to one sense of this word. So, it tries to identify, what are the main hubs and every other word in my co occurring graph will be connected to one or the other hubs. So, you know everything is divided into these hubs. So, how do you detect these hubs and how do you connect the words to one of these hubs.

So, in HyperLex algorithm, the idea is that the different usage of the target words form highly interconnected bundles or high density components; and each component will have one of the nodes this is hub that is having a high degree than the other nodes. So, how do you start applying this algorithm? So, firstly you have to construct your co occurrence graph. Co-occurrence graph we have seen already in the previous week that I find out how many times these two words occur in the corpus and I use some function of that to find out how much their association is strong how strong is their association. And I will probably retain only those words that have a very high association. So, I start by connect correct building this graph from the corpus co occurrences.

Now, so what do you do this graph is connected around this word this ambiguous word like a space in the previous example. So, first thing I will do I arrange all the nodes in this graph in the decreasing order of their degree. So, I will forget about the word space, but every other word will be connecting the decreasing order of their degree. So, what I will assume that whatever the hubs are will have high degree shape. So, take the node that is having the highest degree among all the connections and this will be the first hub. Try to find out what is the neighborhood of this hub take them to this sense cluster remove this from the graph altogether from the remaining graph find out what is the node with the highest degree make it the second hub. Find out its neighborhood make it the second sense cluster remove it, keep on doing that.

So, once I have arrange the nodes in the graph G in decreasing order of degree, now select the node from g that has the highest degree and this will be the hub of the first high density component. Now, delete this hub and all its neighbors from the graph. From the remaining graph, again repeat the step 3 and 4 to find out what are the hubs and what are their high density components.

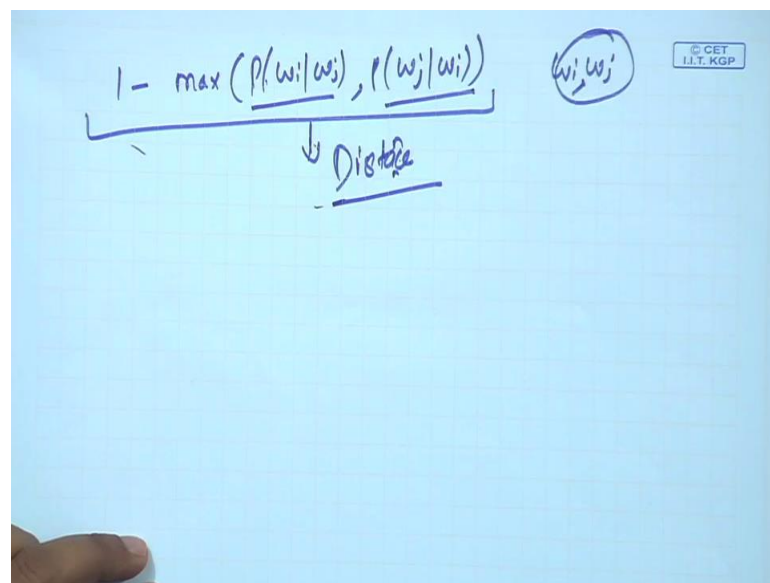
(Refer Slide Time: 19:00)



So, let us try to see this algorithm on a simple example. So, here what do you have, you have the word bar that is the central word; and I want to detect what are its different senses. So, what we have done we have first try to construct the co occurrence graph. So, this is what you are seeing is a co occurrence graph that tells what is the strength of

association between two words. One thing here, I will talk about what is association measure. So, according to this association measure, you will take a threshold and choose only those edges that are above or below the threshold depending on how you are defining your threshold. In this particular case, so the number denotes what is the distance between two words. So, you are only retaining those words whose distance is below a threshold. So, say your threshold is 0.25, so we written only those connections that are below 0.25. And distance can be captured by something that is inversely proportional to the association.

(Refer Slide Time: 20:18)



The image shows a handwritten formula on a blue grid background. The formula is
$$1 - \max\left(\frac{p(w_i|w_j)}{p(w_i)}, \frac{p(w_j|w_i)}{p(w_j)}\right)$$
 with a bracket underneath. An arrow points from the bracket to the word "Distance" which is underlined. To the right, the words w_i, w_j are circled. In the top right corner, there is a small logo that says "CET I.I.T. KGP".

So, in this case it is 1 minus probability of max of probability. So, it is 1 minus max of probability w_i given w_j , and probability w_j given w_i . So, we take the max of that to find out what is the association would be w_i, w_j . And this will now capture the distance, because this corresponds to the similarity how similar they are, their condition probability. If I take 1 minus max of that that is how much they are different what is the distance. So, once we have done that for all the words, all the pair of words, I will take only those that are having distance below a threshold that means, they are quite clear and that is captured in the left hand side of the figure.

Now, once you have done that now you apply your algorithm that is finding out what are different hubs and taking their different neighborhoods. So, first I will arrange all the words in decreasing order of the degree. Now, what are the nodes here that are having

the highest degree? So, you find the words like iron, iron has a degree of 1, 2, 3, and 4; coffee has degree of 1, 2, 3 and 4; wine also has a degree of 1, 2, 3 and 4. So, now there are multiple words that are having the highest degree. So, we will choose some preference may be either the lexical graphical order on some or some ordering on the, or some travels or ordering on the graph. And suppose you say from I will choose the one from the left, and you choose the word like iron here as your first hub.

So, what is the next step make iron as your first hub and then take all the neighbors of iron and put them with this hub. So, we are taking the two neighbors gold and steel and putting them with this hub here. And that becomes your first HyperLex components. Now how do you find the next hub, you remove this hub and all its neighbors. So, I remove iron, gold and steel from the graph. Again find out what is the node with the highest degree? You will find the word like coffee here. So, take coffee as the second hub take its neighbors, so it will be chocolate, cocktail and wine. So, these become its neighbors, this becomes my second hub and all the component with that.

Remove that from the graph now find out the next hub. So, this can be the word soap that is having a degree 3, you take the words soap and wax as your third hub and its component. And then finally, you will have the word like pressure and dyne that is your fourth hub. So, this is my hub and these are different descendants of the hub. So, from your corpus, you are focusing one word k bar, and you are trying to construct the co occurrence matrix finding out the association between various words, building out this graph, from there you are detecting what are your various hubs and the descendants. And this becomes your instruction; this is your target word different hubs and their descendants. And this is what you have obtained in a completely unsupervised manner, because nobody told you how many senses the word bar would have or what are the different senses of bar you found it automatically by using the corpus data.

(Refer Slide Time: 23:54)

Delineating Components

- Attach each node to the root hub closest to it.
- The distance between two nodes is measured as the smallest sum of weights of the edges on the paths linking them.

Computing distance between two nodes w_i and w_j

$$w_{ij} = 1 - \max\{P(w_i|w_j), P(w_j|w_i)\}$$

where $P(w_i|w_j) = \frac{freq_{ij}}{freq_j}$

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 13 / 14

So, for all other words other than hub, you attach them to the root hub that is closest to them. And how do you find what is the closest hub you can take the distance between that node and the different root hubs. And this distance is simply the summation over the path length. What is the path length you just keep on adding the path length that is the distance between any node and the root hubs, attach each node to one of the root hubs only. And this is something that I talked about that how do we compute the distance between two nodes in my original co occurrence graph I take this 1 minus max probability of w_i given w_j and probability of w_j given w_i .

So, we can see that how do we start from my corpus and construct different senses of a word, so y hub and descendants. Now at run time, so we are talking about this problem what is in the disambiguation. So, at run time, I am given a sentence where this word is provided; and I want to find out what is it is appropriate sense that is used. Among all the possible senses I have fine formed by this algorithm. So, what is the approach?

(Refer Slide Time: 25:14)

Disambiguation

- Let $W = (w_1, w_2, \dots, w_i, \dots, w_n)$ be a context in which w_i is an instance of our target word.
- Let w_i has k hubs in its minimum spanning tree
- A score vector s is associated with each $w_j \in W (j \neq i)$, such that s_k represents the contribution of the k th hub as:

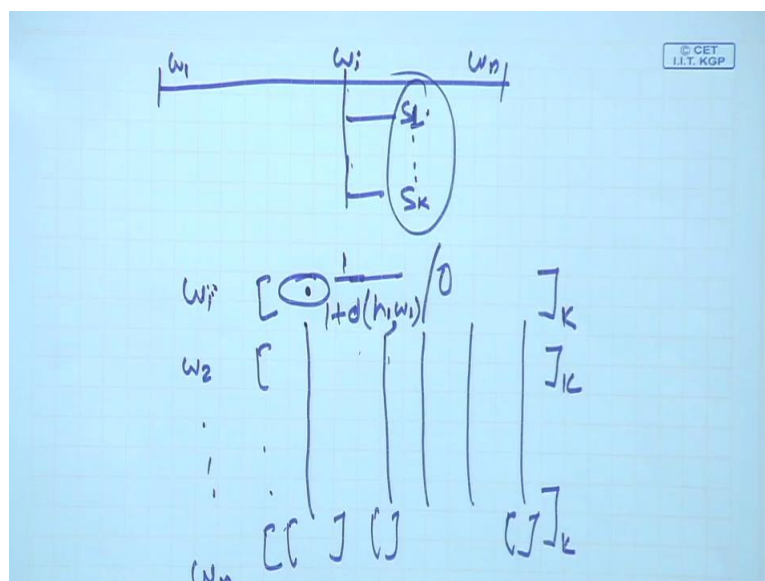
$$s_k = \frac{1}{1 + d(h_k, w_j)} \text{ if } h_k \text{ is an ancestor of } w_j$$
$$s_i = 0 \text{ otherwise.}$$

- All score vectors associated with all $w_j \in W (j \neq i)$ are summed up
- The hub which receives the maximum score is chosen as the most appropriate sense

Pawan Goyal (IIT Kharagpur) Word Sense Disambiguation - II Week 8, Lecture 4 14 / 14

So, let us say I have this context or the sentence where this word w_i occurs and w_i is my target word that has multiple senses. So, suppose they are it has k senses, k hubs that are k senses. So, what do we do, we associate a score vector s with each word in the context such that s_k denotes what is the contribution of the, what is the contribution it will have to the k th hub. And this is simply taken by 1 divide by 1 plus distance between the hub and the word. If the hub is an ancestor of the word w otherwise it is 0. And you do that for all the words in my context and sum those over. So, find out a simple vector that tells me which sense has how much score whichever have how much score and whichever has hub has the highest score I will choose that.

(Refer Slide Time: 26:28)



So, to tell that again, so what we will do, so we will have a sentence w_1 to w_n and I have this word w_i that has s_1 to s_k , k hubs. Now, I want to find out which of these k senses is used in this particular example, particular sentence. So, what will I do for each word w_1, w_2 up to w_n , I will construct this score vector. So, score vector has that many dimensions as the number of hubs, so it has k dimension. So, I will construct this vector for each of the words.

So, now what are the entries in this vector? This entry tells me how much contribution this word will make to the first hub. And this contribution would be if the particular hub h of the sense one is an ancestor of this word w_1 , it will be 1 divided by 1 plus distance of h_1 and w_1 . So, that is if the distance is high this score will be low if the distance is smaller this will give a high score.

On the other hand, if this word, if the hub is not ancestor, I will put a score of 0 ; like that I will put all the different values here. And finally, once I have all the values, I will add all these. So, for each hub, I will add all the possible scores; and I will take a final vector that is how much contributions all the words together are making for hub 1 , hub 2 and hub k . And I take the one that is having the maximum score among all these and that becomes my winner sense. So, this is the overall idea of this approach. So, what we have done here we did not start with any distance defined sense, we used a corpus and defined our own senses by seeing what are the different components that occurred together. So,

idea is that for a given sense of a word different words would have a high co occurrence they will make some sort of cluster. So, identify different cluster size different senses.

Now, once you have these senses if you want to use them for word sense disambiguation at run time whenever the word is used, find out from the context words which of thus different hubs they are closer to. So, whichever hub gets the high score becomes your disambiguate sense, so these are some different ideas for approaching this problem words and disambiguation. Although I said earlier, there are many, many different approaches that have been proposed.

So, in the next lecture, we will briefly talk about an idea of word sense discovery that is how do you find out if the word has got a new sense in the corpus. This is a relatively new field, and we will see how the ideas that we have developed in these lectures to how do we use them to find out if a word has got new sense in the corpus.

Thank you.