NaturalLanguageProcessing-Lecture17

Instructor (Christopher Manning): – in some sense, that's not a very deep topic. There's some kind of ideas as to how people organize word meaning, but in some sense the main resolve is that languages have a lot of words and you need to know their meanings to do anything useful in natural language processing so it's sort of more important rather than having these analytical techniques. On the other hand, there are some quite interesting algorithms that they've developed to learn word meanings. So here's my warm up question for my very studio audience. So the word "pike" what meanings does the word pike have?

Student: Fish.

Instructor (Christopher Manning):Fish. It is a kind of fish. Yep.

Student:[Inaudible]

Instructor (Christopher Manning):Of?

Student:A weapon.

Instructor (Christopher Manning):A weapon. Yeah. Yes, so it's a kind of fish and a kind of weapon –

Student: Short for turnpike.

Instructor (Christopher Manning):Short for turnpike, yes. So you can have the – what's that pike that goes across New Jersey? Yeah, so it's a road. Any other meanings for the word pike? Okay. I did my homework before class. I bet there's at least one more meaning that you would recognize. Part of this shows how senses of words are very domain specific. I'll give a hint. It's coming up later this year in Beijing. In sport – at the Olympics with the pike. Anyone watch the Olympics ever? [Inaudible]. So in diving and in gymnastics, you have a pike as a kind of dive. Did you know that? Yeah, yeah. Some people [inaudible] that they do that meaning. As an Australian, it turns out that there's also an additional meaning of pike, which is used as a verb, which means to kind of withdraw and not follow through in doing things. People say something like, "He was gonna come for beer, but he piked," meaning that he decide not to go. But I don't expect you to know that one, but that again shows that often there are lots of dialect and lots of uses of different words. Something that's also kind of vaguely interesting is just well, how do all of these meanings come to be? It turns out that most of those meanings do actually have something to do with each other historically. Supposedly, the Oxford dictionary tells me that the fish, the pike, was named after the weapon because it has a kind of a pointy head like the weapon the pike and, well, it turns out that the turnpike is kind of named after that sort of star shape as well with roads when they have those kind of turning things. So they're kind of historical reasons often why words are related. Not

always, sometimes they just come together by chance, but that doesn't mean that the people who use these words know all the historical stuff.

So I'll go on to talk about different word senses more. But let me before I do that just say quickly a couple of announcements, which I'll also send email out about. So the final projects were officially due Wednesday at midnight, now, some people have already asked about whether you can have more time and things like that. I guess the basic answer is no, and the reason for that is it turns out that the spring quarter, especially this one, is really tight grading deadline to get things ready and graded before commencement and there's just no chance that we could possibly do that when there are people that already have lots of late days left unless we stick to that deadline. So I'm prepared to make one small concession for people who are out of late days, which is I will say it is okay if you hand it in by Thursday by 10:00 A.M., but I think really that is the limit of what we can do unless we have some to grade before the weekend starts there's no way that we'll be able to get through reading them all. So then as well as handing in the final project; during the exam slot we're gonna have final project presentations so they're gonna be in this room and they're scheduled for Monday morning so in our kindness and own desire to get some extra sleep, we're not actually gonna start at 8:30, we're gonna start at 9:30 and the plan is, essentially, that there will five minute for each group to give their presentation. In general, this has been quite a fun thing to actually see what different people have been working on and have been able to achieve, but it's a mandatory thing that we expect that at least one person, and preferably all from each group, to turn up for the final presentations. And so for the nature of what to do what we want is something that's very short, like, five plus or minus one, PowerPoint slides, or if you have a moral objection to using PowerPoint you can make them in open office and providing there's something we can use as slides, but what we would like to do is gather them all in advance because the only way that we can make things kind of run on time for short presentations is to actually have them all running on one computer.

So, essentially, what you should be aiming to do is have an elevated pitch style presentation where effectively there's a slide saying what's the problem that you are working on, there's a slide that says something about the methods that you used, there's a slide that shows some of the results that you have and there should be a slide that has really concrete example stuff. I mean, it's very hard for people to get much of a sense of what you're doing if it's all completely abstract whereas if you actually show us some examples of what the input and output looks like, then that's sort of much more concrete and visible. Finally, on my third reminder, the gates are now open for you do official evaluations of the course, and we do very much appreciate getting any feedback on what you thought of the course and how it could be better and things like that. So there's the official [inaudible] where they essentially bribe you to take part by only giving you access to your course results on an early date providing you fill in the evaluation. As well as that, I also will encourage you that there are at least now two sites that kind of do an official public course commentary as well, so there's the dot dot dot standard courses.com and then course [inaudible] at standard.edu commentary on those is perhaps kind of easier and more pleasant and more public as well.

Okay. Those are all my announcements and so I will go on. Okay. So lexical semantics – it's completely obvious that we spend a fair part of doing compositional semantics and you can have all the clever compositional semantics as you want, but you can't actually do anything unless you actually know what words mean. And in fact, going from the other direction, many people would argue that the sort of natural language applications, the lexical semantics is largely where it's at, that most of what you need is knowing means of words and some of these subtle issues of how meanings combine sort of really either not so many commonly needed or rather be on the state of the art about natural language applications anyway. But that's tricky because word meaning is all this messy stuff that there are all of these words with all of these interesting meanings that we need to deal with. One kind of ambiguity with words is their part of speech. That's normally handled separately by doing part of speech disambiguation and we didn't spend a lot of time, specifically, on that, but implicitly, the parses that you guys build also did part of speech disambiguation. Some of the senses that I had a moment ago for pike you could distinguish verb and noun senses, so my Australian, "he piked," is a verb. You could also use the infantry weapon sense as a verb. You could say, "I piked him through the heart," or something like that. Here are some of the well-known examples of words with different senses, bank, you can think of multiple senses of that, score, games, music etcetera, right, direction, legal rights, set stock, notice that those are all kind of short common words. I'm sure there are exceptions, but essentially, all short common words have lots of words senses. Look them up in the dictionary and you'll find them.

So we have this problem where words have lots of senses and it seems that effects and is part of a lot of the applications that we have so everything through information retrieval to word to machine translations and natural language understanding where we kind of need to know the senses of words. And finally, you then also have these words that are spelled the same, but pronounced differently. So for a word like bass, it can be pronounced either bass or bass with different meanings, fish is a bass. And if you're gonna do applications like speech synthesis, you then need to know which word sense is required to pronounce it correctly. An example of a lexical entry for the word stalk which comes from Elders, so Elders was a pioneering dictionary that was done in the late 1980s in the UK. So Elders was, essentially, the first dictionary that was created by people making electronic corpora and actually looking at what was found in the large corpus and arranging the dictionary based on appearance and frequency in a corpus. Elders was also the first dictionary where the publishers were willing to make it available to researcher for less than a truly extortionate amount of money and so if you go back in the history of computational linguistic all of the early work [inaudible] with arrangeable dictionaries was done with Elders.

So here we go. Stalk; a supply of something for use, a good stock of food, goods for sale, some of the stock is being taken with being paid for, the thick part of a tree trunk, a piece of wood used as a support or handle as for a gun or tool, the piece which goes right across the top of an anchor from side-to-side, a plant from which cuttings are grown, stem onto which another plant is grafted, a group of animals used for breeding, farm animals, usually cattle, a family line, money lent to the government at a fixed rate of interest, the money owned by a company divided into shares, a type of garden flower with a sweet

smell, a liquid made from the juices of meat bones. You should actually take a moment to sort of look at these and just realize what a [inaudible] enterprise this is. So people who produce dictionaries take words, if it's a modern corpus based sense, they look at a bunch of examples of it in a corpus using the concordance tool and even in the old days when Samuel Johnson was doing it, examples of usages of a word were collected on index cards and people would look through them and they, effectively, do a clustering path. So example, if you take these first two senses, should they really be divided off as two senses? It sort of seems like there's at least half an argument that they're the same sense. You have a supply of something and sometimes you have it sitting at home and sometimes that you have it sitting in a store. Okay. So as well as having word senses and synonyms, you can also think of words as being organized in a hierarchy or a taxonomy. And the standard representation which you know old fashion computer scientists might think of is a hierarchy gets called in lexical semantics and hyponyms and hypernyms which are going in an opposite directions in a lexical hierarchy. So cars is a kind of vehicle, dog is a kind of animal. Traditionally, this wasn't information that was represented in dictionaries, that dictionaries have commonly listed senses of words and synonyms of words, but traditionally they haven't listed this as a kind of information. That's something that's been addressed for recently. Okay. So if we think of lexicon and draw some pictures, something I haven't mentioned so far is people normally make the distinction between word forms and lexemes. So word forms are particular and [inaudible] forms of a word, runs, running, eat, eats, ate, eaten, and then that's contrast of the lexeme, which is kind of a baseball of a word that you put in the dictionary. Normally, you don't put word forms in the dictionary. And then a lexeme can have various senses as we've discussed.

Sometimes you want to say a word has multiple lexemes. The clearest cases of words that have nothing to do with each other, but can come together in the same lexeme, so something like the bass tone versus the bass fish, that would be two lexemes although they're the same word string. And then over here we have the senses for words. So normally, one lexeme will have multiple senses. So when people think of synonym, really synonymy is best represented as two words sharing one sense. Okay. If you're a computational linguist and you want to do lexical semantics and you're not working for a rich company, by and large, what everyone uses is WordNet, and even if you are working for a rich company, by and large everybody uses word net because it's freely available, no licensing restrictions, no hassles. So it was built at the University of Princeton. WordNet was originally sponsored by George Miller, who's an extremely famous psycho linguist. George Miller is now a very old guy, but he's worked in psycho linguistics since back into the 50s, so he's essential a contemporary of a Chauncey's. He wanted to come up with a new lexical representation which was more in accord with how words are organized and stored in the brain. I think, in practice, as time has gone by, that motivation has been largely lost except in the very, very loose sense that WordNet does contain a larger network of words and you might think that that somehow feels a little bit how your brain organizes information.

And it follows the organization that I just mentioned, so it keeps part of speech separate, which was claimed to be supported by psycho linguistic research and then inside each

part of speech you have this organization where you have lexemes that belong to various synsets where the synsets are the senses that I showed on the previous page. A little quirk of WordNet is it has no coverage whatsoever of closed class parts of speech. It only does nouns, verbs, adjectives and adverbs, which is occasionally annoying because sometimes you'd like to know about things like prepositions that have similar meaning. So very quickly, I'll just show you a few stats on WordNet. So the noun database has about 90,000 lexemes, a few more than that senses, and it has a kind of a rich set of links, so it has hypernyms, hyponyms, has-member, has-part, antonyms, and actually some additional stuff. When you have these synsets, the synsets are effectively groupings of words which have claimed have one sense in common. So, in general, the organization of nouns is very elaborate, and in areas like natural kinds like this, it's especially elaborate and stores tons of stuff that actually regular human beings don't actually know.

Okay. When you go to verbs and adjectives and adverbs, the structure isn't as rich. So the number of verbs is just much smaller, that's just a fact of English. There's only about 10,000 verbs in English. There's not a huge number like nouns. So then for the rest of the time, I then want to talk about some of the things that people do in the main computational lexical semantics. So the thing I'll spend the most time on is word sense disambiguation of trying to find out the senses of words. But then I'll spend a bit of time at the time talking about words senses and working out word senses similarity and other fun things you can do in lexical semantics. Okay. The task here is to find out what sense of a word is intended by the context in which it's used. So if we take these examples here, "The seed companies cut off the tassels of each plant making it male sterile," there a plant is a living green thing. "Nissans Tennessee manufacturing plant beat back a United Auto Workers organizing effort with aggressive tactics. There, "plant," is in the sense of industrial factory. You can kind of already see what needs to be done here. We need to do this categorization task to work out which sense is intended in context to help with various applications. And what kind of information can we use? Well, once source of information that we had hoped to use is just prior information, and if you're kind of naive and think plant, that means a green thing most of the time. That's a form of prior information. It turns out that if you're getting your sentences from newswire, as these sentences are, that's actually the wrong prior. This has normally this has been done as a categorization task, which is a supervised learning task. It's also sometimes done as a unsupervised clustering task. Normally, if you're doing that you're settling with just doing a kind of a word sense clumping. Okay. I'll take a quick detour into the history of the early days of word sense disambiguation, which is a little bit interesting.

So the same Warren Weaver, who I quoted before, as initiating all the work in word sense disambiguation, he immediately noted that word senses are going to be a problem with machine translation, so he noted, "A word can often only be translated if you know the specific sense intended. A bill in English could be a pico or a cuenta in Spanish." And so then in the early days of machine translation one of the very prominent people in work with machine translation was Yehoshua Bar-Hillel. And Yehoshua Bar-Hillel actually focused in on this problem of word sense disambiguation and he posed the following the problem. Look at this little text. "Little John was looking for his toy box. Finally, he found it. The box was in the pen. John was very happy." Where he's focusing on the

ambiguity of pen meaning is it the writing implement pen or a pen for having kids in, which I think is not a very current usage. People still put walls around their children to keep them in certain parts of the house, but I think these days they don't generally refer to it as a pen. Bar-Hillel essentially declared that this task of working out which sense of pen was in use in this context, which would be required in most cases so that you could translate correctly into another language was an unsolvable problem and he was so convinced that it was an unsolvable problem that he left the field of machine translation and went off and did mathematics. So what he writes is assumed for simplicity sakes, the pen in English has only the following two meanings; a certain writing utensil or enclosure.

I know claim that no existing or imaginable program will enable an electronic computer to determine the word pen in the given sentence within the given context has the second of the above meanings, whereas every reader with a sufficient knowledge of English will do this automatically. So a lot of recent work in statistical NLP has, essentially, argued this Bar-Hillel guy, he was a bit crazy. We can kind of just slurp a lot of text and look in the context of which words and we can work out the senses of words perfectly well. Bar-Hillel actually states, "Let me state rather, dogmatically, that there exists at this moment no method of reducing the polysemy of the, say, twenty words of an average Russian sentence in a scientific article below a remainder of, I would estimate, at least five or six words with multiple English renderings, which would not seriously endanger the quality of the machine output. Many tend to believe that by reducing the number of initially possible renderings of a twenty word Russian sentence from a few tens of thousands (which is the approximate number resulting from the assumption that each of the twenty Russian words has two renderings on the average, while seven or eight of the have only one rendering) to some eighty (which would be the number of renderings on the assumption that sixteen words are uniquely rendered and four have three renderings apiece, forgetting now about all the other aspects such as change of word order, etc.) the main bulk of this kind of work has been achieved, the remainder requiring only some slight additional effort."

Bar-Hillel then goes onto argue that the program is no, there are a bunch of easy cases that you can do. There are these residue of hard cases that he can't see how automatic methods will be able to get them right. And really, if you look at the current state of the art of statistical and empirical methods and NLP, they're really kind of at the level that Bar-Hillel's talking about. But never the less, in the history of NLP, in the early days of NLP, this was one of the many problems that people tried to approach with deep AI. So, essentially, people built expert systems whose job it was to determine the senses of the word. Small and regal have the dubious distinction of building such an expert system and they're often [inaudible] in modern statistical NLP writings because they were so unwise as to write the word expert for throw is currently six pages long. That's six pages of list code, but should be ten times that size. So compared to that then when statistical methods came along, they showed great promise because they gave the opportunity of providing that you had some supervised training data that you could do automatic disambiguation with high success. So the alternative approach of the statistical NLP, the quotation that's most not commonly referred to is this work of Firth. So Firth was a British linguist in the

30s and 40s whose work is actually very little known in the United States, but has been kind of picked up by statistical NLP people for saying, “You shall know a word by the company it keeps.” I actually also rather like Wittgenstein’s later writings that relate to this point and he writes, “You say the point isn’t he word, but its meaning, and you think of the meaning as a thing of the same kind as the word, though also different from the word. Here the word, there the meaning. The money, and the cow that you can buy with it. But contrast money and its use.” I don’t actually know what that means, but in another passage he says something more understandable. He says, “For a large class of cases, though for all, in which we employ the word “meaning” it can be defined thus the meaning of a word is its use in the language.” So Wittgenstein’s’ later writings is attributed with advocating for this position of a use theory of meaning where the representation of a meaning of a word is just the context in which it appears. The knowledge of the meaning of the word means that you know which context. He can say whether a word is appropriate in a context or not.

This is kind of what you get in semcor, so this is a boring piece of construction text that’s talking about something slipped into place across the roof beams and it’s giving the sense of words like slip place, roof beams in terms of WordNet senses. So how can you do word sense disambiguation? One kind of approach that doesn’t require supervised data that I should just briefly mention is using dictionaries. The method that is most commonly cited is Lesks’s Method. This is Mike Lesk, who you might know from other context like information retrieval and [inaudible] libraries. So the Lesk algorithm was essentially that you get definitions from a dictionary and you use a word overlap measure in the definitions and use that to attribute senses. So suppose I want to disambiguate words “pinecone” I can look up pine, it has two senses, kind of evergreen tree of needle shapes leaves and waste away through sorrow or illness; and cone has the mathematical sense, something of this shapes, fruit of certain evergreen trees. Another way to distinguish information is frequency. Notwithstanding what I said about the different senses of plant and [inaudible] being potentially misleading, it turns out that for most words, at least relative to a particular text type, that the usage is just extremely, extremely skewed. So a word like cell has a bunch of different meanings, but if you reading any biology journal boy is it skewed what sense that you’re gonna get. If you can just use the most common sense in the genre that’s a very, very strong source of information.

In WordNet they put the most common sense of the word first in some kind of generic, non domain specific sense. And it turns out that if you’re dealing with rare words for which you have very little training data, which is a lot of words most of the time, that this WordNet first sense heuristic turns out to be just a very strong baseline. There are sorts of creative usages, so, “In his two championship trials, Mr. Kulkarni ate glass on an empty stomach accompanied only by water and tea.” Well, there’s someone eating something wouldn’t normally be called a food stuff. But a lot of the information you have isn’t classical selection or restrictions. Commonly, if you just know the topic of the article that’s worth a lot too. And so that lead to modern statistical work in computational linguistics. In the starting off of statistical methods and computational linguistics there are essentially two key endeavors; one was the machine translation work that started at IBM and the other which was worked under the AT & T [inaudible] largely lead by Ken

Church. Essentially where they started was doing word sense disambiguation, but actually they were doing word sense disambiguation for the purposes of machine translation. So the method that they used was naive based classifiers, which at that time, counted as sort of something very new for most of the computational linguistic AI community. So you have a high probability of a sense and then you have the probability, the context given the sense, and the context probability is just being estimated by taking the probability of different words occurring in that context. But the words are just being modeled as a bag of words so you've got some context window and you're just generating all the words in that context as a multi nominal classifier. So actually the application was machine translation as well and so one clever way of getting training data for word sense disambiguation which is also application relevant is actually to use parallel data because if you're doing MT, you want to learn about word senses that get translated differently and don't really need to learn about word senses that don't get translated differently. So people were extremely, extremely impressed because they showed just extremely good results from doing this for these kind of word senses, so results of over 90 percent accuracy.

An important thing to notice is that these results are for distinguishing two extremely distinct senses of a word. Now, in some sense, you might say well this is the main task that I want to deal with. I'm not really interested in some of those fine grain senses. I only want to know about core senses of things that translate differently and that are really important. And I think the answer is that in a lot of cases with those kind of cases you can get high accuracies in word sense disambiguation, nevertheless, the funny thing that's happened in word sense disambiguation is a lot of more recent work has shifted to looking at much more fine grain senses of the kind found in places like WordNet. So the accuracy has then kind of gone south because if you're trying to distinguish 10 different senses of the word stock, of the kind that we saw beforehand, some of which are very similar to each other, those are way more difficult tasks and you get much worse results. Then I'll show those results in a minute, but before I do, I'll just show you a few little bits of data analysis that are kind of interesting. So in the work with Ken Church and AT & T, his main collaborator was William Gale who was actually a statistician. So one of the nice things about this work, and I think something good that statisticians always do is explore data analysis whereas people in computer science are very bad at doing exploratative data analysis. So their early work in word sense disambiguation actually has some nice little graphs that are just showing interesting properties of the task. So what this graph is showing is let's suppose I have a 10 word window of context on each side of the word I want to disambiguate. I start off here with using the 10 words to the left and the 10 words to the right and then I kind of move further out. So I kind of use words 11 to 20 to the left and right, and then I use words 21 to 30 to the left and right and then 31 to 40 and keep moving out. And so this is a log scale so this is where we're now a 100 words out, a 1,000 words out, 10,000 words out and the kind of interesting result that you get here is probably the best if you use adjacent words as the context. You can be out to almost 90 percent accuracy. So even out at 10,000 words you're doing vaguely above a random baseline.

What this shows is just how much power there is in general in using the context very generally, which essentially then becomes the topic or just the general subject matter of the article. This one then asks the question of well, how big – you saw that the nearby context was very useful, if you make the context bigger, does that help? And so their results were that for a 10 word window you're getting about 87 and as the context size grows and so you've got a 50 word window on each side, you're getting a bit over 90 percent accuracy. And then as you go out beyond there, it's kind of largely flat and just bounces around a little. So this result was taken from my people to mean use the big wide context, use about 50 words on each side to influence your word sense decisions. And for what they were measuring whether just using these bag of words, Naive Bayes classifiers, I mean, that is kind of the right answer. You can just estimate topical associations better with fine useful pointer words. That's a position that's somewhat being refined in later work as I'll mention in a moment. And then this is the learning curve, which is how much data you need to see to do how well. The result from this was that, at least for this, their course [inaudible] word sense disambiguation, you could do quite well with a reasonably small number of examples. Okay. There's been a ton of other work on word sense disambiguation including boot strapping methods to reduce data. I won't go through that in detail; I'll just mention a couple of things down at the bottom here.

These were two principles suggested by David Yarowsky. These two are kind of related, so I'll do this one first. One sense per discourse was his claim that, in general, in a discourse, a piece of text, an article or something, you'll only find one sense of a word. A lot of the time that's true. If an article is using a word in one sense, it just won't use it in any other senses. Later work has refined that claim a little. I think, commonly, what you find is this is true for noun senses, it isn't true for verb senses, that articles can easily use verbs in different senses. One sense per collocation kind of connects up with this general notion of collocations, so Gale and Church did everything just with these bag of words features, but I think modern understanding is that, as well as having these kind of broad topical features, it's just really useful to have specific features that says what is the word to the left and what is the word to the right and often people will say look at the second word to the left and the second word to the right because it just turns out that there are a lot of very particular collocations that choose one sense. So you'll have an expression like "laughing stock" and well, if you just see the word laughing to the left of the word stock, it's always gonna be this one particular sense and if you kind of stop paying attention to all these context words, well, there might just be too many words of that plant or who knows what in a particular text and it'll only confuse you and get it wrong. So, by and large, if there's a clear collocation, it nearly always chooses the same sense and so you also want to pay a lot of attention to that close [inaudible] collocation information.

Rushing ahead. Right. So baselines for word sense disambiguation commonly people use most frequent sense. Sometimes people regard Lesk [inaudible] as a baseline, upper bound, how much humans agree. So rigorous evaluation of word sense disambiguation for these sort of many subtle senses has taken place in senseval. So the task for a senseval one is taking a word like "horse" and distinguishing between a whole bunch of senses or the ones listed in senseval. People have done that both for all words and for a lexical sample. I'll just show you the lexical sample results. So the lexical sample results are

essentially on difficult words that have many senses. So the average number of senses in WordNet for the words that were tested was nine. So that's kind of how these subtle many sense in WordNet words – but often many of those senses are related together and hard to tell apart. So these are the kind of results you get. You probably can't read it well, but down here it says, Stanford cs224n because many years ago we used to use word sense disambiguation as one of the projects and then me and a couple of others took all that cs224 and WordNet, word sense disambiguation systems and tied them together into a classifier combination and entered into senseval. We actually do pretty well because we came in fourth place doing that. The best performance here was 64 percent accuracy and Stanford was doing 61.7 close to that.

So the positive result there is how state of the art the systems that we produce in cs224n are. But the negative result is in some sense, at least for this task of trying to recover WordNet sensors, is really, really difficult and people can't do it. But I think many people now believe that this just is too hard a task, and kind of an uninteresting one because a lot of those fine senses might not matter much. Okay. So then let me go on and touch a couple of other topics. So there's been lots of interest in lexical acquisition of how can we then kind of acquire something about the meaning of words. One way of understanding word similarity is, again, to go straight throughout the source. We could go to WordNet and say, "Well, can we work out meaning similarity in that?" I think I'll kind of quickly skip past that, but the general idea is if you have a hierarchy from WordNet, we should be able to tell what words are similar. But there are lots of problems with that. One of the biggest problems is coverage. Lots of stuff you just won't find in WordNet so here's some kind of a list of words that you don't find in WordNet. Okay. So the alternative is to come up with a representation of word meaning and word similarity that you can introduce much more automatically. So this leads into the area of vector space space lexical semantics. There is another [inaudible] vector space space lexical semantic. There's also been quite a bit of recent work in doing probability simplex based lexical semantics, but what I'll say today is I'll just say a little bit about vector space space lexical semantics. In some sense, this is an old idea. It goes back into linguistics as well. There's been this kind of idea of having word features, which is referred to as componential semantics so you can have various vector dimensions and then you can say, "Well, dog is animate, but it eats me to social," "horses are animate, eat grass, and social," you can then have similarity inside this kind of binary vector between different words.

In some sense, what people do with vector based lexical semantics is like that, but more quantitive. So the general picture for vector based lexical semantics is you have some properties, which are normally distributional properties so you can learn it unsupervised from a lot of data, you turn each word into a vector and then if you only want to do word similarity, you just use those vectors for word similarity and if you want to create clusters of words, you then perform some kind of clustering. Okay. So once you have some word vectors, you can use similarity measures. The traditional ones are using co-sign or you put in distance, which is equivalent providing you're normalizing your vectors to be unit vectors. There's then been some threads of work which actually suggest that you don't do as well with these measures and you do better with measures such as an L1 measure or

some of the various probabilistic measures that are being used. I mean, the sense of that seems to be that the kind of squaring operations that you're using in these measures don't actually make terribly make sense if you're thinking about word count data and that you're doing better with something like an L1 metric. Okay. So here's just an example of the kind of results you get out of that. So this is the Burgess and Lund Model which was used for psycho linguistic purposes. There's a 160 million word corpus, context of the most frequently occurring words. It's commonly the case when people build these models that are for counting things in the context you only use some number of common words. That's just kind of a crude way of stopping your matrix from getting too huge and then practice doesn't really affect accuracy because unless you've seen a word a bunch of times, it's not really very useful as a context clue. Co-sign, 10 word window. I mean, it's a pretty good list they get out. So this is saying the word before the colon, what are the most similar words to it. So scared, upset, shy, embarrass, anxious, worried, afraid; harmed, abused, forced, treated, discriminated, allowed; Beatles, original band, song, movie, British, lyrics. This is typical for what you get from distributional similarity. This is kind of all good as topically associated with Beatles; this is just general distributional similarity. The word most similar to frighten is scared, that's good.

Then a couple after that kind of aren't quite so good, right, upset and shy don't seem – they're kind of emotions that are negative, but they're not so similar to frighten. It works reasonably, but it's hard to get results that are perfect. Another very, very, very well known form of doing this distributional similarity is Landauer Latent Semantic Analysis, which is then using SVD to do dimensionality reduction of your vector and then doing similarity in the reduced space. The claim is, and I think the claim is somewhat true, Landauer sometimes makes some very, very strong claims for LSA, which I think aren't really true, but we claim that you commonly can get some mileage from doing dimensionality reduction in measuring word similarity, I think is true. Okay. Okay. So that's a kind of a general method of sort of just taking this sort of soup of words and working out word similarity. Before time runs out, I thought I'd then just say sort of something about a rather different way of doing unsupervised – it's a sort of learning that you can do over large amounts of text at any rate that has also been explored including by a student at Stanford, Ryan Snow, which is trying to do a much more specific form of learning over large amount of text. So the idea here is what we want to learn is about new hyponyms so new is a kind of links. The motivation for this is we can't just use WordNet now as a kind of links because it just doesn't have very good coverage when it comes down to it.

So if you sort of look at something like these nominalizations like custom ability [inaudible]. Some of them are in WordNet, combustibility and [inaudible], but other ones like affordability, reusability and extensibility aren't in WordNet, but those are words that everyone knows. So can we learn hyponym relationships automatically? This was a field that was, essentially, pioneered by Mary Hearst, who's at Berkeley, and what her observation was is that there's just lots of sentences that, essentially, tell you hyponym relationships. So rather than trying to do some very, very clever form of distributional similarity with some kind of statistical filtering, getting high quality results from that, why don't we instead go for a high precision approach and run it over bust [inaudible]

and essentially just look for the sentences that tell us about hyponym relationships or tell us about synonyms relationships and there are lots. So what Hearst was hand wrote patterns that would find examples of those things.

NP0 such as NP1, NP2 and/or NPI, those are hyponyms. And so she wrote a handful of regular expressions. I seem to only have five on this list, but I remember there were six of them that were kind of obvious ones. So there's XY and other things, so temples, treasuries and other important specific buildings; the such as ones including all common law countries, including Canada and England, and especially animals, especially cats and dogs; so she ran that over a lot of text and learned patterns quite successfully. So more recently, Ryan Snow has been trying to do that in less hand specified, more machine learning based ways. So what he's doing is passing up sentences to give dependency parses like this and then potentially saying any par in the dependency parse could be a pattern for learning examples of things. So if I have a pattern like this one between oxygen and elements, that's a potential pattern and this pattern between abundant and oxygen, that's a potential pattern. I want to learn patterns as a good indicator of things being a hyponym. How will I boot strap this? I'll boot strap this by using known hyponyms from WordNet and then I'll try and acquire other hyponyms. So these are the details of their algorithm. So they collect a huge number of noun pairs, they find positive and negative examples that are being hyponyms using WordNet, they parse them all, they train a hyponym classifier and turn it into a logistic regression with plus and minuses for whether they're good patterns or not. So in total, they define 70,000 patterns that are automatically defined. The question then is how good are these patterns? That's the graph over here. This is the precision which is of all the times the pattern matched, how often was it a real hyponym relationship?

And the kind of interesting thing is that these red marks show the patterns that were the hand specified Mary Hearst patterns so Mary Hearst, she thought through it correctly or maybe did a little bit of corpus research, so essentially, I mean, Mary Hearst found all the best patterns that were the kind of [inaudible] patterns of having reasonable precision and recall. She also had one pattern that was a bit of a dud, but that was pretty good going when it comes down to it. But the interesting this is that there are a bunch of other patterns so she didn't have [inaudible] pattern, but that pattern is a reasonably high precision and recall pattern. I did have three slides on one more topic, but I think maybe I'll just say that's it for today. I think I'll call it the end for the day and say that's my tour of lexical semantics and so then there's the one more lecture on Wednesday, which talks about question answering systems.

[End of Audio]

Duration: 75 Minutes