



AUT

🎓 Text Mining Quizzes

Post on: Apr 9, 2024 Last edited: Apr 9, 2024 31 Views

Quiz 1

Question 1:

The subtle difference between the words play, player, played playing is described by which of the following terms?

- semantics
- phonetics
- syntaxics

morphology

Question 2:

Which one of the following is not a reason for studying Text Mining?

~~Take advantage of vast amount of information encoded as numerical data.~~

- Take advantage of the vast amount of information encoded as texts, both online and offline.
- To communicate with machines.
- To interface with organized information in databases.



wenyang blog

nonyms and coreferences are tools used for:

- Expressing the same meaning.
- Joining parts of a sentence.

~~Cohherence~~

- Expressing multiple meanings.

Question 4:

"The river's bank was breached by the flood" and "My loan got approved by the bank" are sentences emulating _____ ambiguities.

Correct Answer: lexical

Question 5:

Which one/s of the following is true about python in the context of this course.

~~It is a programming language extensively used in the commercial sector.~~

- Its an IDE.
- Its a programming language for starters.

~~Its a simplified programming language with an abundance of data and text processing libraries.~~

- Its a type of snake which will be studied in detail in this course.
- Its the only programming language that can be used to do NLP.

Question 6:

Which one/s of the following is true about POS tagging and parsing.

~~Parsing is a higher level task compared to POS tagging.~~

- Parsing and POS tagging are mutually dependent on each other.

~~Parsing is based on POS tagging.~~

- POS tagging is based on Parsing.
- POS tagging and Parsing are essentially the same thing.

Question 7:

What are semantic ambiguities?

~~Words with more than one sense~~



wenyang blog



~~Different words with the same spelling~~

- Words with different spelling.

~~Polysemous words~~

Question 8:

Why is NLP considered a "hard" problem to solve?

- Natural language occupies a lot of space on the storage device.
- There is a very large amount of data stored as natural language.
- Knowledge represented as numbers can be ambiguous while that represented as natural language is not.

~~Natural language knowledge representation is context based.~~

~~NLP is multidisciplinary in nature~~

Question 9:

What is the significance of "natural" in natural language processing?

~~Its language as produced by humans.~~

- Its language as produced by only native speakers of a language.
- It deals with text stored natively in a computer.
- It only deals with language produced by indigenous communities of the world.

Question 10:

Which one/ones of the following describe syntax of a language?

- How words are vocalized.

~~How words are put together to form correct sentences~~

- How words are constructed.
- How meaning is constructed from words.

Question 11:

What is the difference between Text Mining and Natural Language Processing?

- They are the same thing.



wenyang blog

~~processing algorithms.~~



- TM involves numerical data as well while NLP doesn't.
- NLP includes numerical data as well while TM doesn't.

Question 12:

Which one/ones of the following explains the difference between text mining and natural language processing (NLP)?

- Text mining is applied while NLP is theoretical.

~~Text Mining uses algorithms and results developed by NLP~~

~~Text mining is applied while NLP is more fundamental.~~

- They are the same thing so there is no difference.
- NLP uses algorithms and results developed by Text Mining.

Quiz 2

Question 1:

Which one/s is an example of a situation when a word could be different than a token?

- For a word such as "pre-processing"

~~For a word such as "couldn't"~~

- For a word such as "New Zealand"
- For words such as "Ministry of Health"

Question 2:

Which one/s of the following describes n-gram language model?

- It is a model based on documents as a whole.

~~It is a model in which clusters of words are extracted and these clusters are counted and used in the model.~~

- It is a model based solely on tokens.
- It is a model based on basic units of a language, for English, it is the alphabets.

Question 3:

Which one/s of the following is not a valid dimension reduction strategy for language

wenyang blog

~~removing all nouns.~~



- Using just singular forms of words.

~~Removing all adjectives.~~

- Using words Stems.

- Using just word Lemma.

Question 4:

Which one/s of the following is true about regular and irregular words?

- The past tense of "go" is "went" is an example of regular verb.

- The plural of "car" is "car" is an example of irregular noun.

~~Regular words follow the general rules and irregular ones don't.~~

- The past tense of "play" is "played" is an example of irregular verb.

~~The plural of "fish" is also "fish" is an example of irregular noun.~~

Question 5:

Which one/s of the following is true about morphology?

- It is used relevant only in the English language

- It can modify a word by adding characters in the middle of a word.

~~It is a tool used to change the meaning of a word slightly by adding suffixes and prefixes.~~

~~It is used in some form in all languages.~~

Question 6:

Which one/s of the following is true about inflectional and derivational morphology?

- In inflectional morphology, the class of the word may change as long as the stem of the word remains the same.

~~In derivational morphology the class of the word may or may not change(friendly,unfriendly).~~

~~In inflectional morphology the class of the word does not change.~~

- The words "play", "player", "played" is an example of inflectional morphology where the words remain in the noun class.

wenyang blog

"play", "playing", "played" is an example of inflectional morphology



where the words remain in the verb class.

- In derivational morphology, the class of the word remains the same.

Question 7:

Consider the sentence below:

"I was sitting on the park bench drinking water from a"

In order to predict the next word after "a" which of the following conditional probabilities can be used?

~~P(glass | park bench drinking water from a)~~

- P(bottle| I sitting park bench drinking)

- P(glass | park bench drinking)

~~P(bottle | water from a)~~

Quiz 3

Question 1:

According to the confusion matrix below which one/s of the following statements is true.

		Machine Computation				
		VB	TO	NN	Totals	
Ground Truth	VB	127	20	15	162	
	TO	10	312	400	722	
	NN	7	10	50	67	
	Totals	144	342	465		

- The machine easily confuses VB as TO.
- The machine easily confuses the NN as TO.

~~The machine is not good at identifying between the NN and TO categories.~~

**Question 2:**

A student used the following code and pattern matcher in an attempt to extract clause patterns of the form <pnoun> <verb> <noun>. However, he got the following output with the empty array for the result.



Johnny / PROPN Jumped / VERB in / ADP the / DET river / NOUN []

Which one/s of the following could be the possible reason for this?

matlab



The defined pattern is not present in the sentence being tested.

- The result array is not initialized, hence it's empty.

If the sentence being tested was "Johnny jumped river", the output would be correct.

- The tagging is incorrect hence the results array is empty.
- The pattern definition for <pnoun> <verb> <noun> is incorrect.

Question 3:

Is the following item True or False?

A token in a language vocabulary necessarily has a unique POS tag.

- True

False

Question 4:

Which one/s is true about True positives.

Are identified data points in a category which has also been identified in the category by humans.

- Are data points from the category which are not identified into the category by the machine.

Are identified data points in the category which are correctly identified by the machine.



**Question 5:**

Which one/s are true about closed and open class words?

- Adjectives are a closed class.

~~Pronouns are a closed class.~~

~~It is difficult to POS tag open class words.~~

~~Adjectives are an open class.~~

- It is easy to POS tag closed class words.

~~Open Class words are productive that's why it "open"~~

Question 6:

Which one/s of the following could be a possible output from the code run below?

python



- COMP700 Quiz [Quiz] Auckland [Auckland] New Zealand [Zealand]

~~COMP700 Quiz NP [Quiz] Auckland NP [Auckland] New Zealand NP [Zealand]~~

- COMP700 Quiz NP Quiz Auckland NP Auckland New Zealand NP Zealand

- [COMP700 Quiz NP Quiz] [Auckland NP Auckland] [New Zealand NP Zealand]

Question 7:

Use the following confusion matrix to compute the Precision value for VB.

		Machine Computation			
		VB	TO	NN	Totals
Ground Truth	VB	127	20	15	162
	TO	10	312	400	722
	NN	7	10	50	67
	Totals	144	342	465	

wenyang blog

 0.78 0.91

0.88

Question 8:

Which one/s of the following is true about Treebank grammar?

~~It might not cover all possible types of sentences~~ Treebank grammar consists of rules defined by humans. Treebank grammar covers all possible types of sentences.~~The rules in Treebank grammar cover all the sentences in the corpus from which the rules are derived.~~**Question 9:**

Which one/s of the following is NOT true about the following rules?

 $NP \rightarrow Det\ Nominal$ *$NP \rightarrow ProperNoun$* *$Nominal \rightarrow Noun \mid Nominal\ Noun$* A Nominal can be preceded by a Determiner.~~A NP can be a determiner followed by a Proper Noun~~~~A Nominal can be a Noun followed by another Nominal Noun~~ A Nominal can be a Nominal followed by a Noun A Nominal can be a Noun**Question 10:**

Which one/s of the following is true about the dependency parsed tree?

~~"ate" is the main verb~~ "icecream" is the subject of the clause. Spoon is a prep object of "John"~~Spoon is a prep object of "with".~~

wenyang blog ~~rep object of "icecream"~~

**Question 11:**

Use the following confusion matrix to compute the Recall value for TO.

0.43

Question 12:

Which one/s of the following is true about Named Entity Recognition?

~~It can be used to identify people.~~

~~It is the process of identifying different types of entities in texts.~~

- It the process of assigning names to all the tokens in the text.
- It is the process of identifying POS tags in text.

Question 13:

When the code below is run, the output is: [bus, into, with]

Which one/s of the following is true about the code and the output.

python



- The main verb's head node has 3 children

~~The main verb has 3 children.~~

~~The main verb has 3 children, out of which one of them is the subject.~~

- The main verb should have 4 children, but it outputs 3 meaning one of them has been missed by the model.
- The main verb has 2 children and one subject.

Question 14:

Which one/s of the following are true about false negatives?

~~Are data points from the category which has been incorrectly identified by the machine.~~

~~Are data points from the category which has not been identified into the category by the machine.~~

- Are data points from a category which has been incorrectly identified.

points from a category which had been incorrectly identified into the wenyang blog category.



Question 15:

Kappa statistic is used for:

~~Express the level of agreement usually by human evaluators.~~

- Is used to compare agreement between machines from different categories.
- Is used to compare agreement between 2 machines.
- Is used to compare human and machine agreement.

Question 16:

Which one/s of the following is part of a possible output if the following method is called on the sentence "I went to town on a bicycle"

python



- bicycle/VBD <--pobj-- on/NN
- a/NN <--det-- bicycle/VBD
- town/NN <--VBD-- to/IN

a/DT <-- det -- bicycle/NN

I/PRP <-- nsubj -- went/VBD

- I/PRP --nsubj--> went/VBD

Question 17:

What is the main difference between Treebank grammar and CFG?

- Treebank grammar is recursive hence simpler.
- CFG is recursive hence simpler
- Treebank and CFG are both recursive but the recursions in Treebank are simpler.

~~Treebank grammar is not recursive, hence simpler~~



Question 18:

Which one/s of the following is part of a possible output if the following method is called on the sentence "I went to town on a bicycle"

python



wenyang blog ~~t-bicycle/NN~~

- a/NN <--det-- bicycle/VBD
- I/PRP --nsubj--> went/VBD
- bicycle/VBD <--pobj-- on/NN

~~I/PRP ← nsubj went/VBD~~

- town/NN <--VBD-- to/IN

Question 19:

Which one/s of the following is NOT part of CFG?

- Rules
- Terminals

~~POS tags~~

- Non-terminals

Question 20:

Which one/s of the following is true about Dependency grammar?

- Dependency grammar framework defines relations between noun phrases.
- The relations in a dependency grammar framework are same as CFG grammar

~~It defines binary relations between tokens.~~~~The nodes consist of the tokens.~~**Question 21:**

What is Chunking?

~~It's the process of identifying noun phrases~~

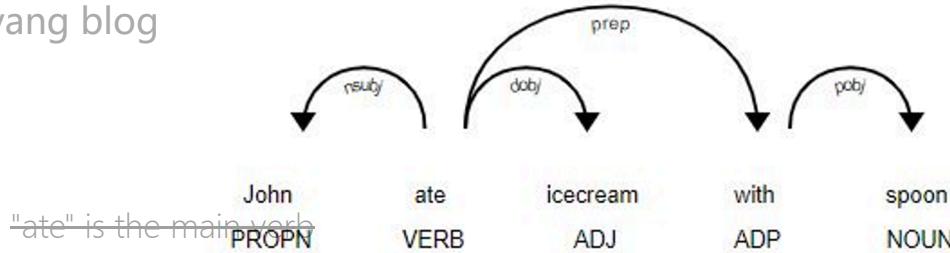
- Its the process of identifying the Dependency grammar tree for a sentence.
- Its the process of identifying verb phrases
- Is the process of identifying clauses in sentences.

Question 22:

Which one/s of the following is true about the dependency parsed tree?



wenyang blog



- "icecream" is ... subject of the clause.
- "with" is a prep object of "icecream"
- Spoon is a prep object of "John"

~~Spoon is a prep object of "with".~~

Question 23:

Which one/s of the following is true about POS tagging?

- It's the process of classifying all the words in a language vocabulary into a given set of tags.
~~It's the process of classifying each token in text into a given set of tags.~~
- It's the process of assigning the syntactical function to a particular instance of a token in text.
- Its the process of assigning the syntactical structure to text.
- It is the process of ranking tokens in text.

Question 24:

Is the following statement True or False/

A Token in a sentence has a unique POS tag.

~~True~~

- False

Quiz 4**Question 1:**

Which one/s of the following is true about BOW model

- BOW model will not be able to capture plural and singular words.
- BOW model is not able to capture stop words.

~~BOW ignores the order of words in a sentence.~~

wenyang blog ~~ees "The dog bit the man" and "The man bit the dog" would have exactly the same model.~~

- The sentences "The dog bit the man" and "The man bit the dog" would have approximately the same model.

Question 2:

Consider the following corpus of 3 documents and 4 terms..

Term	Doc 1	Doc 2	Doc 3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

What is the value of inverse document frequency using the formula $\log(N/dft)$ for the term "insurance" for Doc 3?

- 29

~~0.18~~

- 1.5

- 0

Question 3:

Which one/ones of the following is true about the TF-IDF weighting scheme?

- It decreases with the rarity of the term but increases with the frequency of the term.

~~It increases with the frequency of the term as well as the rarity of the term.~~

- It decreases with the frequency of the term but increases with the rarity of the term.

- It decreases with both the rarity of the term as well as the frequency of the term.

Question 4:

Which one/s of the following is true about vector space model of a document?

- It's the representation of the document with the tokens.

- It's the POS representation of a document.

wenyang blog



bolic representation of a document.

~~It's the representation of the document with a data structure.~~

~~It's the numerical vector representation of a document.~~

Question 5:

Use the following 3 document, 4 term corpus and the Euclidean formula to calculate the Euclidean Distances between Doc 1/Doc2 and Doc1/Doc3. Choose the option that indicates the values in the order: difference between Doc 1/Doc2 and difference between Doc1/Doc3

$$D_E(\overrightarrow{ta}, \overrightarrow{tb}) = \sum_{t=1}^m |w_{t,a} - w_{t,b}|^2)^{1/2}$$

Term	Doc 1	Doc 2	Doc 3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

38 and 100

74 and 100

100 and 38

~~52.10 and 29.46~~

Question 6:

Which one/s of the following is not a dimension reduction strategy?

Latent Semantic Analysis

~~POS tagging~~

~~tokenisatoion~~

Stemming

Question 7:

For the following document matrix what would be the term frequency vector for

wenyang blog



Term	Doc 1	Doc 2	Doc 3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

 1110~~4 33 33 0~~ 27 3 0 14**Question 8:**

A corpus has 1,000 documents with 50,000 terms. Use the following diagram from lectures in conjunction with the corpus information to determine which one/s of the following is true.

~~The number of tokens in the corpus can be 50,000~~~~The value m in the diagram is $\geq 50,000$~~

The corpus information as provided is impossible, as we cannot have 50,000 terms in 1,000 documents.

~~The value n in the diagram is equal to 1000.~~

The value n in the diagram is equal to 50,000

~~The value m in the diagram is equal to 50,000~~~~Number of tokens in the corpus $\geq 50,000$~~ **Question 9:**

Which one/s of the following can be used to calculate TF-IDF values?

$$tf - idft, d = \ln(1 + tft, d) \times \ln(N/dft)$$



$tf - idft, d = tft, d \times \log(N/dft + 1)$

$$tf - idft, d = tft, d \times \ln(N/dft)$$



wenyang blog, $d = \log(1 + tft, d) \times \log(N/dft)$



$\sqcup tf - idft, d = tft, d \times \log(N + 1/dft)$

Question 10:

One of the simplest vectorization techniques is to use the raw term frequency. Which one/s of the following is true about term frequency?

- A frequently occurring term has more information content than a rare term.
- A document which a term 100 times more is necessarily 100 times more relevant.

~~A document which has a term 100 times more is not 100 times more relevant.~~

~~Rare terms are more informative than frequent terms.~~

Quiz 5

Question 1:

Word Embedding can be used to rank documents.

~~True~~

Question 2:

Word Embedding CANNOT be used to compute the difference between two documents.

~~False~~

This is the code used in this Quiz

Question 3:

In the code below how many dimension embedded matrix will be generated?

javascript



~~300 (EMBEDDING_DIM = 300)~~

- A random number
- Cannot be determined from the code snippet.

Question 4:

Use the following code snippet in this question. What is the purpose of the code from lines 27 to 29?

javascript



wenyang blog ~~the input layer input and the hidden layers of the neural network~~



- It defines the output layer and the hidden layers.
- It defines only the hidden layers.
- It defines only the input layer.

Question 5:

Use the following code snippet for this question. For a 10 word vocabulary, how many dimensions would the vector `y_label` on line 23 be?

javascript ▼

Answer choices:

- Can not be determined from the code.

10

- Same as the number of hidden neurons

- 1

Question 6:

Use the code snippet below for this question. What is the purpose of the code on line 36?

javascript ▼

~~It computes the error for each of the inputs and finds the mean.~~

- It computes the error for a single input.
- It sums the output for each of the inputs.
- It sums the weights for each of the neurons.
- It sums the inputs and finds the mean.

Question 7:

Consider the following 3-dimensional representation for the terms plant, tree, shrub:

	plant	tree	shrub
1	2.5	2.4	4.1
2	2.5	2.6	3.8



wenyang blog

3.2

3.0

3.4



Use the dot product between vectors to compute the similarity between the terms. Which one/s of the following is true?

~~Tree and shrub are more similar compared to tree and plant.~~

- Plant and tree are more similar compared to plant and shrub.

~~Plant and shrub are more similar compared to plant and tree.~~

- Plant and tree are more similar compared to tree and shrub.

~~Plant and shrub are more similar compared to tree and shrub.~~

The dot product between two vectors is a measure of their similarity; **a higher dot product indicates more similarity.**

Let's calculate the dot products:

1. Plant and Tree: $(2.5 * 2.4) + (2.5 * 2.6) + (3.2 * 3.0) = 6.0 + 6.5 + 9.6 = 22.1$

2. Plant and Shrub: $(2.5 * 4.1) + (2.5 * 3.8) + (3.2 * 3.4) = 10.25 + 9.5 + 10.88 = 30.63$

3. Tree and Shrub: $(2.4 * 4.1) + (2.6 * 3.8) + (3.0 * 3.4) = 9.84 + 9.88 + 10.2 = 29.92$

Comparing these results, we find:

- Plant and Tree: 22.1
- Plant and Shrub: 30.63
- Tree and Shrub: 29.92

So, the correct statement is:

Plant and Shrub are more similar compared to Plant and Tree.

Question 8:

Which one/s of the following is true about a million-term vocabulary and word embedding?

- It can only be represented by a 1,000,000 dimensional matrix.

~~It can be represented by a 1,000,000 dimensional matrix.~~

- It cannot be represented by a matrix smaller than 1,000,000 dimensions.

~~It can be represented by a 2-dimensional matrix.~~

wenyang blog



Which one/s of the following is true about embedded vector generation from neural networks?

The weights for the hidden layer neurons is used as the term representation for a given dimension.

- The number of hidden layers neural network is equal to the number of terms.

The number of output neurons is equal to the number of terms.

- The trained neural network is used to generate the embedded matrix in the output layer.

The number of input neurons in the network is equal to the number of terms.

The neural net is trained using one hot representation.

- The number of output neurons is equal to the the number of dimensions in the embedded matrix.

Quiz 6

Question 1:

Which one/s of the following is NOT true about the nearest neighbor classifier?

It is the only algorithm that does not need similarity computation.

- Multiple neighbors can be used to decide the category of the test object.
- The algorithm needs a similarity function for computation of the distance between neighbors.

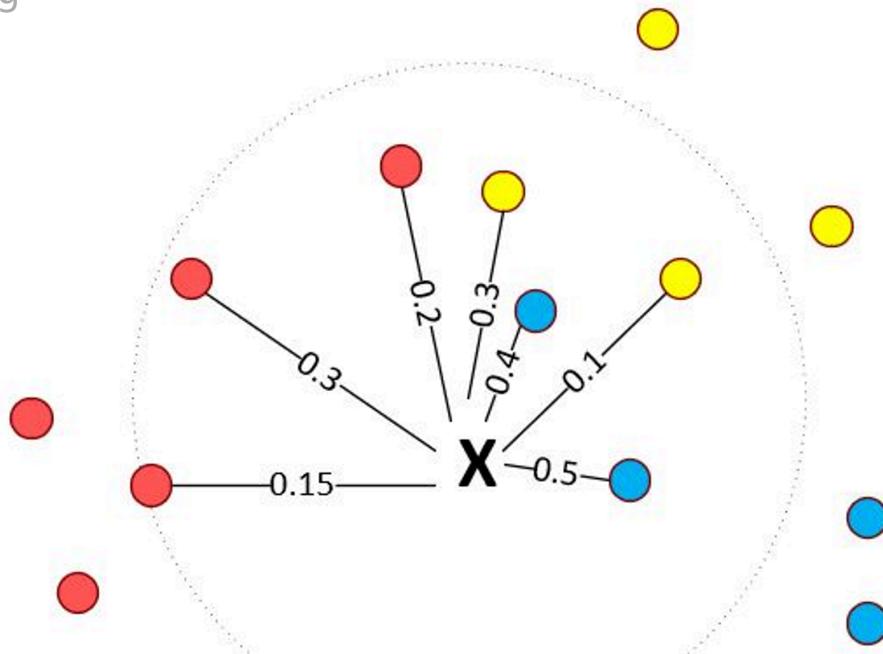
Only one neighbor can be used to decide on the category of the test object.

The algorithm can only be used for classification problems with a single feature that is used to compute the distance between neighbors.

Question 2:

The diagram below represents a KNN classification scenario where the numbers on the lines represent the weight according to its nearness to the object (X) to be classified. Using K=7 and a weighted voting scheme, what would the object X be classified as?



 F~~Blue circle (weight summary is the highest)~~ I Yellow circle It would be a tie between Yellow circle and Blue circle.**Question 3:**

In the diagram below, what would X be classified as using KNN ($K=7$)? (same graph as the above)

 Blue circle~~Red circle~~ Yellow circle There would be a tie between blue and Yellow.**Question 4:**

When using KNN for text categorization, do the documents need to be vectorized?

 True~~False~~**Question 5:**

Which one/s of the following is true about a Naive Bayes text classifier?

The probability of multiple attributes appearing together is equal to the sum of the individual probabilities of each of the attributes.

 The order of words or tokens in text is important.

~~the tokens in the texts are independent of each other.~~

- The principle assumption of a NB classifier is the interdependence of each of the attributes.

~~The probability of multiple attributes appearing together is equal to the individual probabilities of each of the attributes multiplied together.~~**Question 6:**

Which one/s of the following text categorization algorithms is NOT based on neural networks?

~~Decision Trees~~~~Support Vector Machines~~~~KNN~~

- RNN

- LTSM

- Perceptron

- CNN

Question 7:

A validation dataset can be used for:

~~Fine tuning the model parameters~~

- Training a model

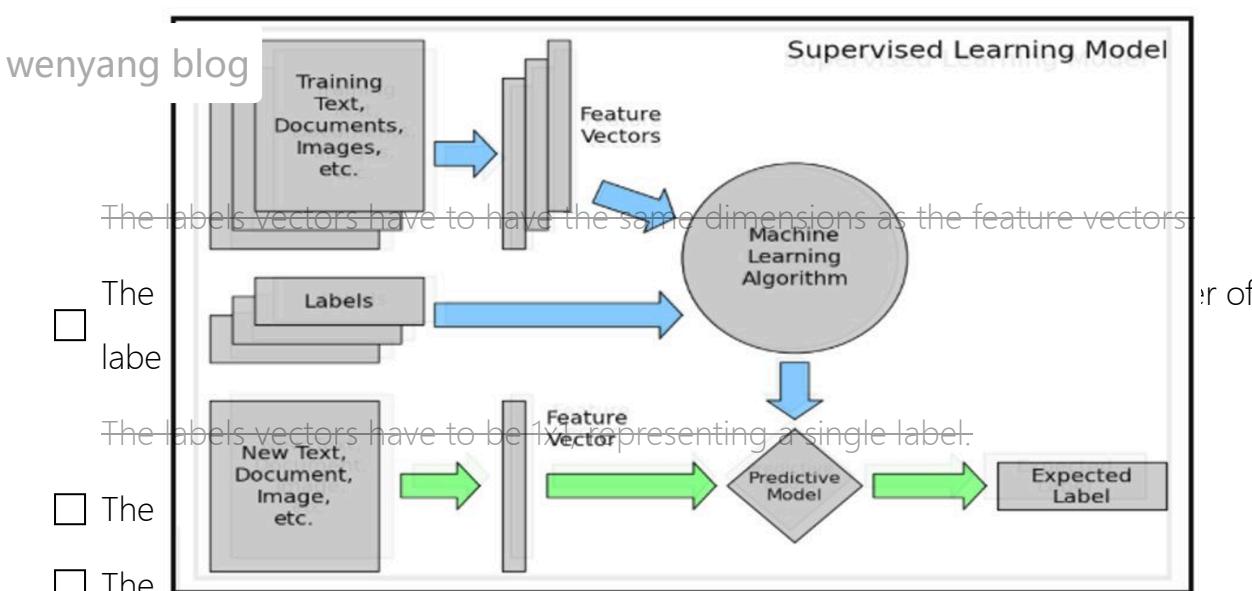
- Can be used for either testing or training a model.

- Testing a model

Question 8:

In the diagram below, which one of the following is NOT true about the labels vector? Assume the Machine Learning Algorithm for this case is some form of Neural Network.





Question 9:

Which one/s of the following is true about deep learning?

- The size of the matrix can either reduce or increase as you go deeper into the layers depending on the kernel function.
- The embedded matrix used in a Deep Learning Neural Network is a 1xn matrix where n is the size of the vocabulary used.

~~It uses a sliding window to select parts of the input matrix to extract selective features.~~

- The output matrix of a Deep Learning Neural Network is always the same size as the input matrix.

~~At the end of the "deep layers," all of the input values are fed into a fully connected layer which is then fed into a softmax function.~~

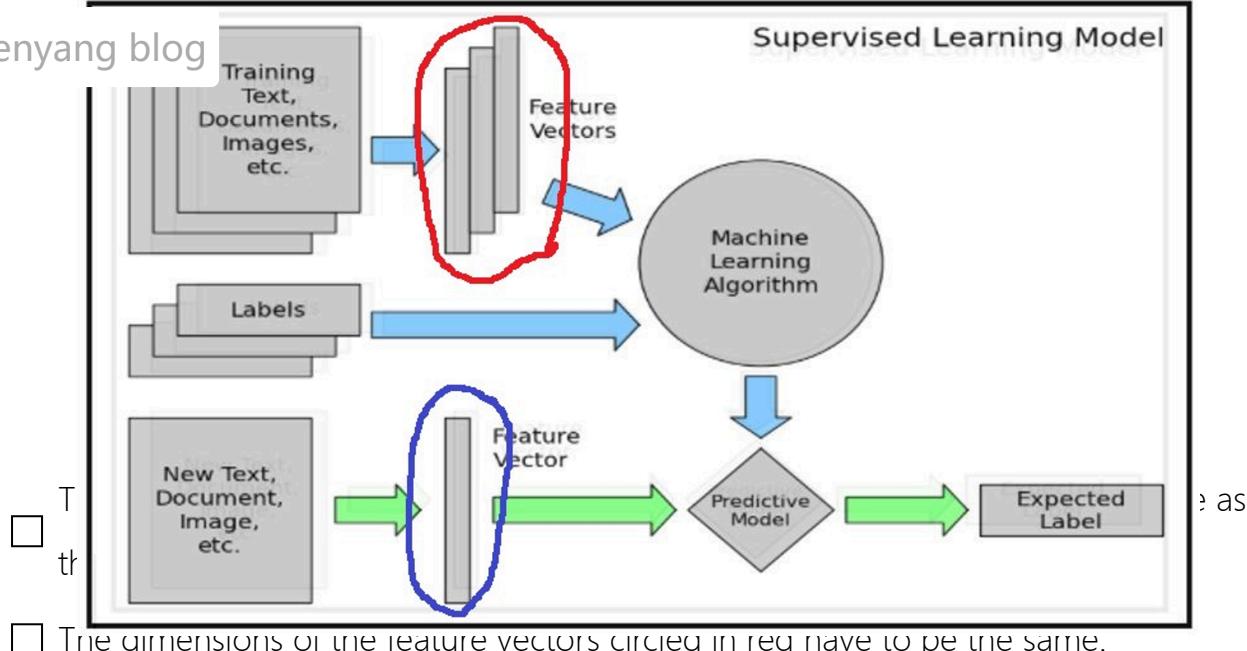
~~As you move deeper into the neural network, the dimensions of the matrix reduce.~~

~~Kernels are used to select parts of the input matrix to construct a specific feature map.~~

- A softmax function is used in all of the layers of a deep learning neural network.

Question 10:

In the diagram below, which of the following is NOT true about the circled feature vectors?



- The dimensions of the feature vectors circled in red have to be the same.

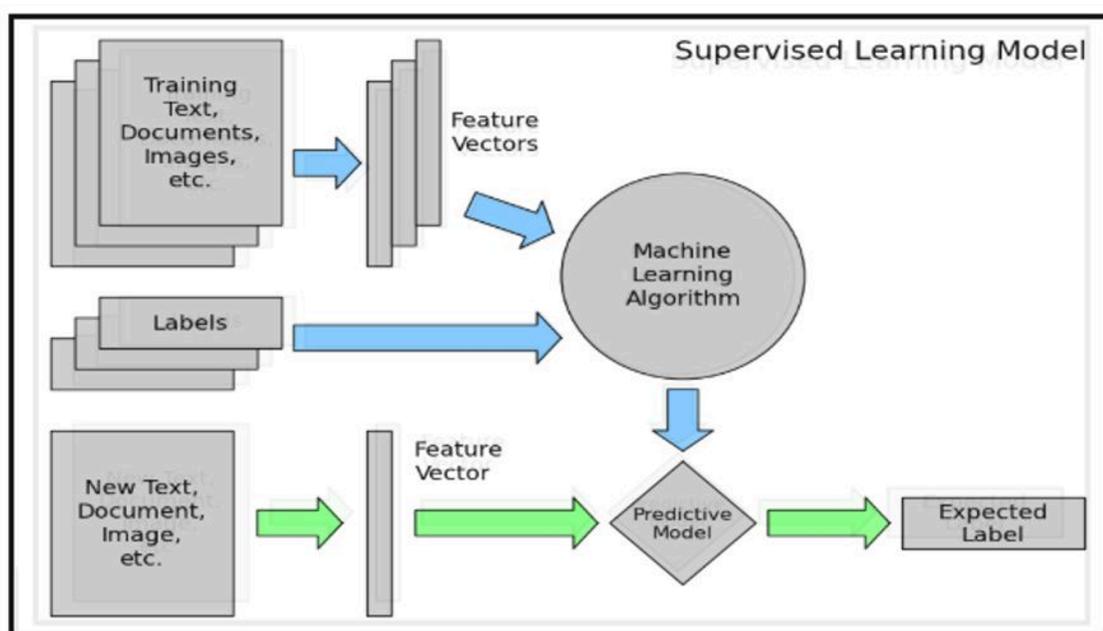
The dimensions of the feature vector circled in blue are always 1x1 representing the label for the document.

The dimensions of the feature vectors circled in red can be different depending on the size of the documents.

The dimensions of the feature vector circled in blue can be different from the dimensions of the feature vectors circled in red.

Question 11:

Consider the diagram below depicting text categorization using a machine learning algorithm. Which of the following cannot be used as a feature in feature vectors?



TFIDF values for the terms in the document.

wenyang blog



of sports related bigrams.

The author of the document.

Frequency of profanities.

The publisher.

TFIDF value of numbers.

Frequency of the POS tag "VBG".

Frequency of numbers.

Presence or absence of the phrase "climate change".

Frequency of trigrams containing the term "war".

Frequency of joining words.

The sentiment.

The date of publication.

The title from the document.

Question 12:

After training a model, its performance is often compared with a baseline model performance. Which ones of the following are true about a baseline model?

It can be any type algorithm with a basic set of parameters without any attempt at fine tuning.

- A baseline model is always a Naive Bayes model.
- Baseline model is trained using only the validation data.
- Baseline is a special type of algorithm, which is used for comparison of other models.

It is the most basic, naive model.

Question 13:

Which ones of the following are true about overfitting in classification problem modeling?

- An overfitted model always overestimates values hence can result in higher error.
- Overfitting is a result of overtraining a model.
- An overfitted model will have a low training and test error and hence is the ideal model.

wenyang blog ~~A well-tuned model will have a high training error and a low testing error.~~



~~An overfitted model will have a low training error and a high testing error.~~

Question 14:

Which of the following statements are true regarding text categorization.

- Euclidean distance cannot be used to categorize documents.
- Similarity measures can only be used to do binary categorization.

~~When using a similarity measure between two documents, they will first need to be categorized.~~

~~A similarity measure can be used to do multi-category classification.~~

~~Any similarity measure can be used to categorize documents.~~

~~Cosine similarity can be used to categorize a document.~~

- Only some similarity measures can be used to categorize documents.
- Similarity measures cannot be used to categorize documents.

Question 15:

Which one/s of the following is true about text categorization?

~~Texts can be classified into 2 categories.~~

~~Multi-category classification can be done using iterative binary classification.~~

- Multi-category classification cannot be done using an algorithm capable of doing only binary classification.
- Texts cannot be classified into multiple categories in a single step.

~~Texts can be classified into multiple categories.~~



Author: wenyang



URL: <https://wenyang.xyz/article/tmquiz>



Copyright: All articles in this blog, except for special statements, adopt BY-NC-SA agreement. Please indicate the source when reprinting.



Quiz7

1 1 / 1 point

Which of the following is **NOT true** about Named Entity Recognition?

- It can be done with only deep learning algorithm.

Any classification algorithm can be used to do it.

It is a typical classification problem.

- It entails recognition of only Names of people and places.
- It is not a classification problem

In essence, it is very similar to other NLP tasks such as POS tagging.

2 1 / 1 point

Use the code snippet below to answer this question. What is the purpose of the code from lines 4 to 26?

```

1 from nltk import ne_chunk, pos_tag, word_tokenize
2 from nltk.tree import Tree
3 import nltk
4 def get_continuous_chunks(text):
5     chunked = ne_chunk(pos_tag(word_tokenize(text)))
6     prev = None
7     continuous_chunk = []
8     current_chunk = []
9     #print(chunked)
10    for i in chunked:
11        if type(i) == Tree:
12            current_chunk.append(" ".join([token for token, pos in i.leaves()]))
13        elif current_chunk:
14            named_entity = " ".join(current_chunk)
15            if named_entity not in continuous_chunk:
16                continuous_chunk.append(named_entity)
17                current_chunk = []
18        else:
19            continue
20
21    if continuous_chunk:
22        named_entity = " ".join(continuous_chunk)
23        if named_entity not in continuous_chunk:
24            continuous_chunk.append(named_entity)
25
26    return continuous_chunk
27
28 txt = "Jacinda Ardern is the Prime Minister of New Zealand but Roenzo isn't."
29 print (get_continuous_chunks(txt))
30
31
32 for sent in nltk.sent_tokenize(txt):
33     for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
34         if hasattr(chunk, 'label'):
35             print(chunk.label(), ' '.join(c[0] for c in chunk))

```

It defines a function to extract multiple tokens representing entities of type PERSON.

It defines a chunker which is later used to identify entities.

- It defines a function to extract multiple tokens representing various types of entities in the text.

It defines a function to classify chunks into various types of entities.

code:

```

from nltk import ne_chunk, pos_tag, word_tokenize
from nltk.tree import Tree
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')

def get_continuous_chunks(text):
    chunked = ne_chunk(pos_tag(word_tokenize(text)))
    prev = None
    continuous_chunk = []
    current_chunk = []

    for i in chunked:
        if type(i) == Tree:
            current_chunk.append(" ".join([token[0] for token in i.leaves()]))
        elif current_chunk:
            named_entity = " ".join(current_chunk)
            if named_entity not in continuous_chunk:
                continuous_chunk.append(named_entity)
            current_chunk = []
        else:
            continue

    if continuous_chunk:
        named_entity = " ".join(continuous_chunk)
        if named_entity not in continuous_chunk:
            continuous_chunk.append(named_entity)

    return continuous_chunk

txt = "Jacinda Ardern is the Prime Minister of New Zealand but Roenzo isn't."

print(get_continuous_chunks(txt))
print("====")
for sent in nltk.sent_tokenize(txt):
    for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
        if hasattr(chunk, 'label'):
            print(chunk.label(), ' '.join(c[0] for c in chunk))

```

3 1 / 1 point

Which one/s of the following could be a reason for an error while running the following code in Google Colab.

```

import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')

with open(DATA_DIR, 'r', encoding='utf-8', errors='ignore') as f:
    sample = f.read()
sentences = nltk.sent_tokenize(sample)
tokenized_sentences = [nltk.word_tokenize(sentence) for sentence in sentences]
tagged_sentences = [nltk.pos_tag(sentence) for sentence in tokenized_sentences]
chunked_sentences = nltk.ne_chunk_sents(tagged_sentences, binary=True)

```

- The file path is not specified.
- Google Drive is not mounted.

The syntax for the "nltk.download" statements are incorrect.

The import statement "import DATA_DIR" is missing.

4 1 / 1 point

Which ones of the following is true about BIO tags used for NER?

It can be used with only non neural network type algorithms.

The second step (NEC) can be used to identify only 3 types of names.

- ✓ It classifies the tokens into 3 categories.
- ✓ The second step (NEC) can be used to identify any number of names.
- ✓ It can be used in a deep learning system.

It can be used to classify tokens into any number of categories.

- ✓ It requires 2 steps to fully do NER.

5 1 / 1 point

Which one/s of the following is true about use of Decision Trees to do NER?

- ✓ It is a Knowledge Engineering approach.

Its a hybrid between Knowledge Engineering and Learning approach.

- ✓ It requires identification of attributes to use of the classification process.

It is a Learning approach.

It will have a high recall but low precision.

6 1 / 1 point

Which of the following is true about Knowledge Engineering based systems and Learning systems?

- ✓ Learning systems are able to generalise better on unseen data.

The disadvantage of Knowledge Engineering bases system is that it required a lot of data.

Knowledge Engineering systems are better suited for Deep Learning algorithms.

- ✓ Knowledge Engineering based systems can be developed with small amount of data.

Learning systems don't need any annotation

- ✓ Learning systems are more suited to Deep Learning algorithms.

7 1 / 1 point

Use the code snippet below to answer this question. What will be printed from the print statement on line 35?

```

1 from nltk import ne_chunk, pos_tag, word_tokenize
2 from nltk.tree import Tree
3 import nltk
4 def get_continuous_chunks(text):
5     chunked = ne_chunk(pos_tag(word_tokenize(text)))
6     prev = None
7     continuous_chunk = []
8     current_chunk = []
9     #print(chunked)
10    for i in chunked:
11        if type(i) == Tree:
12            current_chunk.append(" ".join([token for token, pos in i.leaves()]))
13        elif current_chunk:
14            named_entity = " ".join(current_chunk)
15            if named_entity not in continuous_chunk:
16                continuous_chunk.append(named_entity)
17                current_chunk = []
18        else:
19            continue
20
21    if continuous_chunk:
22        named_entity = " ".join(continuous_chunk)
23        if named_entity not in continuous_chunk:
24            continuous_chunk.append(named_entity)
25
26    return continuous_chunk
27
28 txt = "Jacinda Ardern is the Prime Minister of New Zealand but Roenzo isn't."
29 print (get_continuous_chunks(txt))
30
31
32 for sent in nltk.sent_tokenize(txt):
33     for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
34         if hasattr(chunk, 'label'):
35             print(chunk.label(), ' '.join(c[0] for c in chunk))

```

- ✓ It will print the entity type followed by the content of array c, which could be one or more tokens
- ✓ It will print the entity type followed by a chunk contained in the array c.

~~It will print the chunk label followed by two quote marks followed by the the chunk contained in array c.~~

~~It will print the entity type followed by the first element of the array c.~~

~~It will print the text "label" followed by its entity type.~~

8 1 / 1 point

Using techniques such as regular expressions to identify and label data and then feeding the data into a learning system is an example of :

Ad hoc learning

- ✓ Bootstrapping

Deep learning

Unsupervised learning

9 1 / 1 point

Which one/s of the following is true about the output from the following code.

```

entity_names = []
for tree in chunked_sentences:
    entity_names.extend(extract_entity_names(tree))

print(len(entity_names))
print(len(set(entity_names)))

```

~~Either of the two numbers in the output could be larger.~~

~~The second number in the output will always be larger than the first.~~

- ✓ There would be 2 numbers in the output.
- ✓ The second number in the output will always be smaller than the first.

~~There will be only one number in the output.~~

COMP814

Text Mining

Introduction and Overview

Course Overview and Admin

□ Course Lecturer : Parma Nand

- Room: WZ, level 11.
- Phone: 921 999 ext 5679
- Email: pnand@aut.ac.nz

□ Course Schedule

- 12 x 2 hour classes (room : WG707)
- Lots of student directed research (more than 132 hours)

□ Assessment

- Midway test 1 (40%) – Week 8 (lecture session) (the week back after mid-semester break)
- Assignment 2 (60%) – will be given out in the week 9.
 - Pair based assignment.

■ Non assessed learning milestones

- Quizz every week
 - Self assessment.
 - Answers available immediately after the attempt.
- Lab every week.
 - Learning tool

Format of Lectures

- Lecture followed by lab
- Lectures will introduce new content demonstrated by python examples
- **Labs**
 - Labs will reinforce the content using examples
 - Labs will need to be submitted before the beginning of the next session.
 - Labs will not contribute directly towards course marks but will be used in borderline cases.
 - The submission will need to be a genuine (could be raw) attempt, rather than polished work as you would do for an assignment
- Quizzes
 - Each (Except the test week) week will have a quiz you can do to assess your grasp of the content for that week.
 - You should finish the lab before attempting the quiz.
 - Should finish it before the next session.
 - Answers will be available after finishing the quiz.

Programming environment

- We will use python as the language and Google Colab as the IDE
- You can use Pycharm or another online environment, of your choice, such as Jupyter Notebook. Link given as part of the lab sheet.
 - You can install Jupyter separately, with python, or use the Anaconda package (contains python and commonly data science libraries) for your python install.
- You can download the following if you intend to work in an offline environment.
 - Python environment (3.5 and above)
 - PyCharm as the IDE.
 - OR you can download PyCharm and python as a bundle.
- Despite what you install you will need to download additional packages as and when needed depending on the task at hand.

Text Mining (TM) v Natural Language Processing (NLP)

- Similar but not same
- TM – applied
- NLP – fundamentals
 - Software Engineering v Computer Science
- Other AI disciplines
 - Data Mining
 - Machine Learning

Goals of the course

- To appreciate the value of Text Mining.
- To appreciate the multi-dimensional nature of processing knowledge represented as a natural language.
- To build an understanding of the current state of play in the TM/NLP discipline.
- To build and/or use TM tools.

At the end you would be able to progress research in a chosen area of NLP – either theoretical or implementation

Why TM?

- Needed to take advantage of the vast amount of information encoded in natural languages, online as well as offline.
- Needed even to interface with vast amount of organized information in databases.
- Needed to be able to communicate with machines using natural language (NLG).

Previous/Current State of NLP

- NLP is “hard”
- NLP is still a largely unsolved problem
- NLP is largely multidisciplinary in nature
 - Is partly the reason for NLP being a “hard” problem to solve.
- In spite of this, NLP has progressed immensely in the last few years
 - Since since it became **generative**
 - Why??
- What are LLMs? Pros/cons?

Disciplines of NLP

- **Linguistics** : How words, phrases, and sentences are formed.
- - **Psycholinguistics** : How people understand and communicate using a human language.
- **Computational linguistics** : Deals with models and computational aspects of NLP.
- **Artificial intelligence** : issues related to knowledge representation and reasoning.
- **NL Engineering** : implementation of large, realistic systems, like LLMs, such as ChatGpt and Bart
- **Text Mining** : information extraction from text for a specific purpose (closely related to AI)

Applications of NLP

■ **Text-based applications:**

- - finding documents on certain topics (document categorisation)
- - information retrieval; search for keywords or concepts.
- - (free) information extraction; relevant to a topic.
- - text comprehension
- - translation from a language to another
- - summarization
- - knowledge management

■ **- Dialogue-based applications:**

- - human-machine communication
- - question-answering
- - tutoring systems
- - chatbots
- - problem solving

■ **- Speech – processing**

- **Text to speech**
- **Speech to text**

An Example - Question Answering

■ Text :

First Union Corp is continuing to wrestle with severe problems. According to industry insiders at Pine Webber, their president, John R. Georgius, is planning to retire soon.

■ Question :

Who is the President of First Union Corp?

Basic levels of language processing

- **Morphological Knowledge** - how words are constructed : e.g friend, friendly, unfriendly, friendliness.
- **Syntactic Knowledge** - how words can be put together to form correct sentences, and the role each word play's in the sentence. e.g John ate the cake.
- **Semantic Knowledge** - Words and sentence meaning. e.g
 - They saw a log.
 - They saw a log yesterday.
 - He saws a log.
- **Phonetic** - how words are related to the sounds that realize them
Essential for speech processing.
 - He **leads** the team.
 - **Lead** is a heavy metal.

It gets even more complicated!

- **Pragmatic Knowledge** - how sentences are used in different situations (contexts)
 - Mary grabbed her umbrella.
 - A) It is a cloudy day.
 - B) She was afraid of dogs.
- **Discourse Knowledge** - how the meaning of words and sentences is effected by the previous sentences, e.g. pronoun resolution.
 - John gave his bike to Bill. **He** didn't care much of **it** anyway.
 - John and Tom like apples and **they** eat **them** often.

Some more context examples – Ambiguities without context

- John saw the boy in the park with a telescope
- Fed raises interest rates half a percent in an effort to control inflation

Lexical Ambiguities

- Rice flies like sand.
- Flying planes is/are dangerous.

Semantic Ambiguities

- Words with more than one meaning or sense
- Also called polysemous words
 - John killed the wolf.
 - Bill killed the project.
 - Mary killed Jane.(in tennis or murdered her)

What is a discourse?

- Sentences cannot be processed in isolation
- a **word**, **phrase**, and **utterance** interpretation is shaped by the discourse or dialogue context
- Connected via
 - Coreference
 - Ordering
 - Segmentation

Example by Charles Fillmore:

Please use the toilets, not the pool.

The pool for members only.

Inference in Discourse Processing

- There are several possible ways to interpret an utterance in context
- We need to find the most likely interpretation
- Discourse model provides a computational framework for this search

Some Models of Discourse Structure

- Investigation of **lexical connectivity patterns** as the reflection of discourse structure
- Specification of a small set of **rhetorical relation** among discourse segments
- Adaption of the notion of **grammar**
- Examination **of intentions and relations** among them as the foundation of discourse structure

Discourse Example - dialogue

- 1 A: I'm going camping next week. Do you have a two person tent I could borrow?
- 2 B: Sure. I have a two-person backpacking **tent**.
- 2 -----
- 3 A: The last trip I was on there was a huge storm.
- 4 A: It poured for two hours.
- 5 A: I had a tent, but I got soaked anyway.
- 6 B: What kind of **tent** was it?
- 7 A: A tube tent.
- 8 B: Tube tents don't stand up well in a real storm.
- 9 A: True.
- 9 -----
- 10 B: Where are you going on this trip?
- 11 A: Up in the Minarets.
- 12 B: Do you need any other equipment?
- 13 A: No.
- 14 B: Okay. I'll bring the **tent** tomorrow.

Cohesion

- Assumption: Well-formed text exhibits strong **lexical connectivity** via use of:
 - Repetitions
 - Synonyms
 - Coreference

Discourse Example - dialogue

- 1 **A:** I'm going camping next week. Do you have a two person tent I could borrow?
- 2 **B:** Sure. I have a two-person backpacking tent.

2 -----

- 3 **A:** The last trip I was on there was a huge storm.

- 4 **A:** It poured for two hours.

- 5 **A:** I had a tent, but I got soaked anyway.

- 6 **B:** What kind of tent was it?

- 7 **A:** A tube tent.

- 8 **B:** Tube tents don't stand up well in a real storm.

- 9 **A:** True.

9 -----

10. **B:** Where are you going on this trip?

11. **A:** Up in the Minarets.

12. **B:** Do you need any other equipment?

13. **A:** No.

14. **B:** Okay. I'll bring the tent tomorrow.

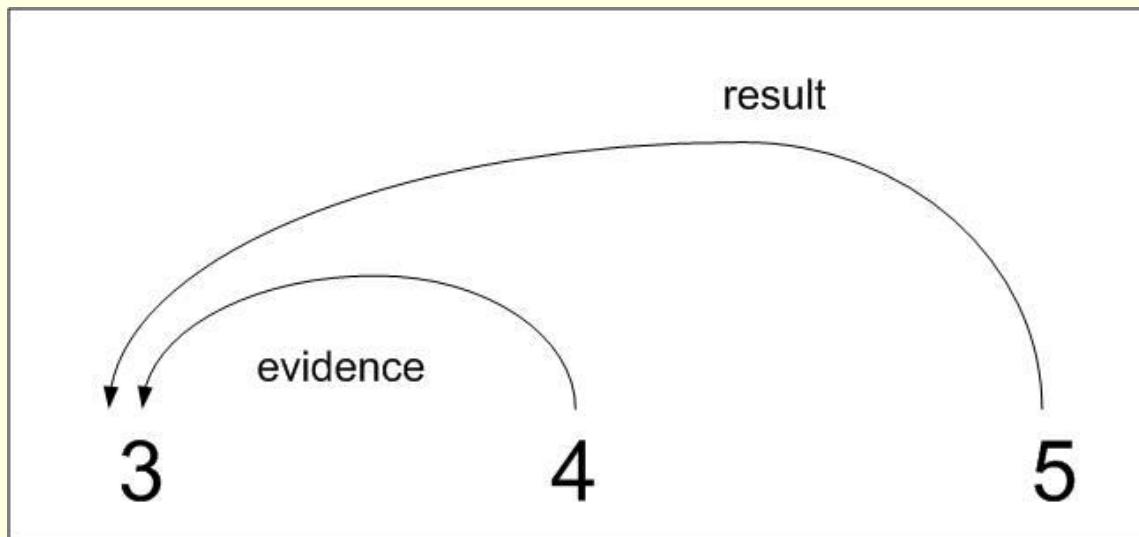
Identify synonyms,
repetitions and co-
references

Rhetorical Structure Theory

- Assumption: Clauses in well-formed text are related via predefined rhetorical relations
- **Evidence:** a claim information intended to increase the readers' belief in the claim

Rhetorical Structure Theory

- 3 The last trip I was on there was a huge storm.
- 4 It poured for two hours.
- 5 I had a tent, but I got soaked anyway.

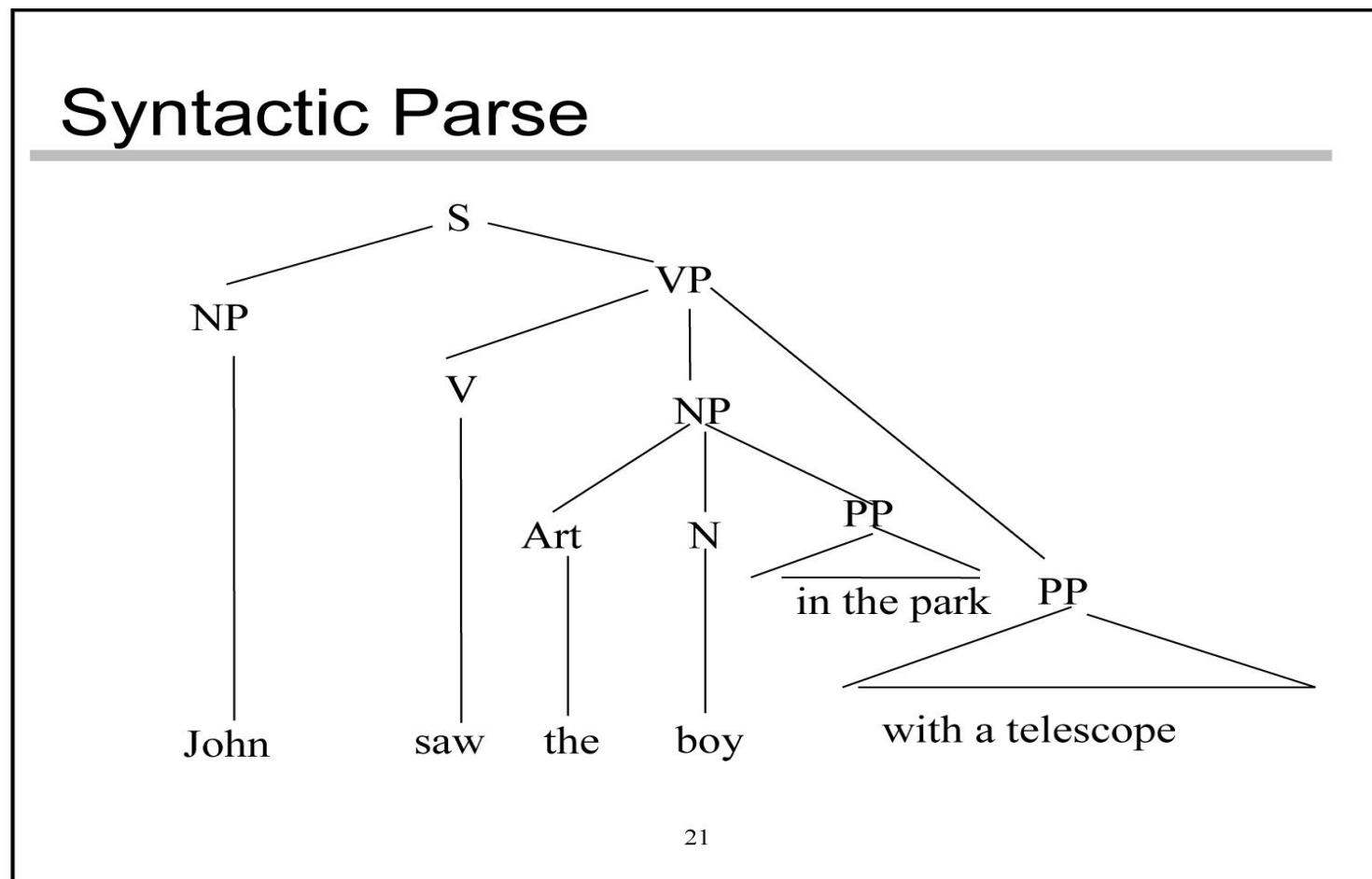


Basic levels of language processing

- All words in a sentence don't have the same status
- Need to identify basic components of sentences
- Need to identify the (syntactic) structure of sentences
- Can be **ambiguous** too!

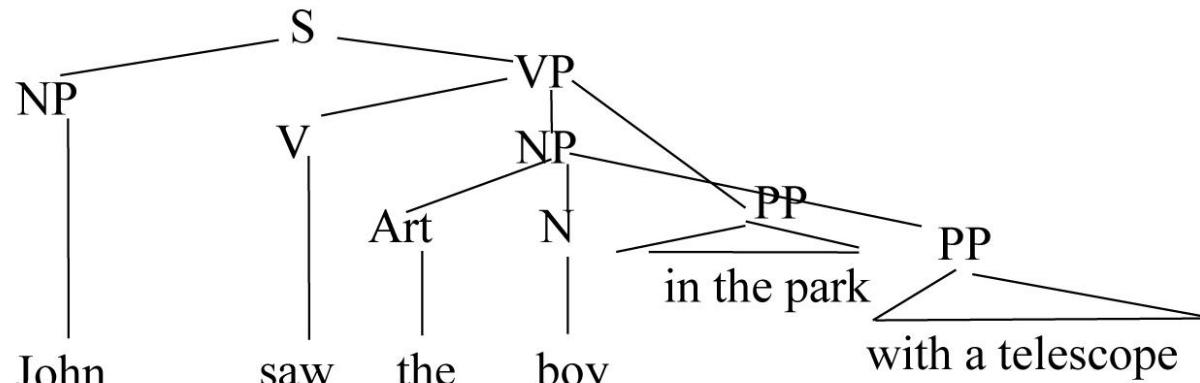
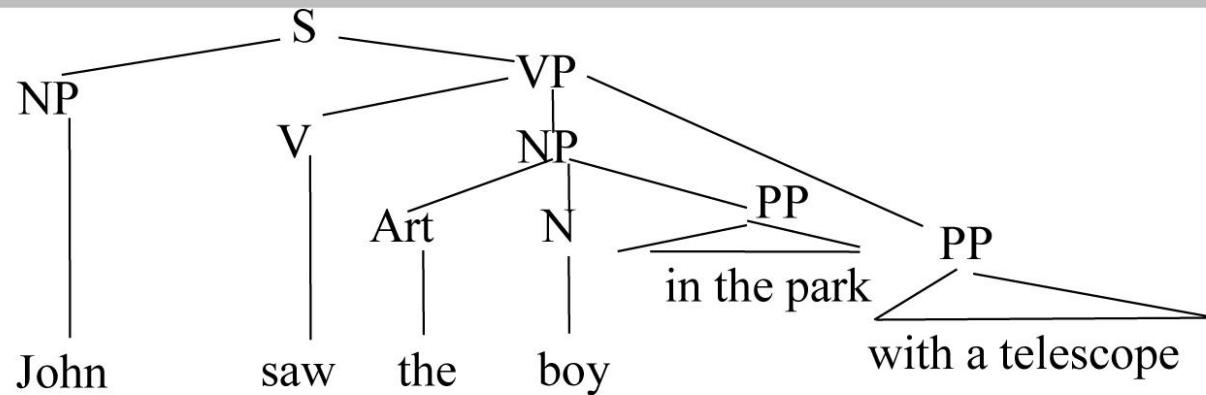
Syntactic structure of a sentence

- John saw the boy in the park with a telescope



Two possible parses

Structural ambiguities



State of the art in NLP Research

- Association of Computational Linguistics (ACL)
 - AAAI -every year /IJCAI -every second year
 - Natural Language Engineering (journal).
 - CICLing – Conference on intelligent Text Processing and Computational Linguistics
-
- Information retrieval/ Extraction
 - MUC - Message Understanding Conf.
 - DUC – Document Understanding Conf.
 - SIGIR – Special Interest Group in IR

Resources

- Machine Readable Dictionaries (MRD)
 - WordNet (www.cogsci.princeton.edu/~wn)
- Large corpora:
 - Penn Treebank –
 - www.cis.upenn.edu/~treebank
 - All Treebank data is released through The Language Data Consortium (LDC)
- RCV1
- TRC2
- Reuters21578
- Ontonotes
- Dbpedia
- Freebase
- WordNet

Penn Treebank Corpora

- Corpus of 4.5 million words of American English
 - Part_of_Speech tagged
 - Syntactic Bracketing

POS tags

PoS Tagging

1.	CC	Coordinating conjunction	13.	NNS	Noun, plural
2.	CD	Cardinal number	14.	NNP	Proper noun, singular
3.	DT	Determiner	15.	NNPS	Proper noun, plural
4.	EX	Existential <i>there</i>	16.	PDT	Predeterminer
5.	FW	Foreign word	17.	POS	Possesive ending
6.	IN	Preposition/subord. conjunction	18.	PRP	Personal pronoun
7.	JJ	Adjective	19.	PP\$	Possesive pronoun
8.	JJR	Adjective, comparative	20.	RB	Adverb
9.	JJS	Adjective, superlative	21.	RBR	Adverb, comparative
10.	LS	List item marker	22.	RBS	Adverb, superlative
11.	MD	Modal	23.	RP	Particle
12.	NN	Noun, singular or mass	24.	SYM	Symbol (mathematical or scientific)

POS tags

PoS Tagging

25.	TO	<i>to</i>	37.	#	Pound sign
26.	UH	Interjection	38.	\$	Dollar sign
27.	VB	Verb, base form	39.	.	Sentence-final punctuation
28.	VBD	Verb, past tense	40.	,	Comma
29.	VBG	Verb,gerund/present participle	41.	:	Colon, semi-colon
30.	VBN	Verb, past participle	42.	(Left bracket character
31.	VBP	Verb, non-3 rd ps.sing. present	43.)	Right bracket character
32.	VBZ	Verb, 3 rd ps.sing. present	44.	"	Straight double quote
33.	WDT	<i>wh-determiner</i>	45.	'	Left open single quote
34.	WP	<i>wh-pronoun</i>	46.	“	Left open double quote
35.	WP\$	Possesive <i>wh-pronoun</i>	47.	’	Right close single quote
36.	WRB	<i>wh-adverb</i>	48.	”	Left close double quote

Syntactic tags

The Syntactic Tag Set

	Tags	
1.	ADJP	Adjective phrase
2.	ADVP	Adverb phrase
3.	NP	Noun phrase
4.	PP	Prepositional phrase
5.	S	Simple declarative clause
6.	SBAR	Clause introduced by subordinating conjunction or 0 (see below)
7.	SBARQ	Direct question introduced by <i>wh</i> -word or <i>wh</i> -phrase
8.	SINV	Declarative sentence with subject-aux inversion
9.	SQ	Subconstituent of SBARQ excluding <i>wh</i> -word or <i>wh</i> -phrase
10.	VP	Verb phrase
11.	WHADVP	<i>wh</i> -adverb phrase
12.	WHNP	<i>wh</i> -noun phrase
13.	WHPP	<i>wh</i> -prepositional phrase
14.	X	Constituent of unknown or uncertain category
Null elements		
1.	*	“Understood” subject of infinitive or imperative
2.	0	Zero variant of <i>that</i> in subordinate clauses
3.	T	Trace-marks position where moved <i>wh</i> -constituent is interpreted
4.	NIL	Marks position where preposition is interpreted in pied-piping contexts ³¹

Sample bracketed text

Sample Bracketed Text

```
((S  
  (NP Battle-tested industrial managers  
    here)  
  always  
  (VP buck  
    up  
    (NP nervous newcomers  
    (PP with  
      (NP the tale  
      (PP of  
        (NP (NP the  
          (ADJP first  
          (PP of  
            (NP their countrymen)))  
          (S (NP *)  
            to  
            (VP visit  
              (NP Mexico)))  
  
          (NP (NP a boatload  
            (PP of  
              (NP (NP warriors)  
                (VP-1 blown  
                  ashore  
                  (ADV (NP 375 years)  
                    ago))))  
          (VP-1 *pseudo-attach*)))))))))  
.)
```

COMP814 – Text Mining

Pre-processing I Tokens and Ngrams

Basic units of language

- Alphabet (symbol)
- Word (Token)
- Phrase (Utterance)
- Document (Discourse)
- Collection of Discourses (Corpus)
- Collection of Corpus (Corpora)

Some basics on Words

- Words are “approximately” equivalent to tokens.
- When might it be different?
- When might we need to do the opposite?
 - Chunking.

Nouns and Verbs in English

- Nouns are simple
 - Markers for plural and possessive
- Verbs are only slightly more complex
 - Markers appropriate to the tense of the verb

Regulars and Irregulars

- It is a little complicated by the fact that some words misbehave (refuse to follow the rules)
 - Mouse/mice, goose/geese, ox/oxen
 - Go/went, fly/flew
- The terms regular and irregular are used to refer to words that follow the rules and those that don't

Regulars and Irregulars

- It is a little complicated by the fact that some words misbehave (refuse to follow the rules)
 - Mouse/mice, goose/geese, ox/oxen
 - Go/went, fly/flew
- The terms regular and irregular are used to refer to words that follow the rules and those that don't

Regular and Irregular Verbs

■ Regulars...

- Walk, walks, walking, walked, walked

■ Irregulars

- Eat, eats, eating, **ate**, **eaten**
- Catch, catches, catching, **caught**, **caught**
- Cut, cuts, cutting, **cut**, **cut**

English Morphology

- Morphology is the study of the ways that words are built up from smaller meaningful units called morphemes
- We can usefully divide morphemes into two classes
 - **Stems:** The core meaning-bearing units
 - **Affixes:** Bits and pieces that adhere to stems to change their meanings and grammatical functions

English Morphology

- We can further divide morphology up into two broad classes
 - Inflectional
 - Derivational

Inflectional Morphology

- Inflectional morphology concerns the combination of stems and affixes where the resulting word:
 - Has the same word class as the original
 - Serves a grammatical/semantic purpose that is
 - Different from the original
 - But is nevertheless transparently related to the original
 - E.g. skip, skipping, skipped
 - Are all different forms of verbs

Inflectional Morphology

- Inflectional morphology in English is fairly straightforward
- But is complicated by the fact that there are irregularities

Derivational Morphology

- Derivational morphology is the messy stuff that no one ever taught you.
 - Quasi-systematicity
 - Irregular meaning change
 - Changes of word class

Derivational Morphology

- Again is a combination of stems and affixes, however this time the resulting word changes the grammatical function of the word or the semantic meaning.
 - E.g. construct is a verb and construction is a noun
- Note: some affixes changes the meaning of the word without changing the grammatical function. These are still derivational.
 - E.g. deconstruction, unfriendly

Derivational Examples

■ Verbs and Adjectives to Nouns

-ation	computerize	computerization
-ee	appoint	appointee
-er	kill	killer
-ness	fuzzy	fuzziness

Derivational Examples

■ Nouns and Verbs to Adjectives

-al	computation	computational
-able	embrace	embraceable
-less	clue	clueless

A simple language model

Word Prediction

- Guess the next word...
 - ... *I notice three guys standing on the ???*
- There are many sources of knowledge that can be used to inform this task, including arbitrary world knowledge.
- But it turns out that you can do pretty well by simply looking at the **preceding words** and keeping track of some fairly **simple counts**.

Word Prediction

- We can formalize this task using what are called ***N*-gram** models.
- *N*-grams are token sequences of length *N*.
- Our earlier example contains the following 2-grams (aka bigrams)
 - (I notice), (notice three), (three guys), (guys standing), (standing on), (on the)
- Given knowledge of counts of *N*-grams such as these, we can guess likely next words in a sequence.

N-Gram Models

- More formally, we can use knowledge of the **counts of *N*-grams to assess the conditional probability of candidate words as the next word in a sequence.**
- Or, we can use them to assess the probability of an entire sequence of words.
 - Pretty much the same thing as we'll see...

Applications

- It turns out that being able to predict the next word (or any linguistic unit) in a sequence is an extremely useful thing to be able to do.
- It lies at the core of the following applications
 - Automatic speech recognition
 - Handwriting and character recognition
 - Spelling correction
 - Machine translation
 - And many more.

Counting

- Simple counting lies at the core of any probabilistic approach. So let's first take a look at what we're counting.
 - *He stepped out into the hall, was delighted to encounter a water brother.*
 - 13 tokens, 15 if we include “,” and “.” as separate tokens.
 - Assuming we include the comma and period, how many bigrams are there?

Counting

- Not always that simple
 - *I do uh main- mainly business data processing*
- Spoken language poses various challenges.
 - Should we count “uh” and other fillers as tokens?
 - What about the repetition of “mainly”? Should such do-overs count twice or just once?
 - The answers depend on the application.
 - If we’re focusing on something like ASR to support indexing for search, then “uh” isn’t helpful (it’s not likely to occur as a query).
 - But filled pauses are very useful in dialog management, so we might want them there.

Counting: Types and Tokens

- How about
 - *They picnicked by the pool, then lay back on the grass and looked at the stars.*
 - 18 tokens (again counting punctuation)
- But we might also note that “*the*” is used 3 times, so there are only 16 unique **types** (as opposed to tokens).
- In going forward, we’ll have occasion to focus on counting both **types** and **tokens** of both words and N -grams.

Counting: Wordforms

- Should “cats” and “cat” count as the same when we’re counting?
- How about “geese” and “goose”?
- Some terminology:
 - Lemma: a set of lexical forms having the same stem word, major part of speech, and rough word sense
 - **Lemma** – valid dictionary word
 - **Stem** – may not be a valid dictionary word
 - Wordform: fully inflected surface form also called inflections
- Again, we’ll have occasion to count both lemmas and wordforms

Counting: Corpora

- So what happens when we look at large bodies of text instead of single utterances?
- Brown et al (1992) large corpus of English text
 - 583 million wordform tokens
 - 293,181 wordform types
- Google
 - Crawl of 1,024,908,267,229 English tokens
 - 13,588,391 wordform types
 - That seems like a lot of types... After all, even large dictionaries of English have only around 500k types. Why so many here?
 - Numbers
 - Misspellings
 - Names
 - Acronyms
 - Etc.

Language Modeling

- Back to word prediction
- We can model the word prediction task as the ability to assess the **conditional probability** of a word given the previous words in the sequence
 - $P(w_n | w_1, w_2 \dots w_{n-1})$
- We'll call a statistical model that can assess this a *Language Model*

Language Modeling

- How might we go about calculating such a conditional probability?

- One way is to use the definition of conditional probabilities and look for counts.

- So to get $P(\text{the} \mid \text{its water is so transparent that})$

- By definition that's :

$P(\text{its water is so transparent that the})$

$P(\text{its water is so transparent that})$

We can get each of those from counts in a large corpus.

How would you get this?

Summary

- We can compute the probability of a token based on the occurrence of other tokens in a (large) corpus.
- Can extend this to ngrams
- Can then be used to do useful tasks such as spelling/grammar correction and other nlp tasks such as POS tagging.
- Statistical technique.

Code Demo

COMP814 – Text Mining

Pre-processing II
POS Tagging
Evaluation
Parsing

Word Classes

- Words that somehow ‘behave’ alike:
 - Appear in similar contexts
 - Perform similar functions in sentences
 - Undergo similar transformations
- ~9 traditional word classes of **parts of speech**
 - Noun, verb, adjective, preposition, adverb, article, interjection, pronoun, conjunction

Reference

- A lot (but not all) of the material in the following slides are from chapter 5 of the book:

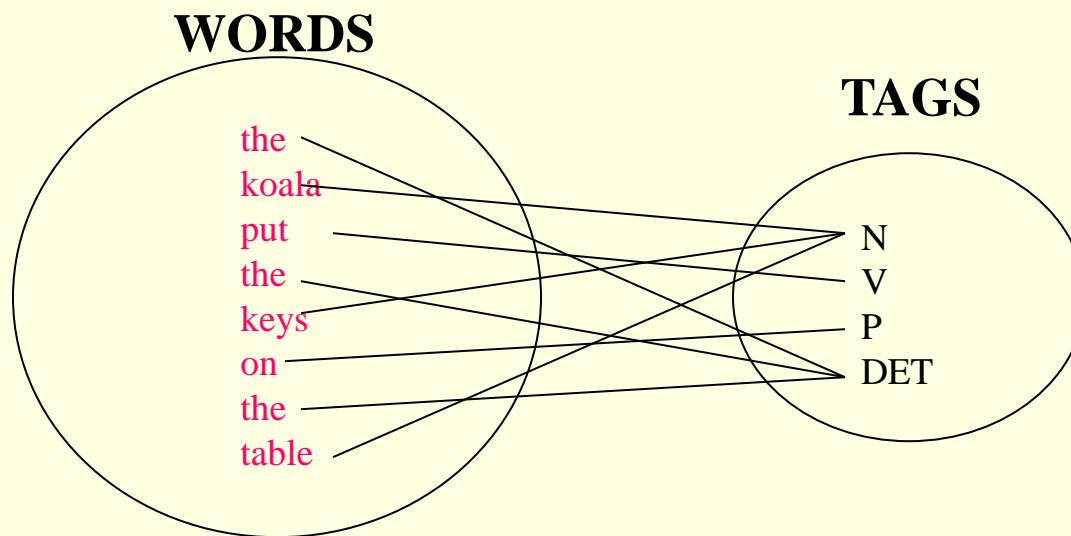
Jurafsky and Martin (2000), Speech and Language Processing.

Some Examples

■ N	noun	chair, bandwidth, pacing
■ V	verb	study, debate, munch
■ ADJ	adjective	purple, tall, ridiculous
■ ADV	adverb	unfortunately, slowly
■ P	preposition	of, by, to, for, at
■ PRO	pronoun	I, me, mine, he his, her
■ DET	determiner	the, a, an, that, those

Defining POS Tagging

- The process of assigning a part-of-speech or lexical class marker to each word in a corpus:



Applications for POS Tagging

- Speech synthesis pronunciation
 - *Lead – verb versus noun*
 - Parsing: e.g. *Time flies like an arrow*
 - Is *flies* an N or V?
- Word prediction in speech recognition
 - Possessive pronouns (*my, your, her*) are likely to be followed by nouns
 - Personal pronouns (*I, you, he*) are likely to be followed by verbs
- Machine Translation
- Information extraction
 - Extract all clauses
 - Dominant concepts
 - Event modelling
 - etc

Closed vs. Open Class Words

■ Closed class: relatively fixed set

- Prepositions: **of, in, by, ...**
- Auxiliaries: **may, can, will, had, been, ...**
- Pronouns: **I, you, she, mine, his, them, ...**
- Usually **function words** (short common words which play a role in grammar)

■ Open class: productive

- English has 4: Nouns, Verbs, Adjectives, Adverbs
- Many languages have all 4, but not all!
- In Lakhota and possibly Chinese, what English treats as adjectives are more like verbs.

Open Class Words

■ Nouns

■ Proper nouns

- AUT, New Zealand, David Seymour, Metropolitan Transit Center
- English capitalizes these
- Many have abbreviations eg. Ack and CHCH

■ Common nouns

- All the rest
- German capitalizes these.

- Count nouns vs. mass nouns
 - Count: Have plurals, countable: **goat/goats, one goat, two goats**
 - Mass: **Not** countable (**fish, salt, communism**) (**?two fishes**)
- **Adjectives:** identify properties or qualities of nouns
 - Color, size, age, ...
 - Adjective ordering restrictions in English:
 - **Old blue book, not Blue old book**
 - In Korean, adjectives are realized as verbs
- **Adverbs:** also modify things (verbs, adjectives, adverbs)
 - **The very happy man walked home extremely slowly yesterday.**

- Directional/locative adverbs (*here, home, downhill*)
- Degree adverbs (*extremely, very, somewhat*)
- Manner adverbs (*slowly, slinkily, delicately*)
- Temporal adverbs (*Monday, tomorrow*)
- Verbs:
 - In English, take morphological affixes (*eat/eats/eaten*)
 - Represent actions (*walk, ate*), processes (*provide, see*), and states (*be, seem*)
 - Many subclasses, e.g.
 - *eats/V* ⇒ *eat/VB, eat/VBP, eats/VBZ, ate/VBD, eaten/VBN, eating/VBG, ...*
 - Reflect morphological form & syntactic function

How Do We Assign Words to Open or Closed?

- **Nouns** denote people, places and things and can be preceded by articles? But...
 - My *typing* is very bad.
 - *The *Mary* loves John.
- **Verbs** are used to refer to actions, processes, states
 - But some are **closed class** and some are **open**
I *will have emailed* everyone by noon.
- **Adverbs** modify actions
 - Is *Monday* a temporal adverbial or a noun?

Closed Class Words

- Idiosyncratic

- Closed class words (**Prep, Det, Pron, Conj, Aux, Part, Num**) are generally easy to process, since we can enumerate them....but

- Is “up” a Particle or a Preposition?

- George eats up his dinner/George eats his dinner up.



??

- Articles come in 2 flavors: **definite (the)** and **indefinite (a, an)**

- What is this in ‘this guy...’?

Choosing a POS Tagset

- To do POS tagging, first need to choose a set of tags
- Could pick very coarse (small) tagsets
 - Eg., N, V, Adj, Adv.
- More commonly used: Brown Corpus (Francis & Kucera '82), 1M words, 87 tags – more informative but more difficult to tag
 - Most commonly used: Penn Treebank: hand-annotated corpus of *Wall Street Journal*, 1M words, 45-46 subset

Penn Treebank Tagset

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	+,%,&
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	"	Left quote	(" or ")
POS	Possessive ending	's	"	Right quote	(' or ')
PRP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, (, { , <)
PRP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(] ,) , } , >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... - -)
RP	Particle	<i>up, off</i>			

Example of tagged sentence

- The/DT grand/JJ jury/NN commmented/VBD
on/IN a/DT number/NN of/IN other/JJ
topics/NNS ./.

Tag Ambiguity

- Words often have more than one POS: *back*
 - The *back door* = JJ
 - On my *back* = NN
 - Win the voters *back* = RB
 - Promised to *back* the bill = VB
- The POS tagging problem is ***to determine the POS tag for a particular instance of a word***

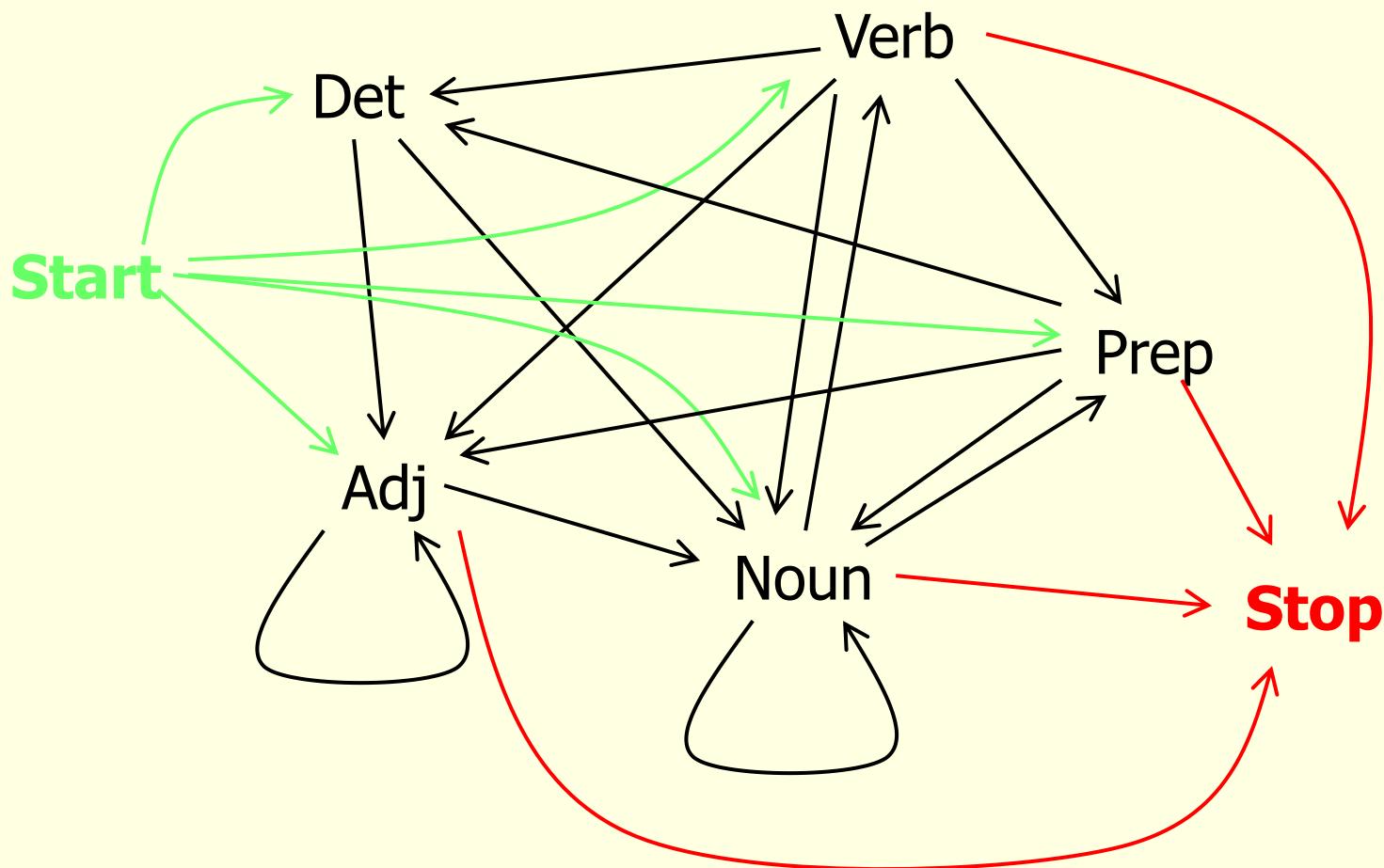
Tagging Whole Sentences with POS is Hard

- Ambiguous POS contexts
 - E.g., Time flies like an arrow.
- Possible POS assignments
 - Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
 - Time/N flies/V like/Prep an/Det arrow/N
 - Time/V flies/N like/Prep an/Det arrow/N
 - Time/N flies/N like/V an/Det arrow/N
 -

Probabilistic Models of POS tagging

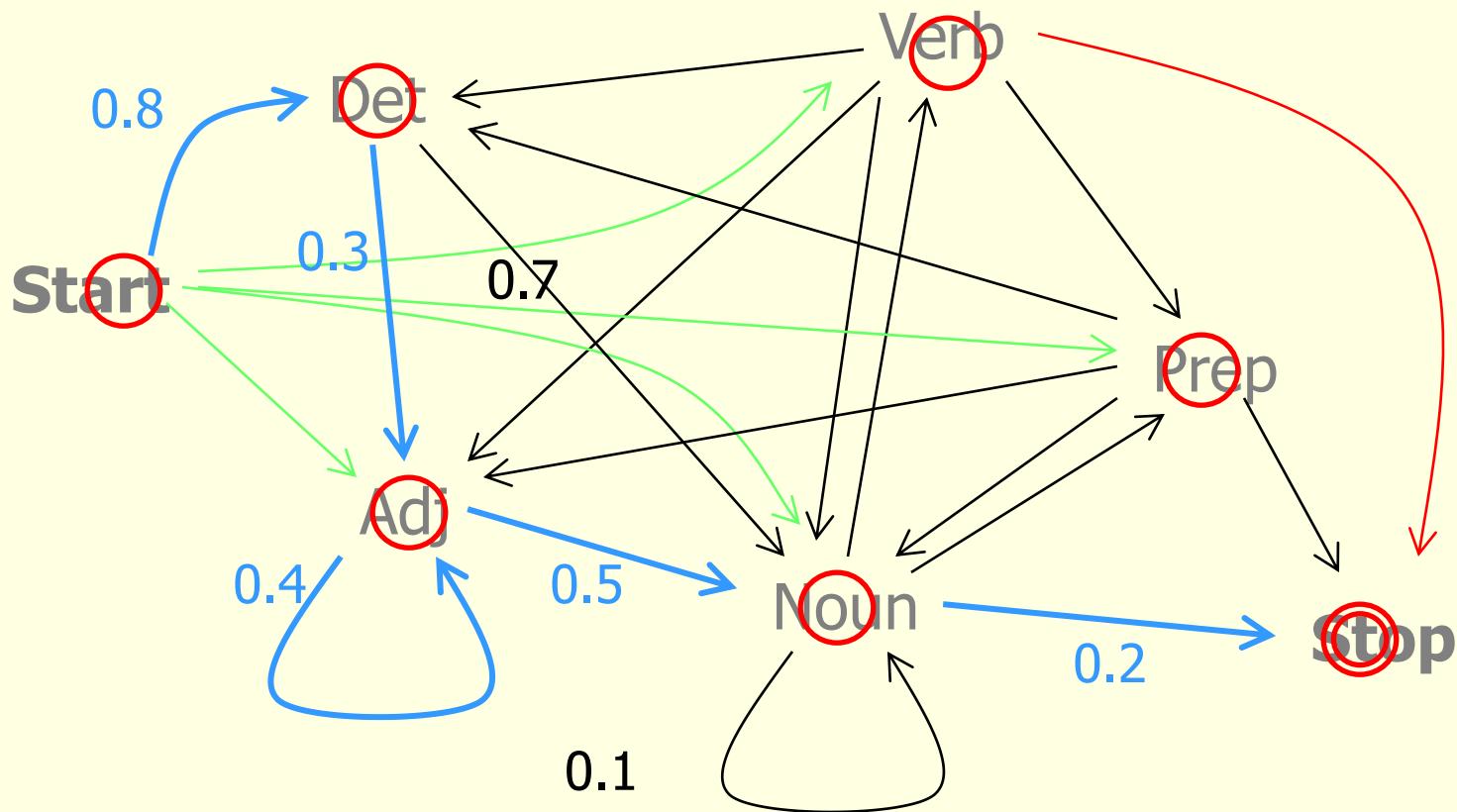
- For tokens w_1, \dots, w_n , find the most probable corresponding sequence of possible tags t_1, \dots, t_n
- We assume that *probable* means something like “most frequently observed in some manually tagged corpus of words”.
- Penn Treebank II (a common training corpus)
 - 1 million words from the Wall Street Journal
 - Tagged for POS (and other attributes)

Markov Model (bigrams)



Markov Model as a FSM

$p(\text{tag seq})$



$$\text{Start} \text{ Det} \text{ Adj} \text{ Adj} \text{ Noun} \text{ Stop} = 0.8 * 0.3 * 0.4 * 0.5 * 0.2$$

Methodology: Error Analysis

- Confusion matrix:
 - E.g. which tags did we most often confuse with which other tags?
 - How much of the overall error does each confusion account for?

		Machine Classification		
		VB	TO	NN
Ground Truth	VB			
	TO			
	NN			

Evaluation Matrices 1

- **True Positives:** Machine identified positives which are also similarly identified positives by human.
- **False Positives:** Machine identified positives which have been identified as negatives as by human.
- **False Negatives:** Machine identified negatives which have been identified as positives by human.
- **True Negatives:** Machine identified negatives which have been identified as negatives by human.

Evaluation Matrices 2

$$Recall(R) = \frac{TP}{TP+FN}$$

$$Precision(P) = \frac{TP}{TP+FP}$$

$$Error\ Rate(ER) = \frac{FP+FN}{TP+FP+FN+TN}$$

$$Accuracy(A) = \frac{TP+TN}{TP+FP+FN+TN}$$

$$F_{value} = \frac{(1+\beta^2).P.R}{(\beta^2).P+R}$$

Exercise : Two class evaluation

$$Recall(R) = \frac{TP}{TP+FN}$$

$$Precision(P) = \frac{TP}{TP+FP}$$

$$Error\ Rate(ER) = \frac{FP+FN}{TP+FP+FN+TN}$$

$$Accuracy(A) = \frac{TP+TN}{TP+FP+FN+TN}$$

$$F_{value} = \frac{(1+\beta^2).P.R}{(\beta^2).P+R}$$

Computed YES	Computed NO
TP	FN
FP	TN

Computed YES	Computed NO
500	25
100	5000

Multi Category Evaluation

		Machine Computation			
		VB	TO	NN	
		VB	420	20	300
Ground Truth		TO	10	115	35
		NN	10	10	380

Evaluating Tagging Approaches

- For any NLP problem, we need to know how to evaluate our solutions
- Possible **Gold Standards** -- ceiling:
 - Annotated naturally occurring corpus
 - Human task performance (96-7%)
 - How well do humans agree?
 - **Kappa statistic**: avg pairwise agreement corrected for chance agreement
 - Can be hard to obtain for some tasks: sometimes humans don't agree
- **Baseline**: how well does simple method do?
 - For tagging, most common tag for each word (91%)
 - How much improvement do we get over baseline?

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	+,%,&
CD	Cardinal number	<i>one, two, three</i>	TO	“to”	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential ‘there’	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VBN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	\$
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	#
PDT	Predeterminer	<i>all, both</i>	“	Left quote	(‘ or “)
POS	Possessive ending	<i>'s</i>	”	Right quote	(‘ or ”)
PRP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	([, (, {, <)
PRP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	(],), }, >)
RB	Adverb	<i>quickly, never</i>	,	Comma	,
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	(: ; ... - -)
RP	Particle	<i>up, off</i>			

Parsing

- Identifying chunks or units in a clause.
- The units have a function in the clause and these functions are attached together.
- This is a step after POS tagging.
- CFG parsing is recursive, hence difficult.
- In practice use either Treebank grammar or even simpler Dependency parse.

Penn Treebank

- Penn TreeBank is a widely used treebank.

- Most well known is the Wall Street Journal section of the Penn TreeBank.

- 1 M words from the 1987-1989 Wall Street Journal.

```
( (S ('' '))
  (S-TPC-2
    (NP-SBJ-1 (PRP We) )
    (VP (MD would)
      (VP (VB have)
        (S
          (NP-SBJ (-NONE- *-1) )
          (VP (TO to)
            (VP (VB wait)
              (SBAR-TMP (IN until)
                (S
                  (NP-SBJ (PRP we) )
                  (VP (VBP have)
                    (VP (VBN collected)
                      (PP-CLR (IN on)
                        (NP (DT those)(NNS assets))))))))))))
        (, ,) ('' ')
        (NP-SBJ (PRP he) )
        (VP (VBD said)
          (S (-NONE- *T*-2) )))
        (. .) ))
```

Treebank Grammars

- Treebanks implicitly define a grammar for the language covered in the treebank.
- Simply take the local rules that make up the sub-trees in all the trees in the collection and you have a grammar.
- Not complete, but if you have decent size corpus, you'll have a grammar with decent coverage.

Treebank Grammars

- Such grammars tend to be very flat due to the fact that they tend to avoid recursion.
 - To ease the annotators burden
- For example, the Penn Treebank has 4500 different rules for VPs. Among them...

VP → VBD PP

VP → VBD PP PP

VP → VBD PP PP PP

VP → VBD PP PP PP PP

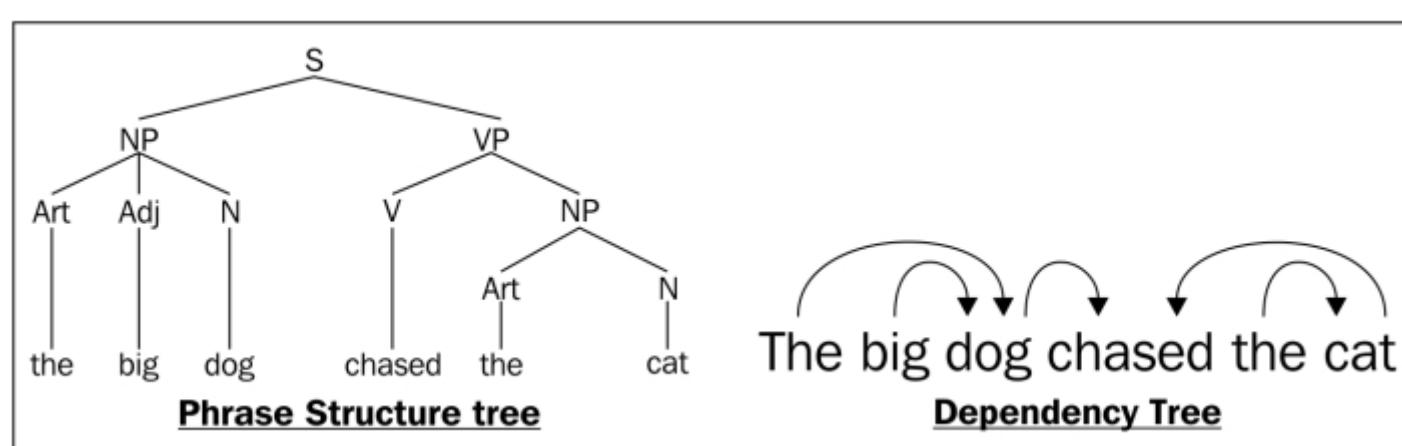
Dependency Relations

Argument Dependencies	Description
nsubj	nominal subject
csubj	clausal subject
dobj	direct object
iobj	indirect object
pobj	object of preposition

Modifier Dependencies	Description
tmod	temporal modifier
appos	appositional modifier
det	determiner
prep	prepositional modifier

Dependency Parsing

- Each linguistic word is connected via a directed link.
- The parse tree captures the (unidirectional) relationship between words and phrases.





Sample Code for last lab

— Demo code for this session

COMP814 – Text Mining

Vector Space Model/ Similarity Computations

Reference

- Mainly chapter 6 & 7 – Introduction to Information Retrieval – Manning et. al
- Etc

What is Vector Space Model and Why?

- Representation of a document or a discourse with a data structure.
- So as to be able to do operations on it.
- Vector of words
- Can do vector and other mathematical operations
- So that documents can be ranked

How can we retrieve relevant documents from an archive?

- Boolean model: all documents matching the query are retrieved
- A query consists of search words combined using logical AND and OR operators.
- The matching is binary: yes or no
- Extreme cases: the list of retrieved documents can be empty, or huge
- A ranking of the documents matching a query is required so that more relevant documents are presented first as in a google search.
 - **What are the possibilities for ranking?**

Document scoring possibilities

- Need a way of assigning a score or weight to a query and document pair
- Most obvious is the frequency of the terms.
- Ignore the order of occurrence
 - **Bag of Words** model
- Are all words equally important?
 - Stop words?
 - Word inflections
 - Plurals and singular words?

Vector representation of a corpus

	d_1	d_2	d_3	d_n
t_1	w_{11}	w_{12}	w_{13}	w_{1n}
t_2	w_{21}	w_{22}	w_{23}	w_{1n}
.
.
.
.
t_m	w_{m1}	w_{m2}	w_{m3}	w_{mn}

Vector Space Representation

- How many dimensions will the vector have?
- Dimension reduction strategies
 - Stop words
 - Stemming
 - Feature selection
 - Latent Semantic Analysis

Frequency count matrices

- Say we are analysing 8 newspaper articles for relevance to National Party

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8
John	15	12	0	0	2	0	0	2
Key	12	0	4	0	0	0	0	8
Government	0	0	0	12	3	0	4	6
National	0	0	5	3	0	6	0	0
Party	4	3	0	0	0	8	7	7
English	0	4	0	1	5	0	0	2

- One Hot representation
 - Vector representing the presence or absence of tokens

Bag of words model (BOW)

- Vector representation doesn't consider the ordering of words in a document
- *The dog bit the man and The man bit the dog would have same representation*
- This is called the bag of words model.
- We will see later that there are models that recover the positional information
- However the BOW model is surprisingly effective in most situations.

Vector Space model: word2vec

- Each term t of the dictionary is considered as a dimension
- A document d can be represented by the **weight** of each dictionary term:
 - $V(d) = (\mathbf{w}(t_1, d), \mathbf{w}(t_2, d), \dots, \mathbf{w}(t_n, d))$
 - Sparse matrix for most documents
- Question: does this representation allow to compute the similarity between documents ?
- Similarity between vectors ?
 - inner product $V(d_1).V(d_2)$
- What about the length of a vector ?
- Longer documents will be represented with longer vectors (and higher frequencies), but that does not mean **they are more important?**

Term Frequency Tf

- The term frequency $tf_{t,d}$ of term t in document d is defined as the number of times that t occurs in d .
- We want to use tf to compute relevance to a query
- Raw term frequency is not what we want:
 - A document with 10 occurrences of the term is **more** relevant than a document with 1 occurrence of the term.
 - But **not 10 times more relevant**.
- Relevance does **not increase proportionally with term frequency**.
- Solution: we can **normalize** it.
 - We can divide the frequency of the term by the number of words

Document frequency

- Frequent terms are less informative than rare terms
- Consider a query term that is frequent in the collection (e.g., *high, increase, line*)
- A document containing such a term is more likely to be relevant than a document that doesn't
- But it's not a sure indicator of relevance.
 - For frequent terms, we want high positive weights
 - E.g. for words like *high, increase, and line*
- But **lower weights than for rare terms.**
- We will use document frequency (df) to capture this.

Inverse Document Frequency idf

- df_t is the document frequency of t , ie. the number of documents that contain t
- df_t is an *inverse measure* of the **information content of the term t**
- $df_t \leq N$, N is *the number of documents*.
- We define the id_f (inverse document frequency) of t as

$$Idf_t = \log_{10} (N/df_t)$$

The base is not important, why?

- Where N is the total number of documents and df_t is the document frequency containing t
- We use $\log (N/dft)$ instead of N/dft to “dampen” the effect

Exercise

- Compute the IDF term for the following with log and without log.

Figure out N

Term	Df(t)	IDF without log	IDF with log
National	1	10 000 000	
party	100	10 000 0	
John	1 000	10 000	
Key	10 000	10.....	
Government	100 000		
English	1000 000		
the	10 000 000		

TF-IDF Computation

- **TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear many more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

$$TF(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}}$$

- **IDF: Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

$$IDF(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$

Tf-idf weighting scheme

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$\text{tf-idf}_{t,d} = \log(1 + \text{tf}_{t,d}) \times \log(N/\text{df}_t)$$

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \log(N/\text{df}_t) - \text{alternative}$$

- Best known weighting scheme in information retrieval
- Increases with **the number of occurrences within a document**
- Increases with **the rarity of the term in the collection/corpus**

Exercise

- Compute the tf-idf weightings for the following 4 terms for the 4 terms in a corpus of 3 documents. Use unnormalized form for the term frequency

	doc1	doc2	doc3
car	27	4	24
auto	3	60	0
insurance	0	0	29
best	14	4	17

- Which term would be the best to use as a search term?

Similarity Computations

- How similar are these?
 - The dog bit the man.
 - The man bit the dog.
 - When the dog bit the man.

Similarity Distance conditions

- Let x and y be any two objects in a set and $d(x, y)$ be the distance between x and y .
- Conditions
 - 1) The distance between any two points must be non negative, that is, $d(x, y) \geq 0$.
 - 2) The distance between two objects must be zero if and only if the two objects are identical, that is, $d(x, y) = 0$ if and only if $x = y$.
 - 3) Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x , ie. $d(x, y) = d(y, x)$.
 - 4) The measure must satisfy the triangle inequality, which is
 - 5)
$$d(x, z) \leq d(x, y) + d(y, z).$$

Euclidean Distance

■ Is the default measure metric

- Measuring distance between text documents, given two documents d_a and d_b represented by their term vectors t_a and t_b respectively, the Euclidean distance of the two documents is defined as

$$D_E(\vec{t}_a, \vec{t}_b) = \left(\sum_{t=1}^m |w_{t,a} - w_{t,b}|^2 \right)^{1/2}$$

where the term set is $T = \{t_1, \dots, t_m\}$. A commonly used measure is *tfidf* value as term weights, that is, $w_{t,a} = \text{tfidf}(d_a, t)$.

Cosine Distance

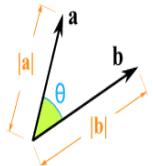
- Defined as the cosine of the angle between two vectors.

$$SIM_C(\vec{t}_a, \vec{t}_b) = \frac{\vec{t}_a \cdot \vec{t}_b}{|\vec{t}_a| \times |\vec{t}_b|},$$

- Cosine distance = dot product of the vectors divided by the product of the lengths.
- The smaller the value the smaller the distance, ie, more similar.

$a \cdot b$
This means the Dot Product of \mathbf{a} and \mathbf{b}

We can calculate the Dot Product of two vectors this way:


$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| \times |\mathbf{b}| \times \cos(\theta)$$

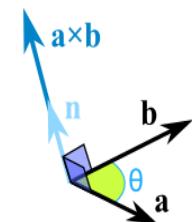
Where:
 $|\mathbf{a}|$ is the magnitude (length) of vector \mathbf{a}
 $|\mathbf{b}|$ is the magnitude (length) of vector \mathbf{b}
 θ is the angle between \mathbf{a} and \mathbf{b}

So we multiply the length of \mathbf{a} times the length of \mathbf{b} , then multiply by the cosine of the angle between \mathbf{a} and \mathbf{b}

WE CAN CALCULATE THE CROSS PRODUCT THIS WAY:

$$\mathbf{a} \times \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \sin(\theta) \mathbf{n}$$

- $|\mathbf{a}|$ is the magnitude (length) of vector \mathbf{a}
- $|\mathbf{b}|$ is the magnitude (length) of vector \mathbf{b}
- θ is the angle between \mathbf{a} and \mathbf{b}
- \mathbf{n} is the unit vector at right angles to both \mathbf{a} and \mathbf{b}



So the **length** is: the length of \mathbf{a} times the length of \mathbf{b} times the sine of the angle between \mathbf{a} and \mathbf{b} ,

Then we multiply by the vector \mathbf{n} to make sure it heads in the right **direction** (at right angles to both \mathbf{a} and \mathbf{b}).

Jaccard Coefficient

- Compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms.

$$SIM_J(\vec{t}_a, \vec{t}_b) = \frac{\vec{t}_a \cdot \vec{t}_b}{|\vec{t}_a|^2 + |\vec{t}_b|^2 - \vec{t}_a \cdot \vec{t}_b}.$$

- Value ranges between 0 and 1, 1 when exactly same and 0 when no terms are common.
- Another way to define Jaccard is
 - $|\text{Intersection } (A,B)| / |\text{Union } (A,B)|$

Levenshtein Distance

- Also known as the **edit distance**.
- Is the minimum number of single character edits (insertions, deletions or substitutions) required to change one sentence into another.

Hamming Distance

- Between two strings of **equal** length is the number of positions at which the corresponding symbols are different.
- ie, the minimum number of substitutions required to change one string into the other.
- Or (or originally) the minimum number of errors that could have transformed one string into the other.

Pearson Correlation Coefficient

Pearson's correlation coefficient is another measure of the extent to which two vectors are related. There are different forms of the Pearson correlation coefficient formula. Given the term set $T = \{t_1, \dots, t_m\}$, a commonly used form is

$$SIM_P(\vec{t}_a, \vec{t}_b) = \frac{m \sum_{t=1}^m w_{t,a} \times w_{t,b} - TF_a \times TF_b}{\sqrt{[m \sum_{t=1}^m w_{t,a}^2 - TF_a^2][m \sum_{t=1}^m w_{t,b}^2 - TF_b^2]}}$$

where $TF_a = \sum_{t=1}^m w_{t,a}$ and $TF_b = \sum_{t=1}^m w_{t,b}$.

This is also a similarity measure. However, unlike the other measures, it ranges from $+1$ to -1 and it is 1 when $\vec{t}_a = \vec{t}_b$. In subsequent experiments we use the corresponding distance measure, which is $D_P = 1 - SIM_P$ when $SIM_P \geq 0$ and $D_P = |SIM_P|$ when $SIM_P < 0$.

Demo Code TFIDF

COMP814 – Text Mining

Word Embedding

So far...

- Word to vector models is able to entail the importance of a word (TFIDF) and the frequency of particular words in documents.
- Can we embed semantics into words so that we can express that **king** and **queen** are related and **queen** and **woman** are related?
- Or, that **queen** and **woman** are more related than **queen** and **tree**.

Language Models

- Count based methods
 - Compute the statistics of how often some word co-occurs with its neighbour in a large corpus.
 - BOW models, TF and TFIDF models.
 - **Sparse** vectors.
- Predictive models
 - Directly try to predict a word from its neighbours from learned small, **dense** embedding vectors.
 - Word2vec is a computationally efficient predictive model for learning word embeddings from raw text.
 - Continuous BOW (CBOW) and **Skip-gram** are two predictive models.

Two forms of meaning representation

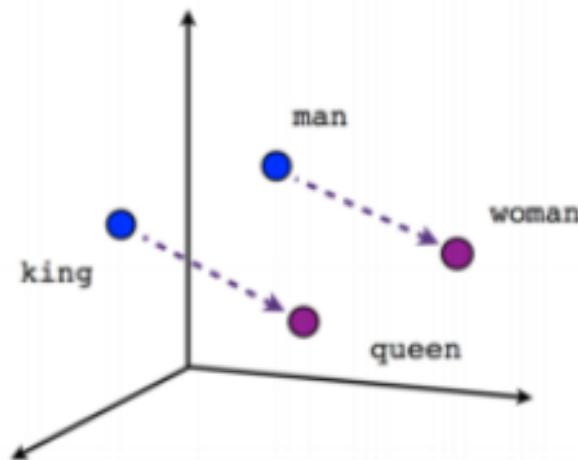
- **Denotational:** representation of meanings/concepts by symbols, art, set of alphabets
- **Distributional:** meaning determined by other neighbouring words (context) in which a word appears.

government debt problems turning into banking crises as has happened in
saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Word Embedding

- Words are mapped into an n-dimensional space
- Enables us to represent “semantics” between words.
- Skipgram and CBOW models by Stanford



Male-Female



Verb tense

Word Embedding models

- Predict the (context) neighbouring words given a focus word
- Or predict the focus word given a context

Vector Representations

- We want to be able to represent that “King” and “Queen” are related.
- **Word embeddings:** where semantically similar words are mapped to nearby points.
- Projections on an n-dimensional plane.
- Based on distributional hypothesis: words that appear in the same context share semantic meaning.
- Two models:
 - **Continuous Bag or Words(CBOW)** : predicts target word from context words
 - **Skip-gram model:** does the inverse, ie. Predicts context words from the target word.

Skip-gram model

- “*the quick brown fox jumped over the lazy dog*”
- Window - words to the left and to the right
 - Focus word: *quick*
 - Context words: the and brown for window size of 1.
 - Words: **the** and **brown**
 - Whole dataset: (*quick, the*), (*quick, brown*), (*brown, quick*), (*brown, fox*),
- The problem decomposes to prediction of (*input, output*) pairs.
 - For a large enough corpus this will capture context of the target word.
 - Models semantically or otherwise related words. ie. Context for a word.

Skip-Gram model

- Train a NN to on data and use the weights vector as the look up.
- Same principle applies for CBOW but in inverse.

An example for window size 2

Source Text

Training Samples

The quick brown fox jumps over the lazy dog. →

(the, quick)
(the, brown)

The quick brown fox jumps over the lazy dog. →

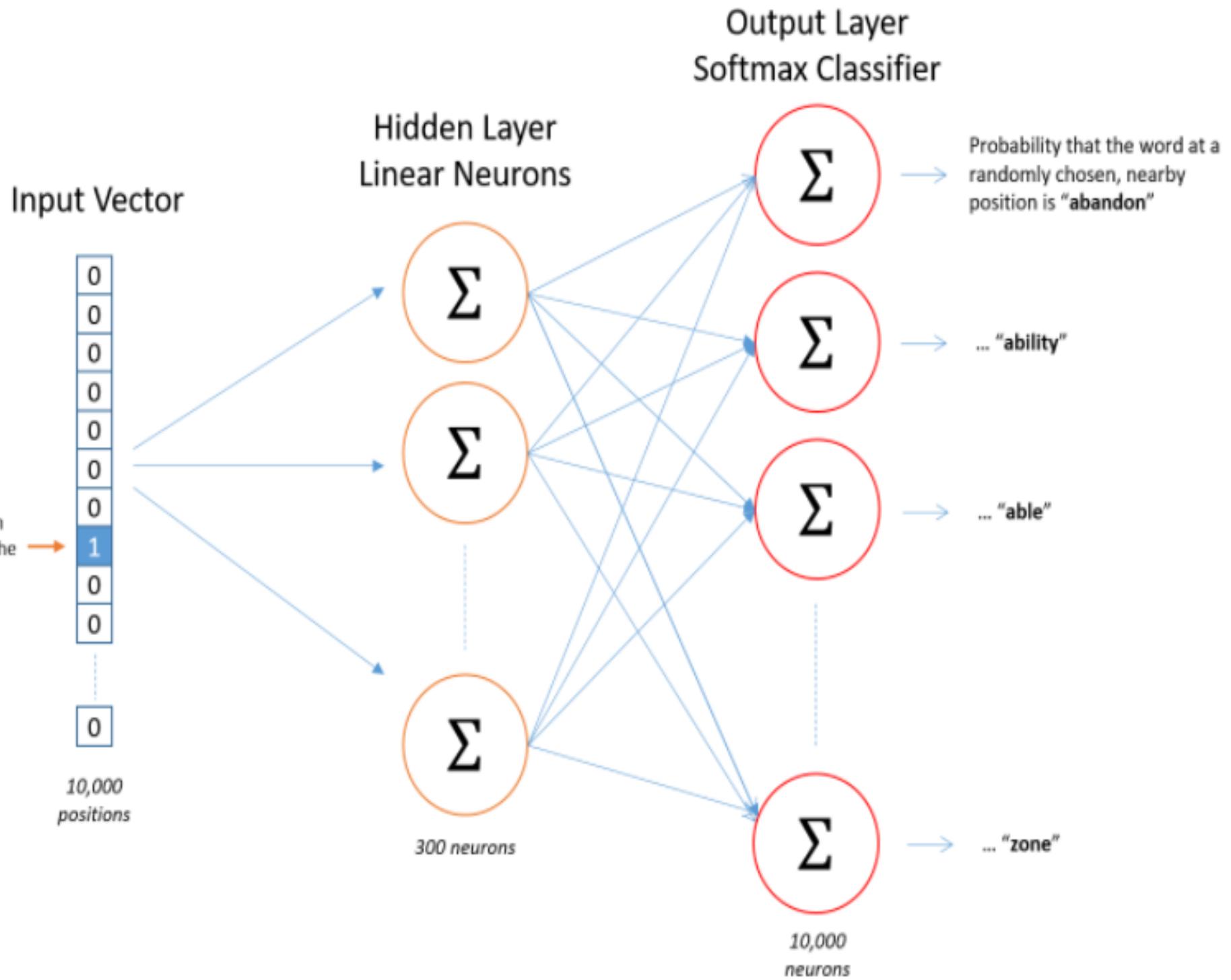
(quick, the)
(quick, brown)
(quick, fox)

The quick brown fox jumps over the lazy dog. →

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)

The quick brown fox jumps over the lazy dog. →

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

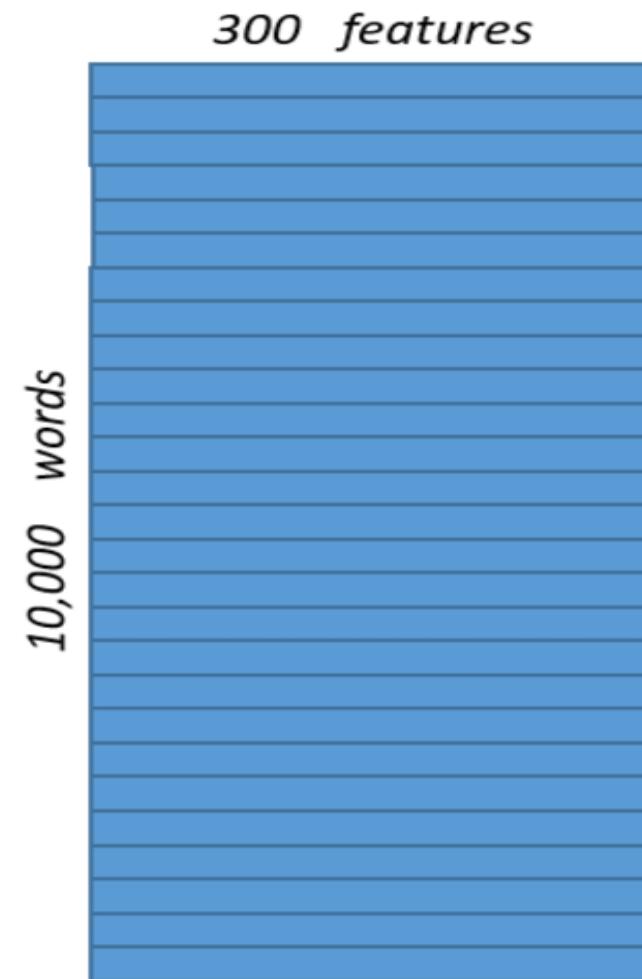
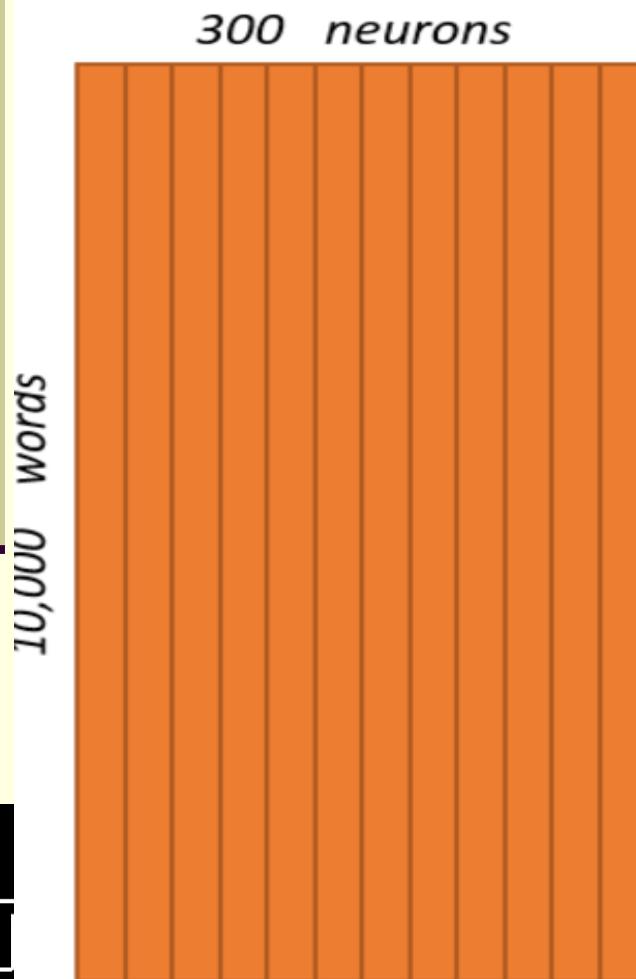


The matrix functions as lookup table

Hidden Layer
Weight Matrix



*Word Vector
Lookup Table!*



Word Embedding using NN

- Unsupervised learning
- Algorithm uses NN for word vector representation
- Does not use NN in a conventional sense
- Uses the weights as the embedding matrix.

Matrix as a lookup table

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{1} & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10} & \boxed{12} & \boxed{19} \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

GloVe: Global Vectors for Word Representation

- Developed by Stanford as open source project.
- Trained on Wikipedia
- Global Vectors for Word Representation ([GloVe](#)) is provided by Stanford NLP team. Stanford provides various models from 25, 50 , 100, 200 to 300 dimensions based on 2, 6, 42, 840 billion tokens.

Snapshot of Glove

Glove.6B.50d.txt

vuo1 -0.81467 -0.54463 0.63766 -1.2146 0.019283 -1.4186 1.1791 -0.83274 -0.1101 0.17914 0.057443 0.95155 0.11615 -0.542 -0.44769 0.72886 0.25029 0.89701 0.13379 -0.8196 -0.23162 0.30085 -0.27718 1.775 1.2592 -0.35793 -0.24092 0.18772 -0.35008 -0.302
tavčar -0.42006 -0.44263 0.81196 -0.0047469 -0.61735 -0.028478 0.5101 1.4337 -0.22903 0.59825 0.63654 0.18272 0.12608 0.39912 0.01056 -0.6746 -0.027251 -0.0023884 0.75227 0.25848 -0.96552 0.21364 -0.081466 0.14021 1.0323 1.0365 0.70845 -0.099035 0.21728 0.48
fardell -0.33704 -0.9913 0.30297 -0.39216 -0.18761 -0.22253 0.017483 0.94932 0.43463 0.010208 0.20107 -0.076423 0.52859 -0.076757 -0.34678 0.075904 -0.42472 -0.32924 0.026745 0.074706 -0.19253 0.39236 0.13884 -0.516 0.72429 1.172 0.17368 -0.2065 0.13477 0.89
nurc 0.20211 0.30983 1.1775 0.076136 -0.6354 -0.091129 0.67233 0.0091902 0.67643 -0.33846 0.53264 0.28375 0.57154 0.70066 -0.97071 -0.062547 -0.018489 0.18681 0.36407 0.20892 -0.79968 0.41133 -0.15771 -0.53014 0.33026 1.4519 0.94849 -0.93601 -0.86101 0.48532
sauzal -0.40113 0.69858 0.058603 0.021985 -0.013685 -1.3266 0.38678 0.74245 0.79957 1.3198 -0.39142 -1.5502 0.48349 0.067254 -0.31846 -0.13412 0.046513 0.34346 0.95132 1.3061 -0.38022 0.4687 0.24481 0.65362 -0.046524 0.82016 -0.44002 1.0456 1.0559 0.10355 -0.1
gousmi 0.080899 -0.82028 0.96931 0.069916 -0.85533 0.26591 0.31307 -0.054876 0.14654 0.079161 0.67231 -0.46226 0.2368 -0.024785 -0.87202 0.12938 -0.39201 0.5617 0.59757 0.51352 -1.1742 0.37081 -0.15366 -0.471 0.69054 0.59882 0.46039 0.23888 -0.61784 0.65993
numskulls -0.34393 -0.5313 0.40101 -0.69059 0.27025 -0.11384 0.20543 0.36861 -0.063199 0.35953 -0.41987 0.18358 0.32359 0.70078 0.39738 -0.73859 0.26871 0.11734 0.39961 -0.066761 -0.45876 0.1053 0.29265 0.25155 0.61768 1.0692 -0.35211 -0.6991 0.25961 0.0213
visalakshi 0.2622 0.22444 -0.2133 -0.38308 0.20452 -0.84946 0.72275 0.84874 1.1191 0.24385 0.19162 0.49844 0.93509 -0.29487 -0.30063 0.4565 0.11722 0.067107 0.79719 0.94901 -1.0595 0.55656 0.00024112 -0.012278 0.31597 0.85238 1.2608 -0.34851 0.057652 0.35867
lieserl 0.41247 0.042613 -0.055336 -0.28202 0.47983 -0.1222 -0.082391 0.63397 -0.0065611 0.52192 0.47806 0.58869 -0.0021574 -0.37093 0.14592 0.21515 -0.48807 0.47691 0.5772 0.15688 -0.39073 0.34066 -0.60596 -0.30251 0.72078 0.54174 -0.37043 0.2248 0.34997 0.
ulundurpet 0.12802 -0.42887 0.39333 -0.18398 -0.99918 -0.45869 0.75997 0.45617 0.57099 0.70857 -0.10079 -0.81964 0.75076 -0.1217 -0.53979 0.28039 0.29952 0.70266 0.52826 0.55219 0.097351 -0.73508 0.12458 0.59674 -0.083382 0.44398 0.34162 0.74523 -0.30224 0.4
arpita -0.47987 -0.24207 -0.46097 0.11266 -0.42204 -0.12263 0.26864 0.44569 -0.34467 -0.19032 0.30396 0.45327 0.45422 -0.49473 0.20065 0.22533 -0.22697 0.10112 1.4572 0.66894 -0.27476 0.32178 0.053409 0.21384 0.75779 1.1381 0.51047 -0.080393 -0.13708 0.08862
ineta -1.4196 0.00048096 0.25966 0.30973 -0.2295 0.52714 0.32103 0.47756 -0.66579 -0.031912 0.52059 -0.35151 0.46014 0.020467 0.27494 -0.004111 -0.076034 -0.54099 0.79975 -0.094049 -0.45483 -0.48901 -0.12172 0.38061 0.19902 1.4841 1.1206 0.25895 0.5096 -0.01
whirlow -0.96523 0.6683 -0.25507 -0.036634 -0.30133 -0.0066741 -0.53408 0.31895 0.26328 -0.34029 -0.37309 -0.42184 0.70655 0.38561 0.016944 0.66546 0.3297 0.81763 0.43584 -0.33597 0.16891 -0.12377 -0.92441 0.1741 0.55521 0.72956 0.22202 0.99021 -0.095615 -0.
wanz -0.47564 -0.57784 0.11614 -0.49852 -0.69533 0.079624 0.24805 0.18032 0.62608 -0.52454 0.10518 0.81569 -0.19997 -0.62045 -0.020506 -0.25143 -0.070136 0.76055 0.58671 -0.95001 1.2227 -1.3232 -0.67015 -0.0237 0.15597 0.29452 -0.96432 -0.32938 0.51445 0.058
t22 -1.2211 0.27066 0.82415 -0.20607 -0.84074 -1.3603 0.44268 0.24782 0.36704 0.05251 -0.37049 -0.42112 0.13556 1.0905 -0.048555 -0.035829 0.15691 -0.24283 0.57121 0.072466 0.52102 0.29082 0.18695 0.3848 0.063254 1.0079 -0.42316 -0.22858 0.20228 0.4123 -0.52
slann -0.39779 -0.33714 0.36308 -0.41577 -0.24234 -0.020363 0.22913 0.55821 -0.49158 -0.157 -0.35809 -0.49801 0.46588 0.47346 -0.42498 -0.3301 0.13551 -0.057302 1.0107 0.18716 -0.6082 -0.30279 -0.66632 0.2081 0.35085 1.214 -0.2607 -0.66409 -0.38749 0.44655 -0.
kunimoto -0.39396 -0.44632 0.10236 0.036734 -0.56225 0.14542 0.27142 0.4771 -0.28237 0.40496 0.30712 0.11696 0.086197 0.58789 -0.494 -0.24348 -0.15639 -0.26068 0.76081 -0.25517 0.075262 -0.21533 -0.63993 -0.28086 0.088038 0.94451 0.75477 0.23091 -0.16676 0.0
altares 0.034001 -0.18371 -0.087926 -0.18267 -0.21546 -0.7086 0.27737 0.72547 0.028812 -0.069737 -0.093149 0.22339 0.54237 0.9837 0.44975 -0.8776 0.3655 0.073673 0.56261 0.55366 -0.44146 -0.71219 -0.19037 0.49301 0.15332 1.3274 0.068386 0.16759 0.033585 -0.
cipolletti -1.0088 -0.55374 -0.92561 0.13024 -0.07803 -0.64357 0.40808 0.6215 0.37307 0.4798 0.036388 -0.29611 0.73975 -0.61042 0.057497 -0.57967 0.24621 -0.17201 0.25262 0.44779 0.097345 -0.24989 -0.31964 0.10106 0.25032 0.90412 0.66423 0.0069118 -0.23768 0.
sherwoods 0.019383 -0.49104 0.24497 -0.67895 -0.30405 -0.60039 -0.24945 0.62099 -0.13443 -0.067268 0.17334 0.16677 0.36566 0.43017 -0.38987 -0.034329 0.071327 -0.43188 0.85444 0.427 -0.13526 -0.16348 0.246 -0.1535 0.026835 1.0439 -0.14284 -0.36727 -0.54334 -0.
suliana 0.28677 -0.90521 -0.36906 0.26354 -0.094759 -0.24936 1.1724 0.71973 0.06801 0.50144 0.18845 -1.1023 0.77814 -0.63758 -0.49451 -0.13198 -0.204 0.95657 0.67583 0.76056 -0.35156 -0.38642 0.074414 0.77701 0.29037 0.5988 0.22375 0.29643 -0.17826 -0.01398
29km -0.28064 0.15323 -0.30591 -0.90841 0.32428 -0.64677 0.6134 0.52447 0.2436 -0.068487 -0.93159 -0.88973 -0.10408 0.33877 0.16929 -0.095657 0.51134 -0.65365 -0.008701 0.21586 0.39866 0.0046171 -0.27819 0.54804 0.36839 0.73974 0.37151 0.41071 0.2484 0.81776
sigarms -0.74397 0.082164 -0.0091471 0.4129 -0.42255 0.10125 -0.18602 0.21051 -0.59037 0.66988 -0.13711 0.63894 0.042985 -0.63123 -0.62241 0.024485 -0.87917 -0.53674 0.56378 0.11545 0.37125 -0.014403 0.23307 0.15689 -0.76922 1.1551 -0.32294 -0.52755 0.31421
katuna -0.30016 -0.80268 -0.46637 -0.29822 -1.032 -1.0705 0.84562 0.70225 0.11996 -0.7183 -0.61271 -0.92747 0.29668 -0.2894 -0.21143 0.27755 0.19278 0.26899 0.67493 0.99469 0.18606 0.15348 -0.30083 0.99714 0.53874 0.15414 0.79983 0.60903 0.55637 0.046631 -0.
AUT
16/18

Word2Vec/Skipgram explained further – youtube video

- <https://www.youtube.com/watch?v=UqRCEmrV1gQ>

Skipgram code demo

Text Categorization/Classification

Deep Learning

The problem of (Text) Classification/Categorization

- **Theoretical:** how can objects be differentiated?
- **Practical:** what models & algorithms work well for differentiating among classes of objects?
- Plan for the lecture:
 - Look at a few motivating examples of categorization
 - Define the problem formally for texts
 - Cover simple two methods for categorization in some detail
 - **K-NN** and **Naïve Bayes**
 - These methods form the building blocks of other classification models
 - Look at how these methods perform and what they learn
 - Classification using Deep Learning
 - Lab on Deep Learning and conventional algorithms.

Conceptual categorization

- Identify the features for classification into Cup and Glass



Email classification: Is this spam?

From: "" <takworlld@hotmail.com>

Subject: real estate is the only way...

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

Click Below to order:

<http://www.wholesaledaily.com/sales/nmd.htm>

Text Categorization Examples

Assign labels to each document or web-page:

- Labels are most often topics such as Yahoo-categories
 - e.g., "*finance*," "*sports*," "*news>world>asia>business*"
- Labels may be genres
 - e.g., "*editorials*" "*movie-reviews*" "*news*"
- Labels may be opinion
 - e.g., "*like*", "*hate*", "*neutral*"
- Labels may be domain-specific binary
 - e.g., "*interesting-to-me*" : "*not-interesting-to-me*"
 - e.g., "*spam*" : "*not-spam*"
 - e.g., "*contains adult language*" : "*doesn't*"

Shopping on the Web

- Suppose you want to buy a cappuccino maker as a gift
 - try Google for “cappuccino maker”
 - try Yahoo! Shopping for “cappuccino maker”



File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Stop Refresh

Address http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=cappuccino+maker

Go Links

[Advanced Search](#) [Preferences](#) [Language Tools](#) [Search Tips](#)

cappuccino maker

Google Search

[Web](#) [Images](#) [Groups](#) [Directory](#)Searched the web for **cappuccino maker**.

Results 1 - 10 of about 17,800. Search took 0.09 seconds.

[Espresso Machines at Cooking.com - Best brands, selection, prices!](#)

Sponsored Link

www.cooking.com Cookware, appliances, cutlery, cook's tools, bakeware and more!

[Cappuccino makers at low prices CLICK HERE](#)

Sponsored Link

www.goodmans.net Lowest prices, fast shipping & 30 day money back guarantee.

Category: [Shopping](#) > [Home and Garden](#) > [Kitchen and Dining](#) > [Appliances](#) > [Coffee Makers](#)**[Cappuccino Maker from Nespresso Store - Four Unique Models](#)**

Sponsored Links

... Click Here for **cappuccino maker** from Nespresso **Cappuccino Maker**: Unique espresso machine / **cappuccino maker** and capsule system created by Nestlé, the ...www.nespressostore.com/cappuccino-maker-d.html - 10k - [Cached](#) - [Similar pages](#)

[Espresso Machines & Coffee](#)
 Espresso Machines & Espresso Coffee
 No charge for Shipping
[www.1stincoffee.com](#)
 Interest:

[DeLonghi 10 Cup Coffee Cappuccino Maker](#)

[Coffee & Espresso Machine](#)
 Krups, Delonghi, Capresso, Saeco,
 Cuisinart KitchenAid,Solis,LaPavoni
[everythingbagel.com](#)
 Interest:

... DeLonghi **Cappuccino** Makers DeLonghi 10 Cup Coffee **Cappuccino Maker** PreviousItem Item 2 of 2 Next Item, DeLonghi 10 Cup Coffee **Cappuccino Maker**, ...www.globalmart.com/page/c/cc80.htm - 20k - [Cached](#) - [Similar pages](#)**[Stainless Steel Espresso/Cappuccino Maker](#)**

[Coffee For Less](#)
 Buy the Lavazza Espresso machine
 for \$900
[www.coffeeforless.com](#)
 Interest:

... Features: Separate controls for **cappuccino**. 8 high. Gift box. Great camping item.More Coffee Makers. **SS-Cappuccino-Maker** Retail price: \$82.00 Our price: \$69.75.www.1-800-espresso.com/s-s-cappuccino-maker.html - 5k - [Cached](#) - [Similar pages](#)**[Amazon.com: buying info: Melitta Espresso/Cappuccino Maker \(4- ...\)](#)**

[Cappuccino maker - Sears](#)
 Shop Sears.com & get great deals on
 Cappuccino maker and more!
[www.sears.com](#)

... Melitta Espresso/**Cappuccino Maker** (4-cup) Our Price: \$29.99 Usually ships

within 24 hours Product Description Make coffee like the pros. ...

www.amazon.com/exec/obidos/ASIN/B00005OTY8/ - 39k - [Cached](#) - [Similar pages](#)**[Cappuccino maker instructions...](#)**

Internet

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://search.shopping.yahoo.com/search/all/_y=s:22708228,d:14489115,p:s;search?is=1&p=cappuccino+maker&tool=0&did=

Go Links



Shopping Home - Yahoo! - Help

Four ways to shop: Shopping Used Auctions Classifieds

Welcome, guest

My Shopping Account - View Cart - Sign In

Search Results Found 306 products in 113 stores for "cappuccino maker"

Shopping Home

Search

cappuccino

Search in:

 Shopping Only

All of Shopping

 Shopping, Auctions & Classifieds[Advanced Search](#) · [Store Search](#)

Narrow By Price

[\\$1 - \\$20](#) (8)[\\$25 - \\$50](#) (49)[\\$50 - \\$100](#) (88)[\\$100 - \\$200](#) (72)[\\$200 - \\$400](#) (58)[\\$400 - \\$2000](#) (31)

By Department

[Electronics & Camera](#) (36)[Gourmet & Kitchen](#) (152)[Home, Garden, & Pets](#)
(80)[Music](#) (1)

View by: store | relevance | price

< Previous | Showing 1-10 of 113 | [Next >](#)Stores [see all stores](#) with name or description matching "cappuccino maker"

Search Results Found 306 products in 113 stores for "cappuccino maker"

[Overstock.com](#)

Featured

[Cuisinart Iced Cappuccino Maker](#)

\$56.99

Refreshing iced hot coffee drinks will be yours in minutes with the Cuisinart iced **cappuccino** and hot espresso **maker**. Enjoy 4 cups of iced or 8 cups of hot coffee at a time, as well as an attractive and innovative European design.· [See all matches at this store](#) (2)[JCPenney](#)

Featured

[Krups® Espresso/Cappuccino/Latte Maker](#)

\$99.99

[QVC](#)

Featured

[Briel Quick Froth Cappuccino Maker](#)

\$59.98

The Briel Quick Froth **Cappuccino Maker** is designed with an automatic milk frother. Simply slip it onto your espresso machines steam wand, turn the steam knob on and presto. It draws milk out of any container, perfectly froths it, then dispenses it.

Observations...

- Broad indexing & speedy search **alone** are not enough.
- **Organizational view** of data is critical for effective retrieval.
- Categorized data are easy for user to browse.
- Category **taxonomies** become most central in well-known web sites (Google, Bing, Yahoo!, Lycos, ...).

Some Text Categorization Applications

- Web pages organized into category hierarchies
- Journal articles indexed by subject categories (e.g., the Library of Congress, MEDLINE, etc.)
- Patents archived using *International Patent Classification*
- Patient records coded using international insurance categories
- E-mail message filtering
- News events tracked and filtered by topics
- Blogs filtered by topics, sentiments etc.

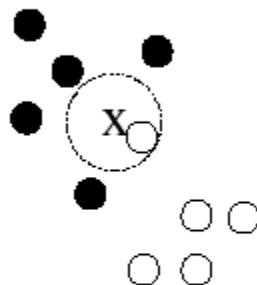
Approaches to Text Categorization (emphasizing the “k-Nearest Neighbor approach” and Naïve Bayes approach)

Key Components of Nearest Neighbor

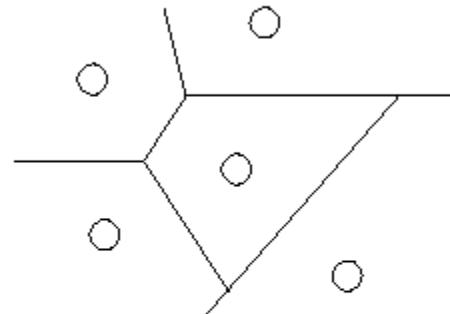
- “Similar” item: We need a functional definition of “similarity” if we want to apply this automatically.
- How many neighbors do we consider?
- Does each neighbor get the same weight?
- All categories in neighborhood? Most frequent only? How do we make the final decision?

1-Nearest Neighbor (graphically)

1-NN: assign "x" (new point) to the class of its nearest neighbor



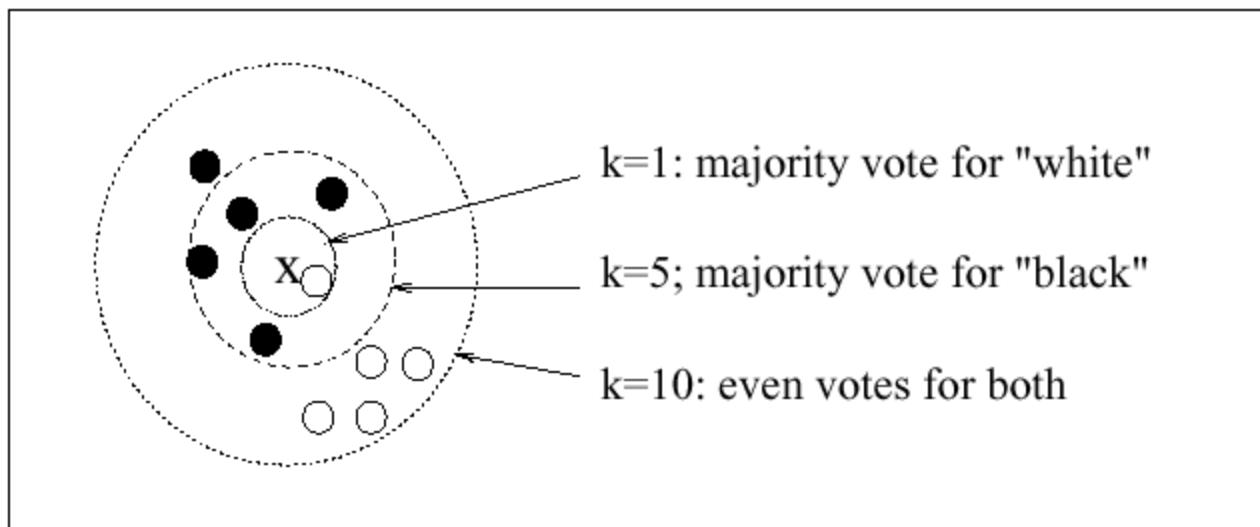
assign "x" to "white"



decision surface divided by points
("Voronoi diagram")

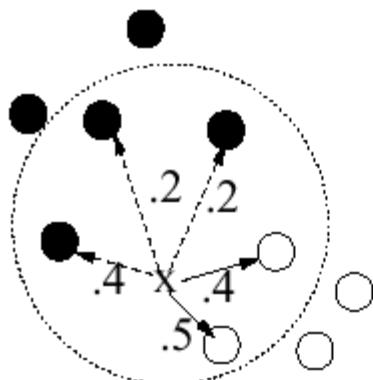
K-Nearest Neighbor using a *majority* voting scheme

K-Nearest Neighbor using a *majority* voting scheme



K-NN using a weighted-sum voting Scheme

k-NN using a weighted-sum voting scheme



kNN ($k = 5$)

Assign "white" to x because
the weighted sum of "whites" is
larger than the sum of "blacks".

Each neighbor is given a weight according to its nearness.

kNN for Text Categorization (Yang, SIGIR-1994)

- Represent documents as points (vectors).
- Define a similarity measure for pairwise documents.
- Tune parameter k for optimizing classification effectiveness.
- Choose a voting scheme (e.g., weighted sum) for scoring categories
- Threshold on the scores for classification decisions.

Possible Similarity Measures

■ Cosine similarity

$$\cos(\vec{x}, \vec{y}) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}$$

- Euclidean distance
- Jaccard Distance
- Kernel functions
- Kullback-Leibler distance (distance between two probability distributions)

NB Classifier parameter estimation

- $P(c_j)$
 - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \dots, x_n | c_j)$
 - Could only be estimated if a very large no. of training examples was available

NB Assumptions

■ Conditional Independence Assumption:

- The probability of observing the conjunction of attributes is equal to the **product** of the individual probabilities
- Features are independent of each other

$$C_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod P(x_i | c_j)$$

- Ignore position of words, else too many possibilities
- Use bag of words for probability computations

Training a NB classifier

- Extract **vocabulary** from training corpus
- Calculate $P(c_j)$ and $P(x_k|c_j)$ terms
 - For each c_j in C do
 - $docs_j$ – subset of documents for which class is c_j
 - $P(C_j) = \frac{|docs_j|}{|total \# docuemnts|}$
 - $Text_j$ – single document containing all $docs_j$
 - For each word x_k in vocabulary
 - N_k – no. of occurrences of x_k in $Text_j$
 - $P(x_k|c_j) = \frac{|n_k+1|}{n+|Vocabulary|}$

Naive Bayes is Not So Naive

- Naïve Bayes: First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms

Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
- Robust to Irrelevant Features

Irrelevant Features cancel each other without affecting results
Instead Decision Trees can **heavily** suffer from this.
- Very good in domains with many equally important features

Decision Trees suffer from *fragmentation* in such cases – especially if little data available
- A good dependable baseline for text classification (but not the best)!
- Optimal if the Independence Assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for the problem
- Very Fast: Learning with one pass over the data; testing linear in the number of attributes, and document collection size
- Low Storage requirements

Alternative counting scheme – TF-IDF

- Term frequency-inverse document frequency
- A statistical measure used to evaluate how important a word is to a document in a collection or corpus (or a category)
- The importance increases with frequency but is offset by the frequency of words in the corpus.
- A simple ranking function can be to sum up the tf-idf for each query term.
- Can also be used for stop word filtering. That is, discard the lowest tf-idf terms.

Approaches to Automated Text Categorization

- Decision trees
- Naïve Bayes
- K nearest neighbour
- Perceptron
- Support Vector Machines
- CNN
- Many others...

Evaluation Methodology and Performance Metrics

Classification Performance Measures

- Given n test documents and m classes in consideration, a classifier makes $n \times m$ binary decisions. A two-by-two contingency table can be computed for each class.
- Referred to as **confusion matrix**

	Machine Yes	Machine No
Ground Truth Yes	TP	FN
Ground Truth NO	FP	TN

Classification Performance Measures

	Machine Yes	Machine No
Ground Truth Yes	TP	FN
Ground Truth NO	FP	TN

P = Precision

R = Recall

F₁ = F-value or F-score

A = Accuracy

$$P = \frac{TP}{TP+FP}$$

$$F_1 = \frac{2PR}{P+R}$$

$$R = \frac{TP}{TP+FN}$$

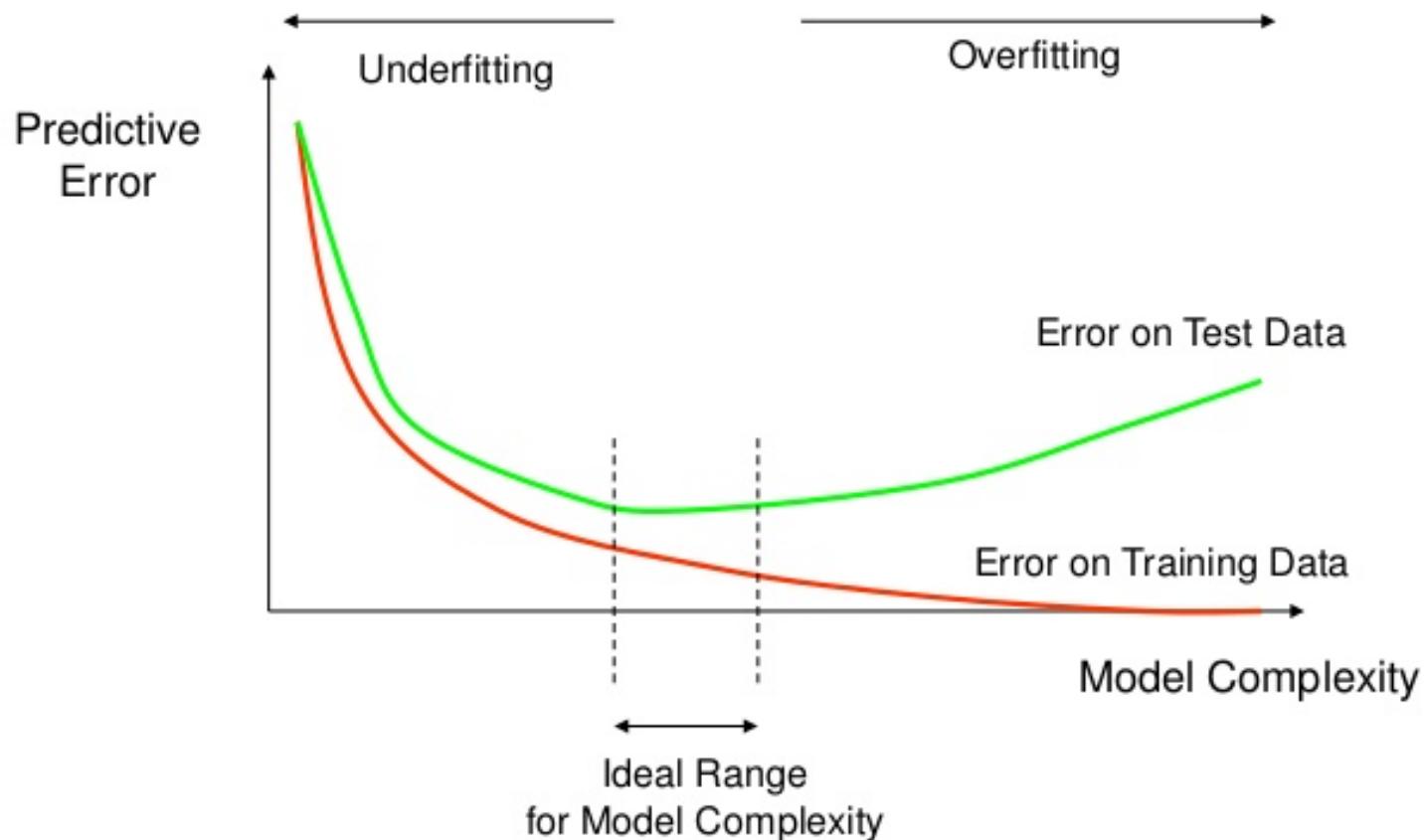
$$A = \frac{TP+TN}{TP+FP+FN+TN}$$

What about more than 2 classes?

Evaluating Results

- Results on training corpus might not be mirrored in the real world. Why?
- Want to avoid **overfitting**.
- Need separate test data. Hold out 10–20% of the corpus.
 - N-fold cross-validation
 - Leave-one-out cross-validation
- Separate **development/train** and **validation** and **test** sets.
- Need a measure of performance and comparison to **baseline**.

Overfitting/Underfitting

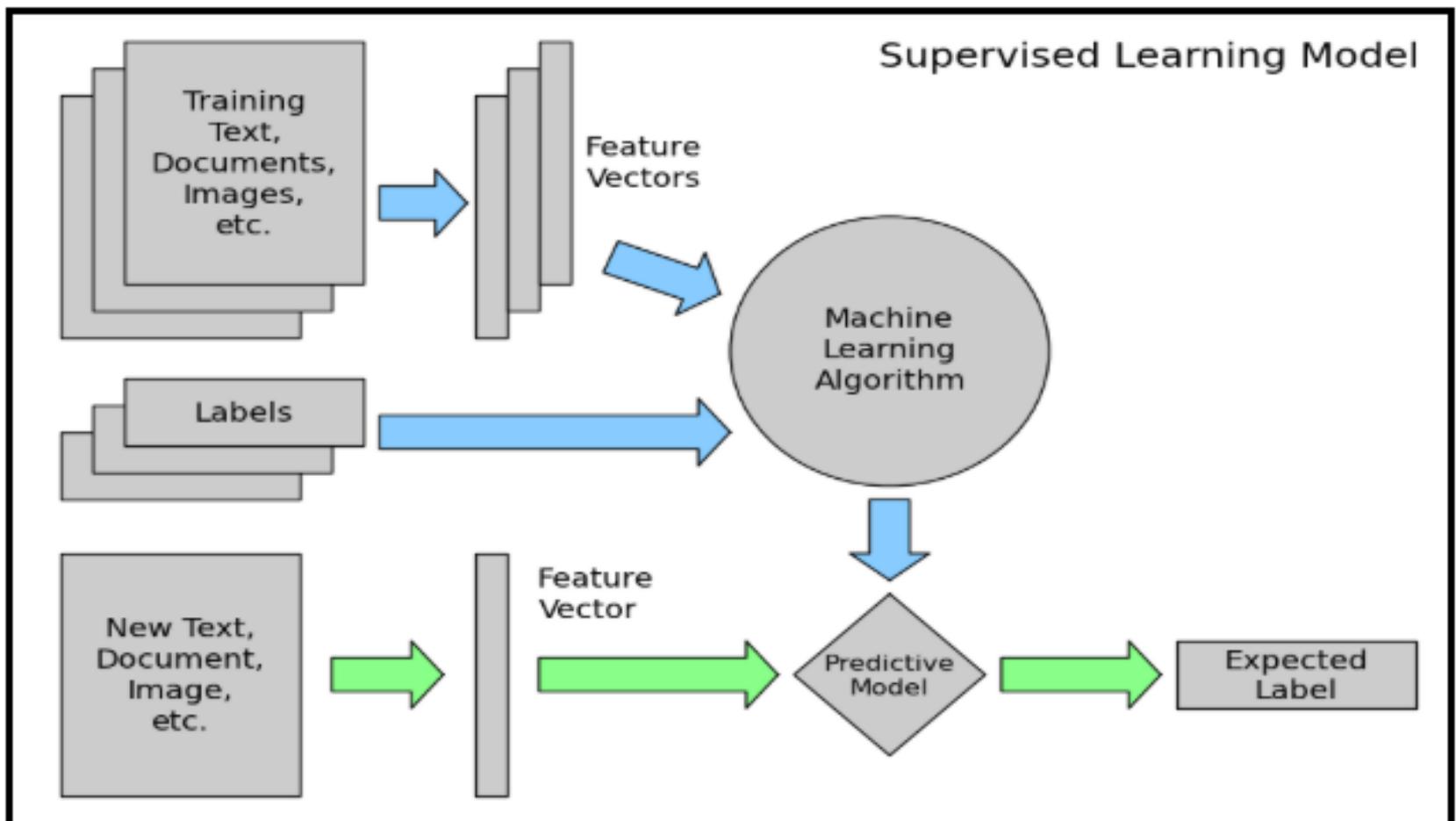


Baseline performance

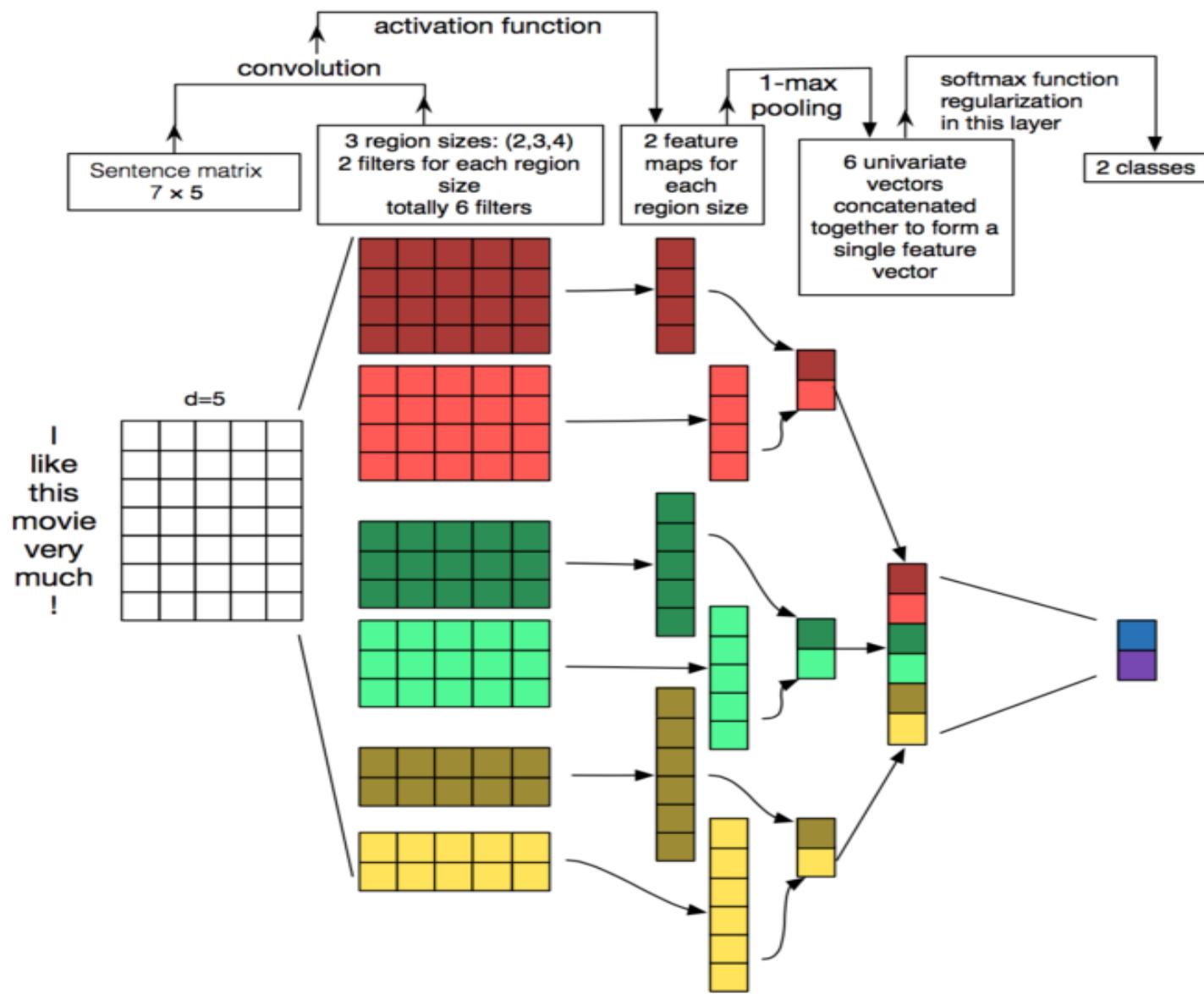
- **Baseline:** The minimum performance level that you're trying to improve on.
- Could be performance of competing system.
- Could be performance of dumb but easy method:
 - Random choice, most-frequent answer, very simple heuristic, ...
- Comparison should be made on the same test data for results to be fully meaningful.
- **Gold standard** comparison – compared with best achieved results.
 - Leaderboard

Deep Learning

Classification using Deep Learning



Convolutional Neural Network (CNN)



Matrix Convolution

- Cross channel sum of the receptor region with a **kernel/mask** computed over a **sliding window** given a **stride** (and padding) plus a **bias term**.

2	2	1
3	1	-1
4	3	2

Input matrix (3x3)
no padding
1 channel

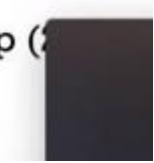
1	-1
-1	0

Kernel (2x2)
Stride 1
Bias = 2

$$1*2 - 1*2 - 1*3 + 0*1 + 2 = -1$$

-1	

Feature map (1x1)



Matrix Convolution

- Cross channel sum of the receptor region with a kernel/mask computed over a sliding window given a stride (and padding) plus a bias term.

2	2	1
3	1	-1
4	3	2

Input matrix (3x3)
no padding

1	-1
-1	0

Kernel (2x2)
Stride 1

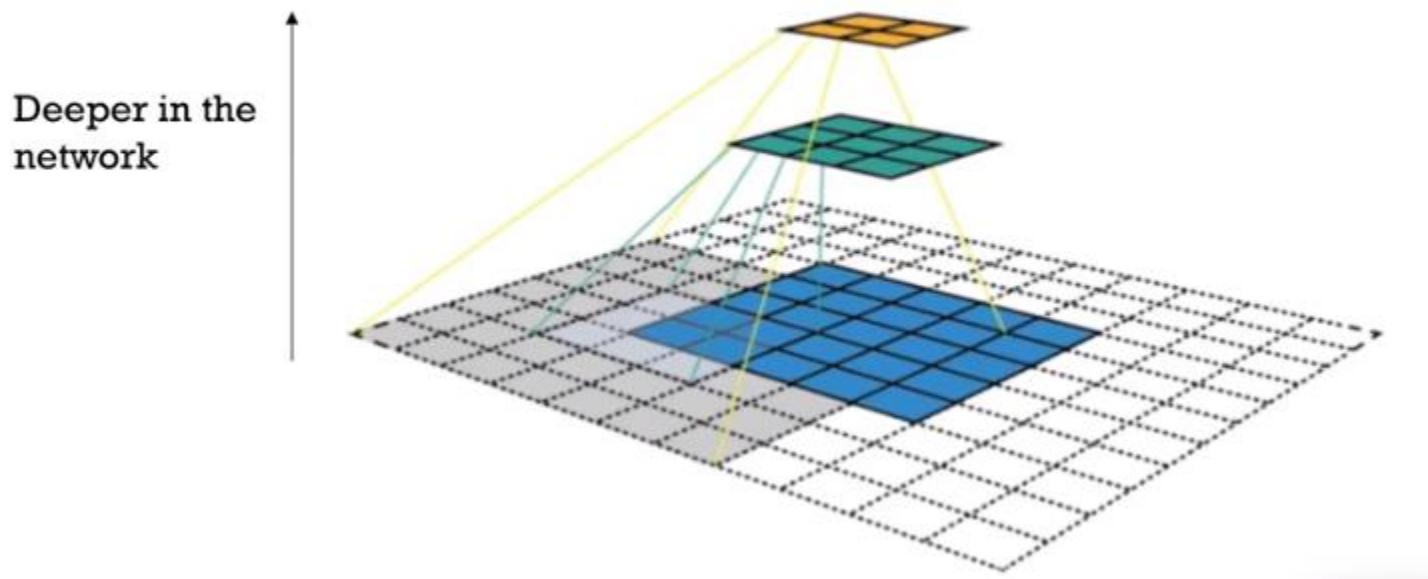
-1	2
0	1

Feature map (1x1)

$$\begin{aligned}1*2 - 1*2 - 1*3 + 0*1 + 2 &= -1 \\1*2 - 1*2 - 1*1 + 0*-1 + 2 &= 2 \\1*3 - 1*1 - 1*4 + 0*3 + 2 &= 0 \\1*1 - (-1)*1 - 1*3 + 0*2 + 2 &= 1\end{aligned}$$

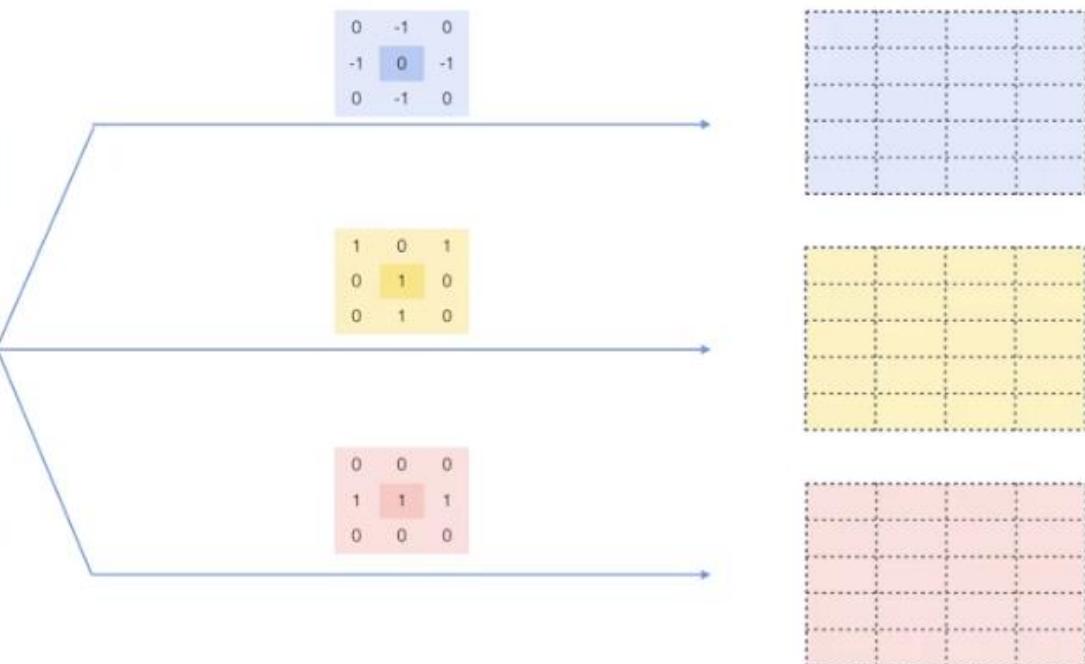
Convolution to reduce dimensions

- To prevent combinatorial explosion.



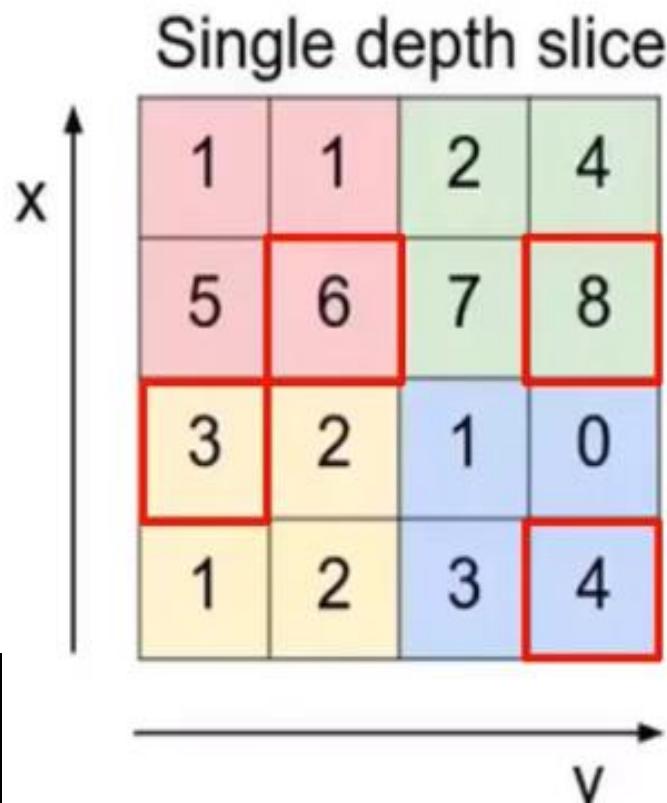
3 convolutions with different kernels

52	34	14	5
45	12	17	11
29	20	19	27
99	85	60	55
120	112	88	29

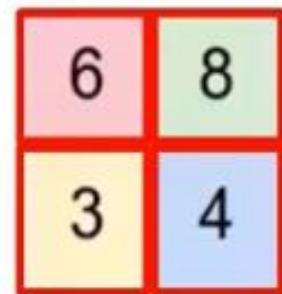


Max pooling

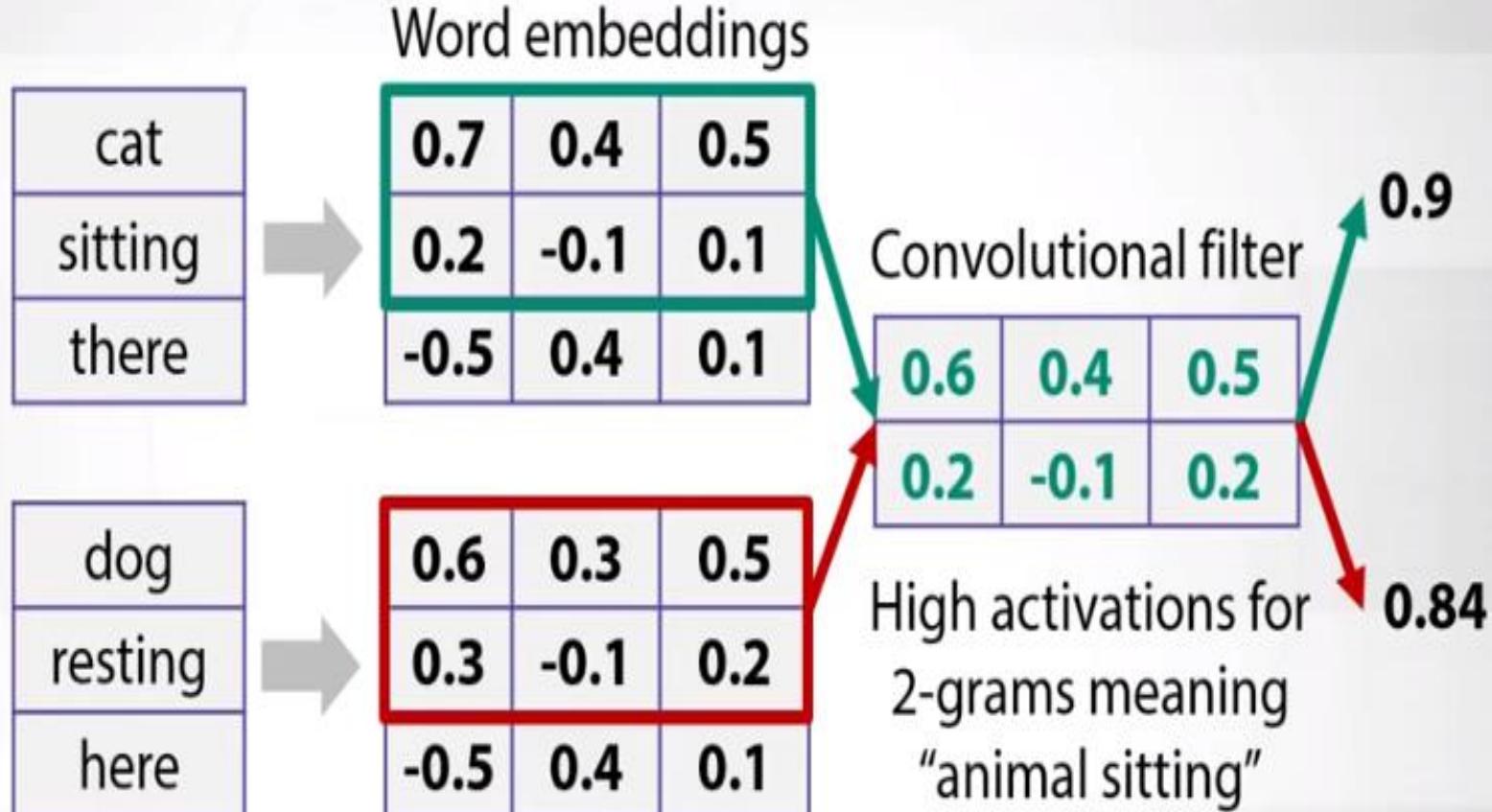
- Picks the biggest number from a window/region



max pool with 2x2 filters
and stride 2



Convolve to lower dimensional matrix



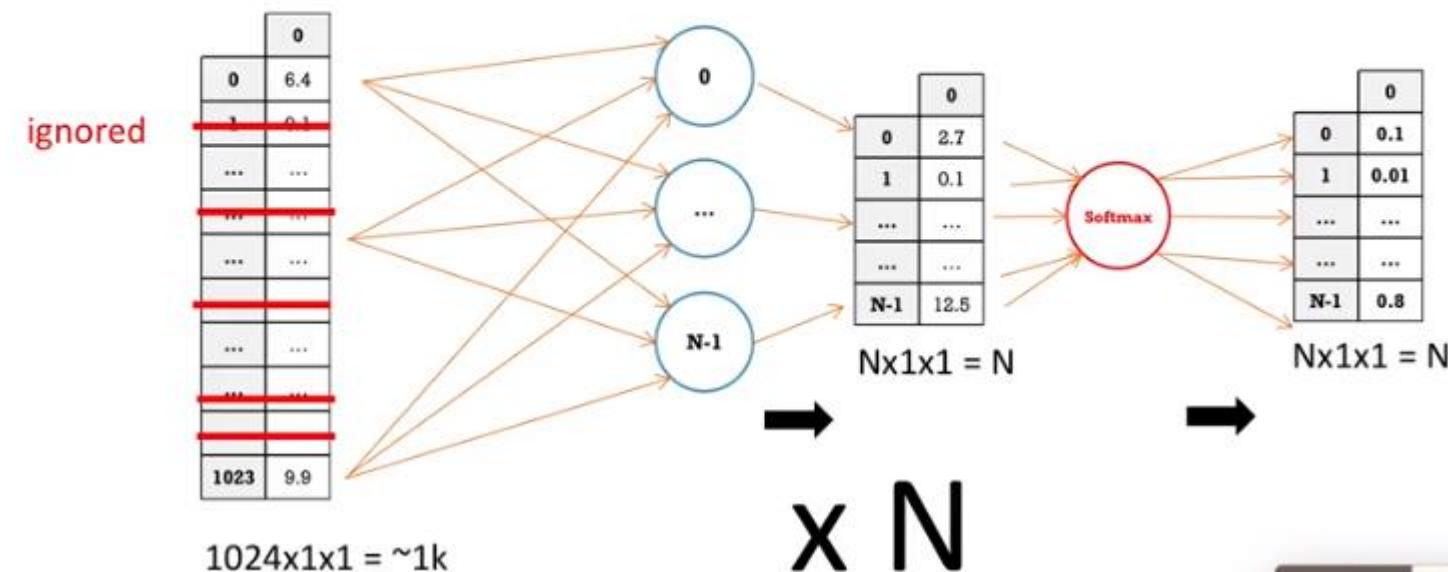
Max pooling



Fully connected neural net at the end.

- Flattened matrix, Dropout, softmax for n categories

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{N-1} e^{z_j}}$$



GloVe: Global Vectors for Word Representation

- GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.
- Can download the embedding to plug into CNN models instead of training your own.
- <https://nlp.stanford.edu/projects/glove/>

Deep Learning Summary

- Word embedding
 - N-grams
 - Sliding window
 - Strides
 - Kernels
 - Max pooling
 - Activation functions
 - Flattened matrix
-
- Fully connected NN
 - dropout
 - Softmax function
 - Back propagation
 - Loss function

Demo Code

Text Classification Demo

CNN Demo

text_sentoken (data)

Classification_Data_Computers (data)

COMP814 – Text Mining

Information Extraction (NER)

Test Details – Time, etc.

- The Mid semester test will be held in the lecture session in week 8, after the mid semester break.
 - This falls on **30 April, 2024**
- The test will be done online via Canvas.
 - The test can be accessed via the **Assignment tab on the left** in the home page or via the **link in module 8**.
- The test will **start at 2.30pm** (half an hour after the lecture start time)
- The test is timed for about an hour, however you will be given 1.5 hours in case of any technical/internet issues.

Test Details

- The test will be open book and you are free to access any online resource, including lecture notes and/or lab code.
- You can do the test from home or any other quite place with good internet. The lecture room will be open if you want to use this as quite space.
- You will be able to see all the questions in one go and you will be able to go back and review your answers before the submission.
- You will be prompted to submit at the end of the allocated one and half hours.
- Your test score will be available afterwards.

Test format.

- Each student will have different variations of the test from a questions bank.
- Some of the types of questions will be:
 - Multichoice
 - Multi answer (marks will be deducted for incorrect choices)
 - Fill in the blank
 - Matching
 - ordering

Named Entity Recognition and Classification

<PER>**Mr John Minto** </PER> taught COMP814 in <DATE>**semester 2 2023**</DATE>.

<PER>**Mr John Minto** </PER> also stood for the elections in <LOC>**Auckland**</LOC>.

<ORG>**Fisher and Pykal** </ORG> bought <ORG>**Haier**</ORG>.

- Identify mentions in text and classify them into a predefined set of categories of interest:

- Person Names: **Mr John Minto**
- Organizations: **Fisher and Pykal, Haier**
- Locations: **Auckland**
- Date and time expressions: **semester 2**
- E-mail: **mkg@gmail.com**
- Web address: **www.elena.co.nz**
- Names of drugs: **paracetamol**
- Names of ships: **Queen Marry**
- Bibliographic references:
- ...

Knowledge NER vs. Learning NER - generic

Knowledge Engineering



- + very precise (hand-coded rules)
- + small amount of training data
- expensive development & test cycle
- domain dependent
- changes over time are hard to maintain

Learning Systems



- + higher recall
- + no need to develop grammars
- + developers do not need to be experts
- + annotations are cheap (er)
- require lots of training data

Rule Based NER (1)

- **Create regular expressions to extract:**
 - Telephone number
 - E-mail
 - Capitalized names

Rule Based NER (1)

- Regular expressions provide a flexible way to match strings of text, such as particular characters, words, or patterns of characters

Suppose you are looking for a word that:

1. starts with a capital letter “P”
2. is the first word on a line
3. the second letter is a lower case letter
4. is exactly three letters long
5. the third letter is a vowel

the regular expression would be “^P[a-z][aeiou]” where

^ - indicates the beginning of the string

[a-z] – any letter in range a to z

[aeiou] – any vowel

Perl RegEx

- `\w` (word char) any alpha-numeric
- `\d` (digit char) any digit
- `\s` (space char) any whitespace
- `.` (wildcard) anything
- `\b` word boundary
- `^` beginning of string
- `$` end of string
- `?` For 0 or 1 occurrences
- `+` for 1 or more occurrences
- specific range of number of occurrences: `{min,max}`.
 - `A{1,5}` One to five A's.
 - `A{5,}` Five or more A's
 - `A{5}` Exactly five A's

Rule Based NER (1)

■ Create regular expressions to extract:

- Telephone number
- E-mail
- Capitalized names

blocks of digits separated by hyphens

RegEx = (\d+\-)+\d+

Rule Based NER (1)

■ Create regular expressions to extract:

- Telephone number
- E-mail
- Capitalized names

blocks of digits separated by hyphens

RegEx = (\d+\-)+\d+

- matches valid phone numbers like 900-865-1125 and 725-1234
- incorrectly extracts social security numbers 123-45-6789
- fails to identify numbers like 800.865.1125 and (800)865-CARE

Improved RegEx = (\d{3}[-.\s()]{1,2}[\dA-Z]{4})

Rule Based NER (2)

■ Create rules to extract locations

- Capitalized word + {city, center, river} indicates location
 - Ex. *New Zealand*
 - Auckland city
- Capitalized word + {street, boulevard, avenue} indicates location
 - Ex. Queen Street

Rule Based NER (3)

■ Use context patterns

- [PERSON] earned [MONEY]
Eg. *Frank earned \$20*
- [PERSON] joined [ORGANIZATION]
Eg. *Sam joined IBM*
- [PERSON],[JOBTITLE]
Ex. *Mary, the teacher*

still not so simple:

- [PERSON|ORGANIZATION] fly to [LOCATION|PERSON|EVENT]
Eg. *Jerry flew to Japan*
Sarah flies to the party
Delta flies to Europe

Why simple things would not work?

- Capitalization is a strong indicator for capturing proper names, but it can be tricky:
 - first word of a sentence is capitalized
 - sometimes titles in web pages are all capitalized
 - nested named entities contain non-capital words
Auckland University of Technology is an Organization
 - all nouns in German are capitalized
 - Tweets/Micro-bloggs have “loose” capitalization

Why simple things would not work?

- No lexicon contains all existing proper names.
- New proper names constantly emerge
 - movie titles*
 - books*
 - singers*
 - restaurants*
 - etc.*

Why simple things would not work?

- The same entity can have multiple variants of the same proper name

Prime Minister
Labour Party Leader
Arden



Jacinda Arden

- Proper names are ambiguous

Jordan the *person* vs. Jordan the *location*

JFK the *person* vs. JFK the *airport*

May the *person* vs. May the *month*

Learning Systems

■ ***Supervised learning***

- labeled training examples
- methods: Hidden Markov Models, k-Nearest Neighbors, Decision Trees, AdaBoost, SVM, NN...
- example: NE recognition, POS tagging, Parsing

■ ***Unsupervised learning***

- labels must be automatically discovered
- method: clustering
- example: NE disambiguation, text classification

Learning System



■ ***Semi-supervised learning***

- small percentage of training examples are labeled, the rest is unlabeled
- methods: bootstrapping, active learning, co-training, self-training
- example: NE recognition, POS tagging, Parsing,
...

k Nearest Neighbor - Recap

- Learning is just storing the representations of the training examples.
- Testing instance x_p :
 - compute similarity between x_p and all training examples
 - take vote among x_p k nearest neighbours
 - assign x_p with the category of the most similar example in T

Distance measures

- Nearest neighbor method uses similarity (or distance) metric.
- Given two objects x and y both with n values

$$x = (x_1, x_2, \dots, x_n)$$

$$y = (y_1, y_2, \dots, y_n)$$

calculate the Euclidean distance as

$$d(x, y) = \sqrt{2} \sqrt{\sum_{i=1}^p |x_i - y_i|^2}$$

An Example

	isPerson Name	isCapitalized	isLiving	IsPolitician
John Key(JK)	1	1	1	1
National Party(NP)	0	1	0	0
judith collins(JC)	1	0	1	1
Phil Coye(PC)	1	1	1	1

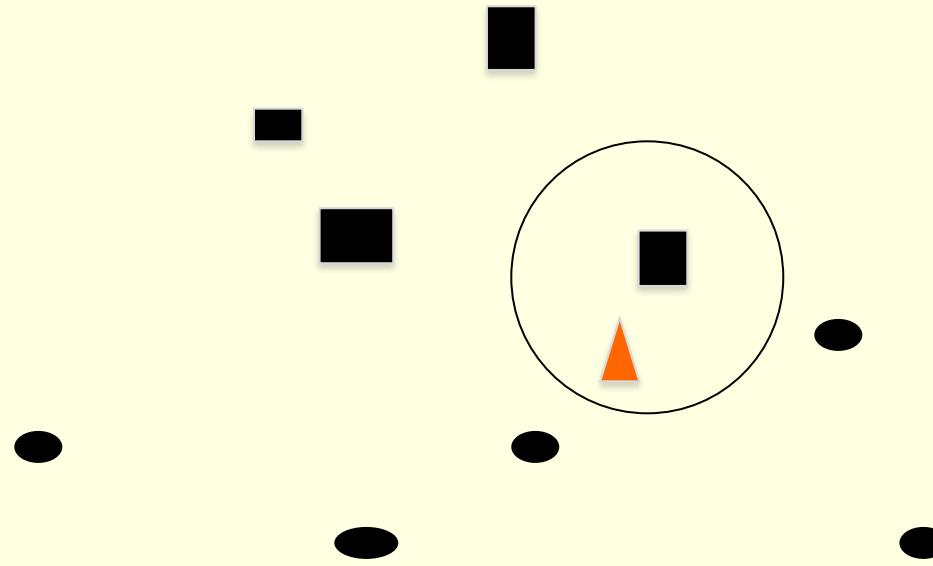
Euclidean distance:

$$d(JK, NP) = \sqrt{2} \sqrt{(1^2 + 0 + 1^2 + 1^2)} = 1.73$$

$$d(JK, JC) = \sqrt{2} \sqrt{(0 + 1^2 + 0 + 0)} = 1$$

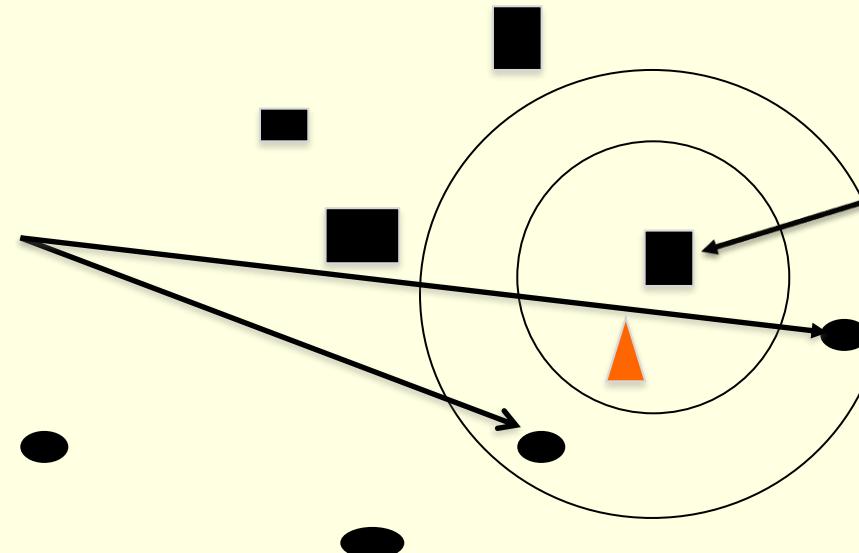
$$d(JK, PC) = \sqrt{2} \sqrt{(0 + 0 + 0 + 0)} = 0$$

1-Nearest Neighbor



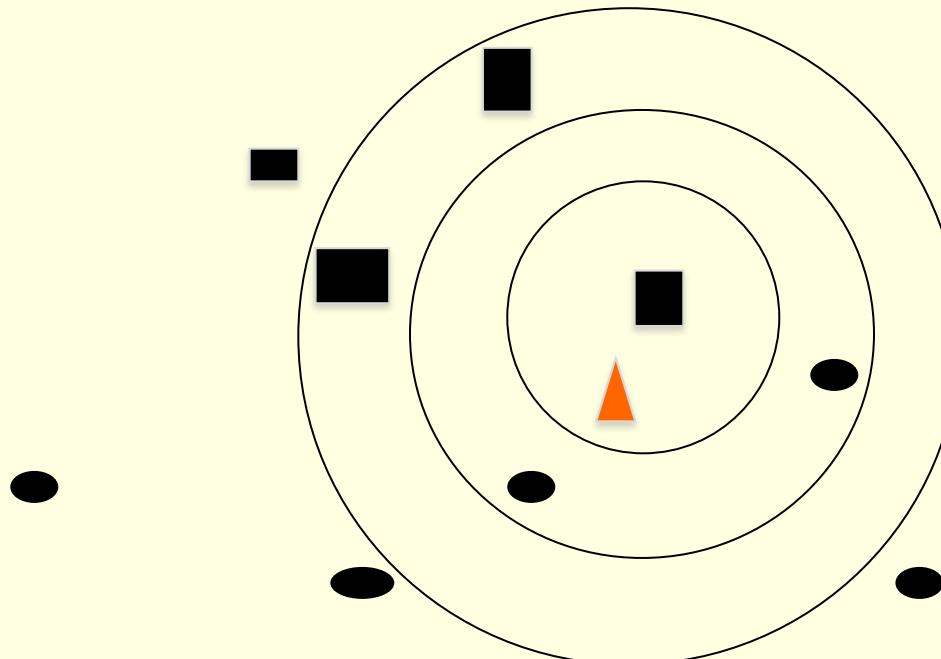
3-Nearest Neighbor

choose the category of the majority of the neighbors



choose the category of the closer neighbor
(can be erroneous due to noise)

5-Nearest Neighbor



the value of k is typically odd to avoid ties

k Nearest Neighbours

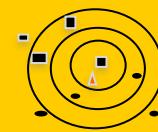
Pros

- + robust
- + simple
- + training is very fast (storing examples)



Cons

- depends on similarity measure & k-NNs
- easily fooled by irrelevant attributes
- computationally expensive



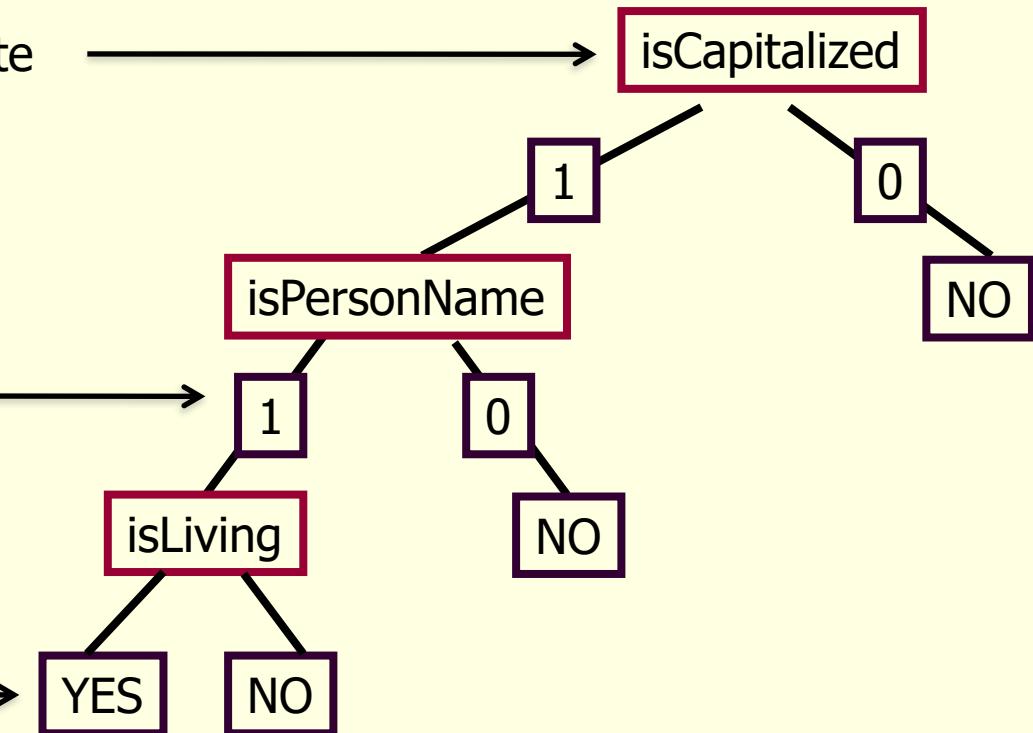
Decision Trees

- The classifier has a tree structure, where each node is either:
 - a leaf node which indicates the value of the target attribute (class) of examples
 - a decision node which specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test
- An instance x_p is classified by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance

An Example

	isPersonName	isCapitalized	isLiving	X is PersonName?
profession	0	0	0	NO
John Key	1	1	1	YES
National Party	0	1	0	NO
Judith Collins	1	1	0	NO

Each internal node tests an attribute



Each branch corresponds to an attribute value node

Each leaf node assigns a classification

Building Decision Trees

- Select which attribute to test at each node in the tree.
- The goal is to select the attribute that is most useful for classifying examples.
- Top-down, greedy search through the space of possible decision trees. It picks the best attribute and never looks back to reconsider earlier choices.

Decision Trees

Pros

- + generate understandable rules
- + provide a clear indication of which features are most important for classification

Cons

- error prone in multi-class classification and small number of training examples
- expensive to train due to pruning

Features for NE Detection (1)

Adam Smith works for ibm in London.

- **Contextual**
 - current word W_0
 - words around W_0 in $[-3, \dots, +3]$ window

Features for NE Detection (2)

Adam Smith works for IBM in London.

- **Ortographic**
 - *initial-caps*
 - *all-caps*

Features for NE Detection (3)

❑ Orthographic (binary and not mutually exclusive)

- | | | |
|-----------------------|-------------------------|-------------------------|
| ❑ <i>initial-caps</i> | <i>all-caps</i> | <i>all-digits</i> |
| ❑ <i>roman-number</i> | <i>contains-dots</i> | <i>contains-hyphen</i> |
| ❑ <i>acronym</i> | <i>lonely-initial</i> | <i>punctuation-mark</i> |
| ❑ <i>single-char</i> | <i>functional-word*</i> | <i>URL</i> |

❑ Word-Type Patterns:

- | | | |
|----------------------|-------------------------|--------------|
| ❑ <i>functional</i> | <i>lowercased</i> | <i>quote</i> |
| ❑ <i>capitalized</i> | <i>punctuation mark</i> | <i>other</i> |

❑ Left Predictions

- ❑ the tag predicted in the current classification for W-3, W-2, W-1

❑ Part-of-speech tag

❑ Functional Words

The more useful features you incorporate, the more powerful your learner gets.

Features for NE Classificaton (4)

- **Contextual**
 - current word W_0
 - words around W_0 in $[-3, \dots, +3]$ window
- **Part-of-speech tag** (when available)
- **Bag-of-Words**
 - words in $[-5, \dots, +5]$ window
- **Trigger words**
 - for person (*Mr, Miss, Dr, PhD*)
 - for location (*city, street*)
 - for organization (*Ltd., Co.*)
- **Gazetteers**
 - geographical
 - first name
 - surname
 - company names

Machine Learning NER

Adam_B Smith_I works_O for_O IBM_B ,_O London_B ._O

■ NED(Named Entity Detection)

- Identify named entities using BIO tags
- B beginning of an entity
- I continues the entity
- O word outside the entity

Machine Learning NER

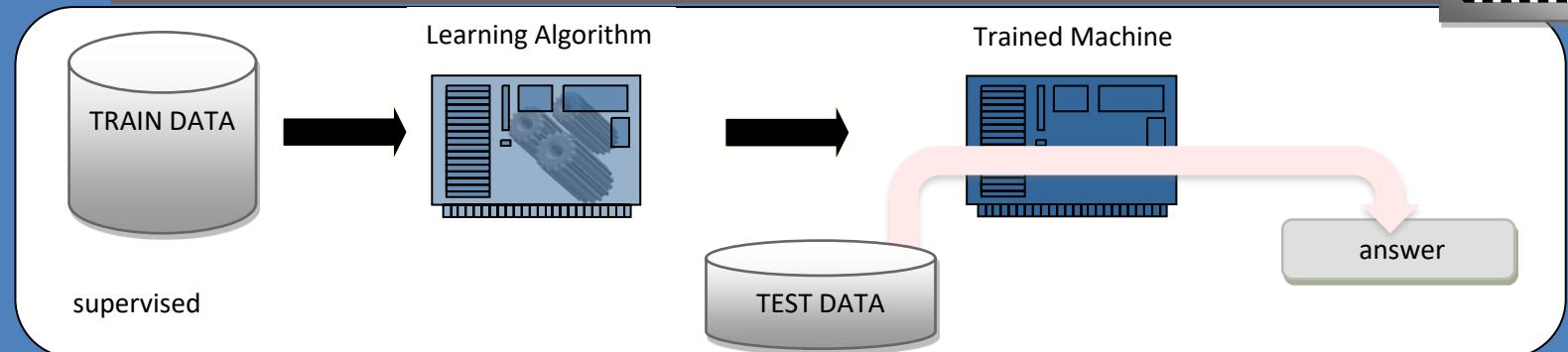
- Adam_B-PER Smith_I-PER works_O for_O IBM_B-ORG ,_O London_B-LOC ._O

- **NED:** Identify named entities using BIO tags
 - B beginning of an entity
 - I continues the entity
 - O word outside the entity
- **NEC:** Classify into a predefined set of categories
 - Person names
 - Organizations (companies, governmental organizations, etc.)
 - Locations (cities, countries, etc.)
 - Miscellaneous (movie titles, sport events, etc.)

United States presidential election of 2008, scheduled for Tuesday November 4, 2008, will be the 56th consecutive quadernnial United States presidential election and will select the President and the Vice President of the United States. The Republican Party has chosen John McCain, the senior United States Senator from Arizona as its nominee; the Democratic Party has chosen Barak Obama, the junior United States Senator from Illinois, as its nominee.

United_0.1.0_0.1.ed.ted.Un.Uni_1.0.1.1_1.null.null.null.States.presidential.election.1.1.1.0.0

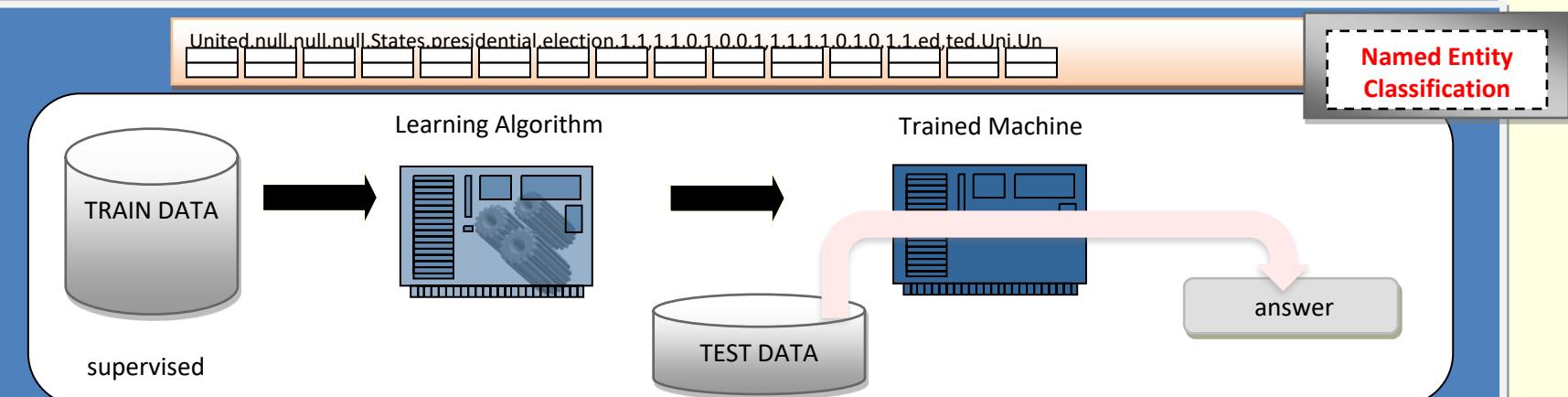
Named Entity Detection



United_B States_I presidential_O election_O of_O 2008_O ,_O scheduled_O for_O Tuesday_O November_O 4_O ,_O 2008_O ,_O will_O be_O the_O 56th_O consecutive_O quadernnial_O United_B States_I presidential_O election_O and_O will_O select_O the_O President_B and_O the_O Vice_B President_I of_I the_I United_I States_I. The_O Republican_B Party_I has_O chosen_O John_B McCain_I ,_O the_O senior_O United_B

United.null.null.null.States.presidential.election.1.1.1.0.1.0.0.1.1.1.1.1.0.1.0.1.1.ed.ted.Uni.Un

Named Entity Classification



United_B-LOC States_I-LOC presidential_O election_O of_O 2008_O ,_O scheduled_O for_O Tuesday_O November_O 4_O ,_O 2008_O ,_O will_O be_O the_O 56th_O consecutive_O quadernnial_O United_B-LOC States_I-LOC presidential_O election_O and_O will_O select_O the_O President_B-PER and_O the_O Vice_B-PER President_I-PER of_I-PER the_I-PER United_I-PER States_I-PER. The_O Republican_B-ORG Party_I-ORG has_O chosen_O John_B-PER McCain_I-PER ,_O the_O senior_O United_B-PER States_I-PER Senator_I-PER from_O Arizona_B-LOC as_O its_O

Results for NE Detection

Carreras et al., 2002	Precision	Recall	F-score
BIO dev.	92.45	90.88	91.66

CoNLL-2002 Spanish Evaluation Data		
Data sets	#tokens	#NEs
Train	264,715	18,794
Validation	52,923	4,351
Test	51,533	3,558

Evaluation Measures
Precision
Recall

Demo Code

deptokens.csv

sampleText2.txt

Lecture 01 Introduction

1. Text Mining is **applied ,objective question**, uses algorithms and results developed by NLP
2. NLP is **fundamentals**, deals with **essential text processing algorithms**
3. Why Text Mining ?
 - a. vast amount of **natural language**,
 - b. vast amount of info in DB,
 - c. communicate with machine with natural language
4. **NLP** progressed **immensely** (wide) , became **generative**
 - a. NLP disciplines
 - i. linguistics
 - ii. psycholinguistics (心理语言学)
 - iii. Computational linguistics
 - iv. Artificial intelligence
 - v. NL Engineering (chatGPT)
 - vi. Text Mining
 - b. NLP Application
 - i. Text-based applications(**classify** docs,**extraction**,
search,comprehension,translate,summarization, knowledge management
)
 - ii. Dialogue-based applications(communication, QA, Chatbots)
 - iii. Speech-processing(text2speech, speech2text)
 - c. Hard Problem
 - i. **context** based
 - ii. **multi-disciplinary**(多学科) in nature
5. (NLP) Basic level of language processing
 - a. **morphological** knowledge(形态/词法 e.g friend, friendly, unfriendly,friendliness)
 - i. inflection Vs **derivational**
 - ii. **stem Vs Lemmatization**
 - b. **Syntactic** (句法, form correct sentences, the role of each word) **Parsing**, **chunk**
 - c. **Semantic**(语义) – embedding vector
 - d. **phonetic**(语音 e.g. lead is a metal; he leads the team)

"Phonetic" 通常涉及到语音 (声音) 方面的内容, 因此更倾向于与 speech processing (语音处理) 相关联, 而不是与 text-based (基于文本的) 处理相关联。
6. (NLP) **complicated** level of language processing
 - a. **Pragmatic** knowledge (实用知识, eg. 下雨了, 撑伞; 怕狗, 撑伞。. context)
 - b. **Discourse** knowledge (语境知识, eg. John and Tom like apples and **they eat them** often. 前面句子影响)
7. Ambiguities
 - a. **Lexical**(in 不同的语境中)

- i. 飞 驾驶 Rice **files** like sand; Flying planes is dangerous;
- ii. 河岸 银行 The river's **bank** was breached by the flood" and "My loan got approved by the bank")

b. **Semantic (polysemous 多义词 words, 杀 : A killed B; A killed the project)**

8. discourse(context)

- a. connected with (**Co-reference 共指**, ordering, segmentation)
- b. inference (推理) utterance (表达);
- c. discourse model is a computational framework

i. lexical connectivity patterns

1. **cohesion 连贯性**

a. **lexical connectivity via : (repetitions\ synonyms\ co-reference 共指)**

1. **Repetitions (重复):**

重复是指在文本中多次出现相同的词汇或短语。这种重复可以在同一句子、同一段落或整个文本中发生。例如: "The cat jumped onto the table. The cat then knocked over a vase."在这个例子中, "cat" 在第一句和第二句中都出现了, 构成了一个重复的模式。

2. **Synonyms (同义词):**

同义词是指具有相同或相似含义的词汇。在文本中, 作者可能会使用不同的词汇来表达相同的概念。例如: "The dog barked loudly."

"The canine howled loudly."在这个例子中, "dog" 和 "canine" 是同义词, 它们都指代狗, 而 "barked" 和 "howled" 也是同义词, 都表示动物发出的声音。

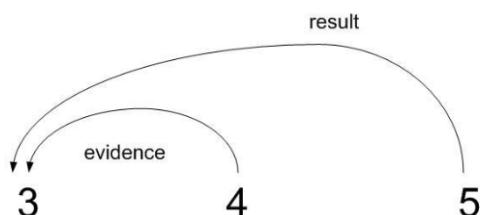
3. **Co-reference (共指):**

共指是指在文本中不同位置出现的词汇指代相同的实体或概念。例如: "John loves his car. He takes good care of it."在这个例子中, "John" 和 "He" 共同指代了同一个人, 即 John, 而 "his" 和 "it" 共同指代了同一辆车。

ii. rhetorica(修辞)l relation

1. **Rhetorical Structure Theory(修辞)**

- a. **Fact:** 3. there was a huge storm
- b. **Evidence:** 4. it poured for 2 hours
- c. **Result:** 5. I got soaked



Rhetorical Structure Theory (RST) 是一种用于分析和描述文本结构的理论框架, 重点关注文本中不同句子之间的关系和组织方式。RST提供了一种形式化的方法来理解文本的组织结构, 其中包括识别主要观点、支持论据、因果关系等。

iii. notion(概念) of grammar

利用语法概念来表达话语 (Discourse)，主要是通过句法、形态和语义来组织和解释文本中的句子，从而形成连贯的话语。以下是一些在中文中利用语法概念来表达话语的方法：

1. 句法结构的使用：

- 利用不同的句法结构来组织句子，使话语流畅清晰。例如，使用主谓宾结构、并列结构或从句等。
- 合理安排句子的顺序，使话语逻辑清晰，内容连贯。可以根据具体情况选择先进后出、先前后出等顺序。

2. 形态的运用：

- 使用正确的词语形态来表达句子的含义，确保用词准确、恰当。例如，使用不同的时态、语态、人称等形态来表达不同的语境和态度。
- 根据句子需要，灵活运用形态变化，使句子更加丰富多样。

3. 语义的考量：

- 注重词语的准确含义和语义关系，避免歧义和误解。确保词语的选择符合语境和话语目的。
- 关注句子之间的逻辑关系和连贯性，确保话语内容有机串联，理解上下文的语义脉络。

通过以上方式，**可以利用语法概念来表达话语，使话语内容更加清晰明了、连贯流畅。**

iv. examination of intentions and relations

这句话探讨了意图之间的关系以及它们作为话语结构基础的重要性。在实际的语言交流中，人们的意图对话语的组织和结构起着关键作用。以下是一个例子来说明这个概念：

假设有一段对话如下：

- A: 你觉得我们应该怎么庆祝我的生日?
B: 我认为我们可以去看一场电影，然后一起吃晚饭。
A: 那听起来不错。我也想试试新开的那家餐厅。

在这段对话中，我们可以看到意图和它们之间的关系如何构成了话语结构的基础：

1. Examination of intentions (意图的审查)：

- A 的意图是询问如何庆祝他的生日。
- B 的意图是提出建议，建议去看电影并一起吃晚饭。
- A 回应了 B 的建议，表示认同，并表达了他也有兴趣尝试新餐厅的意图。

2. Relations among intentions (意图之间的关系)：

- B 的建议与 A 的询问形成了回应关系，表达了合作和共享的意图。
- A 的回应与 B 的建议形成了接受关系，表达了认同和接受的意图。

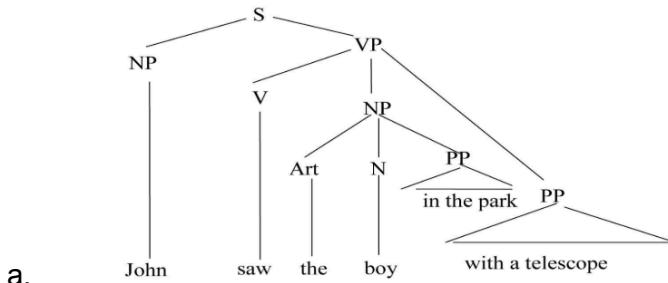
3. Foundation of discourse structure (话语结构的基础)：

- 这些意图之间的关系构成了对话的逻辑结构，形成了对话的发展和连贯性。由于每个发言都是基于前一个发言的意图而进行的，因此意图之间的关系成为话语结构的基础。

9. POS (part of speech tagged 词性) (JJ: adjective; NN: noun; RB: adverb; VB: verb;)

10. Syntactic structure of a sentence(句法) – Parsing

Syntactic Parse



- a. John saw the boy in the park with a telescope
- b. Parsing is higher level than POS tagging
- c. Parsing is based on Pos tagging
- d. **Syntactic Bracketing** (Syntactic tags句法标签)
 - i. s: 简单陈述句;
 - ii. NP: noun phrase
 - iii. VP: Verb phrase
 - iv. PP: Prepositional phrase 介词.

Lecture 02 Token & Ngrams

11. Basic units

a.

1	Alphabet	Symbol
2	Word(isn't)	Token ('is', 'n't')
3	Phrase	Utterance
4	document	Discourse
5	Collection of Discourse	corpus 语料库
6	collection of corpus	corpora 语料库

12. word

- a. different with token: “,-”
 - i. e,g. pre-processing, couldn't pos_tag("couldn't") =[("could", "MD"), ("n't", "RB")]
 - ii. pre-processing: ('pre-processing', 'VBG')
- b. chunking(New Zealand)

13. Nonus (plural 复数 possessive 所有格)

14. Verbs(**Tense** 时态)
15. **Regulars** (walk walks walking walked walked)
16. **Irregulars** (eat eats eating rate eaten)
17. Morphology **morphological**, (friend friends friendly) 词形学
- in all languages
 - morphemes 语素 分为:
 - stems**
 - affixes**
 - 另外一种分类角度, 更详细:
 - inflectional (taught you, class don't change)**
 - skip, skipping, skipped
 - irregular is complex
 - derivational (no taught you, class may changed)**
 - quasi-systematicity
 - irregular meaning change
 - changes of word class
 - verb Adj ← → Nouns**
 - 词性\语义
 - e.g. **deconstruction, unfriendly**
18. word prediction / guess
- N-gram (token sequences of length N)
 - the next word 下一个词
 - the entire sequence of words 词序
 - clusters of words** are **extracted** and these clusters are **counted** and used in the model.
19. Token Applications 作为核心core应用于:
- automatic speech recognition (ASR) 语音识别
 - Handwriting and character recognition 手写识别
 - spelling correction 拼写检查
 - machine translation 机器翻译
 - and more
20. counting
- counting strategy depend on application
 - I do uh mainly business data processing**
 - types (as opposed to tokens)

“ They picnicked by the pool, then lay back on the grass and looked at the stars.”

18 tokens (again counting punctuation), 16 Types

- **Tokens**: Tokens 是指文本中的基本单位，通常由空格或标点符号分隔开来，可以是单词、词组或标点符号等。在给定的句子中，包括单词、标点符号等都被计算在内，总数为 18 个。

- **Types**: Types 是指文本中不同的词或短语的种类数。在给定的句子中，如果某个词或短语多次出现，只计算为一个类型。在这个例子中，“the”这个词被使用了 3 次，但只计算为一个类型，因此句子中的唯一类型数为 16 个。

c. **Wordforms (morphological)**

- i. **lemma(valid dictionary word)** 字典有效词
- ii. **stem(may not valid)**

d. corpora

- i. title

21. Language Modeling

- a. assess the **conditional probability** (评估条件概率)

- i. $P(W_n | W_1, W_2 \dots W_{n-1})$

- ii. "I was sitting on the park bench drinking water from a

- iii. $P(\text{bottle} | \text{water from a})$

- iv. $P(\text{glass} | \text{park bench drinking water from a})$

$P(\text{the} | \text{its water is so transparent that}) =$

$P(\text{its water is so transparent that the})$ ←
 $P(\text{its water is so transparent that})$ ←

- b. reduce dimension strategy

- i. **stop words**
- ii. **stemming, Lemma (singular forms of words)**
- iii. **feature selection**
- iv. **latent semantic analysis (潜在语义分析)**

22. Chunking–NER(named entity recognition)

- a. the process of **identifying noun** phrases

23.

```
import nltk
nltk.download('punkt')
text = nltk.word_tokenize(text1)
#Remove numbers 去数字
result = re.sub(r'\d+', '', text5)
#Remove punctuation- member of [!"#$%&' ()*+, -./:; <=>?@[]^_`{|}~]: 去标点符号
```

```
text6.translate(str.maketrans(' ', ' ', string.punctuation))
#remove white spaces 去空白
text7 = text7.strip()

#remove stop words 去stop words
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
stop_words_NLTK = set(stopwords.words('english'))
tokens = tokenize(text1)
result = [i for i in tokens if not i in stop_words_NLTK]
```

Lecture 03 POS Tagging

24. word Class

- a. N: noun; V:verb; ADJ; ADV;P ; PRO;DET :determiner(the a an that those)
- b. Applications for POS Tagging
 - i. Speech Synthesis pronunciation (lead -verb versus noun) 语音合成
 - ii. word **prediction** in speech recognition (e.g. her + nouns) 语音识别预判
 - iii. machine translation 机器翻译
 - iv. information translation
 - 1. extract all clauses
 - 2. dominant concepts
 - 3. event modelling

- c. closed vs. Open class words
 - i. closed: relatively fixed set (pre, auxiliaries, pronouns, **play a role in grammar**)
 - ii. open: **productive** (noun, verb, adj, adv)
 - 1. noun
 - a. **proper nouns** (专有名词AUT, New Zealand, Ack, CHCH)
 - b. common nouns
 - i. count vs mass
 - 2. adj
 - a. order restrictions (old blue book)
 - 3. adv
 - a. modify things 修改事物(verb,adj, adv))
 - 4. verb
 - a. **morphological affixes**(eat/eats/eaten)
 - b. Subclass(eat: VB; eats VBZ; eaten VBN)

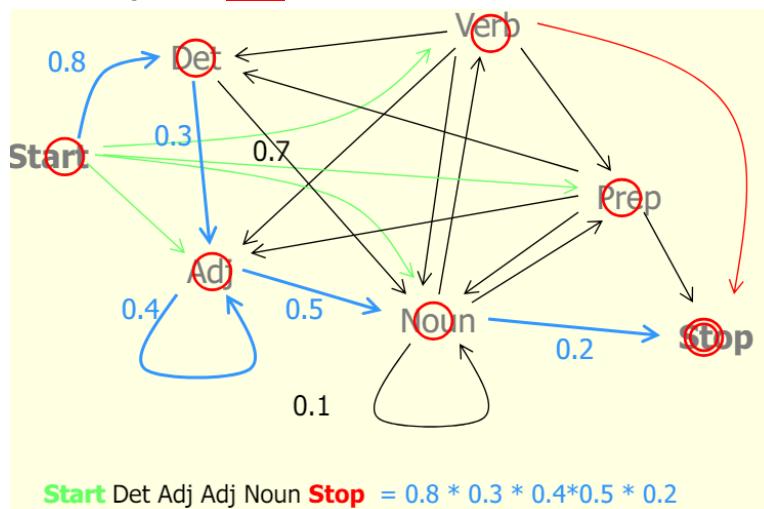
25. Tag Ambiguity

- a. POS tag for particular instance of a word(eg. back : JJ,NN,RB,VB)
- b. whole sentence with POS is Hard,
 - i. Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N

26. Probabilistic Model of POS tagging

- a. **find the most probable corresponding sequence**
理论基础
- b. **probable:** most frequently observed in some manually tagged corpus of words
- c. Penn Treebank 2 (a common training corpus, 1 million words)

27. Markov Model(bigrams)



马尔可夫模型（Markov Model）是一种随机过程模型，描述了一系列状态之间通过转移概率相互转换的过程。马尔可夫模型的一个重要变体是二元模型（Bigrams），它特别适用于序列数据的建模。

在二元模型中，我们考虑的是序列中每个项目的转移概率，而不是项目本身的概率。具体来说，二元模型假设序列中的每个项目仅与其前一个项目相关。这意味着在建模过程中，我们仅考虑当前项目与前一个项目之间的转移概率，而忽略了其他项之间的关系。

二元模型通常用于自然语言处理领域，特别是在语言建模和文本生成中。在文本中，一个大二元模型的示例是 Bigram Language Model，它用于估计一个单词出现在另一个单词之后的概率。例如，在一个英语文本中，“the cat”出现的概率可以由Bigram Language Model估计为 $P("cat" | "the")$ 。

二元模型还被广泛应用于其他序列数据的建模和预测任务，如音频信号处理、生物信息学和金融数据分析等领域。

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

a.

28.

Precision= 点/预测

recall = 点/实际

F1 = $2*(p*r)/(p+r)$

Accuracy = 对/全

$$Precision_0 = \frac{\text{模型预测为类别0且实际为类别0的样本数}}{\text{模型预测为类别0的样本数}}$$

$$Recall_0 = \frac{\text{模型预测为类别0且实际为类别0的样本数}}{\text{实际为类别0的样本数}}$$

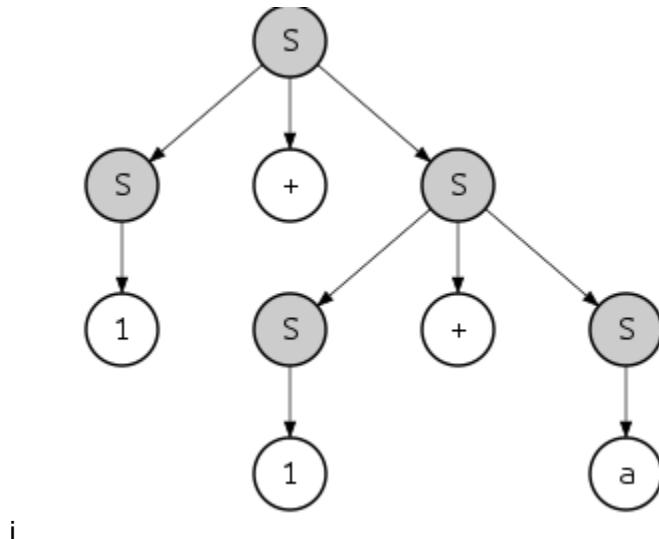
$$\text{Precision} = (\text{预测0 且 实际0}) / \text{预测0}$$

29. Evaluating Tagging Approach

- a. Possible Gold Standards 黄金标准——上限:
 - i. **Annotated** naturally occurring corpus
 - ii. **human task performance** 96-97%
 - 1. **kappa statistic**: avg pairwise agreement,
 - a. Kappa统计量通常用于评估自动标注系统的性能，或者用于评估不同人类标注者之间的一致性。
 - b. between **human evaluators**.
 - c. (-1:完全不一致;0:与随机选择没有差别;1:完全一致)
 - iii. **hard to obtain**: sometimes humans don't agree
 - b. Baseline
 - i. for tagging, most common tag for each **word** 91%
 - ii. how much improvement

30. Parsing (token, tagging, parsing)

- a. **identify chunks / units in a clause**
- b. functions attached together (句子中的不同语言单位的功能通常通过从属关系相互连接在一起。例如，一个从属从句的主语通常与从句的谓语相连接，形成一个完整的语法结构。)
- c. after POS tagging
- d. CFG (Context-Free Grammar) parsing is recursive, difficult. **句子结构分析**



1. CFG Rules:

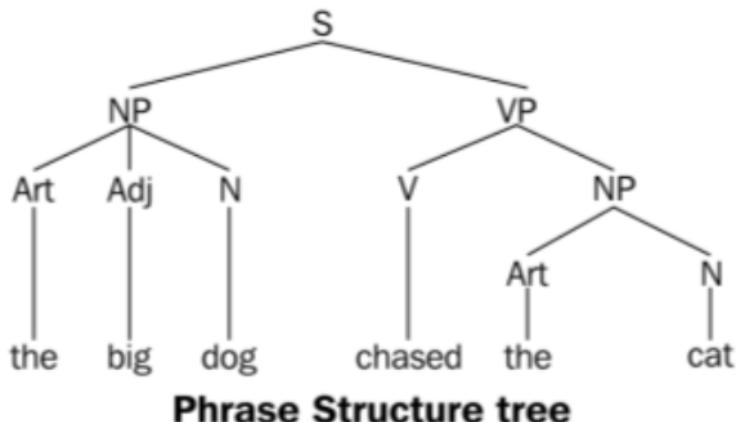
- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ Adj^*\ N$
- $VP \rightarrow V\ NP$
- $Det \rightarrow "the" \mid "a"$
- $Adj \rightarrow "red" \mid "ripe" \mid ...$
- $N \rightarrow "apple" \mid "banana" \mid "orange" \mid ...$

ii.

e. 在实际中使用 Treebank grammar / Dependency parse

i. penn TreeBank(widely used)

1. implicitly define a grammar
2. sub-trees with local rules (这些本地规则描述了子树中各个节点之间的语法关系, 比如节点的标签、节点之间的父子关系等。)
3. **not complete**, decent coverage, **not cover all possible types of sentences**
 - a. 语料库的大小对于构建具有良好覆盖率的语法模型的重要性, 足够大的语料库可以提供足够多样的句子结构, 从而构建出一个较为全面的语法模型
4. very flat to **avoid recursion**, **simpler** than CFG



5.

ii. Dependency Relations

1. Argument Dependency

- a. nsubj (normal **subject**), csubj (clausal subject), dobj (direct object)

2. Modifier Dependency

- a. tmod (temporal 时间 **modifier**), appos(appositional 同位语 modifier)

3.

iii. dependency Parsing

1. each linguistic word is connected directly。一个单词是另一个单词的中心或附属。
2. parse tree captures the unidirectional (单向) relationship between words and phrases
3. binary **relations** between **tokens** (二元关系)



1. 每个语言单词通过有向链接相连接：在依存句法分析中，句子中的每个单词都被视为一个节点，并且这些节点之间通过有向链接相连接。这些链接表示了单词之间的依存关系，即一个单词是另一个单词的中心或附属。
2. 句法树捕捉词语和短语之间的（单向）关系：依存句法分析通过构建依存树（Dependency Tree）来捕捉句子中词语和短语之间的依存关系。在依存树中，每个单词作为一个节点，而每个有向链接则表示一个依存关系。这些依存关系通常是从一个中心词指向其依赖词，因此是单向的。

Lecture 04 Vector Space Model

1. Vector Space Model
 - a. representation of doc or discourse with **data structure**
 - b. can do **vector** and **math operate**
 - c. vector of words
 - d. document can be **ranked**
 - e. **Dimension reduction strategies**
 - i. **stop words**
 - ii. **stemming, Lemma (singular forms of words)**
 - iii. **feature selection**
 - iv. **latent semantic analysis (潜在语义分析)**
2. Retrieve relevant documents
 - a. **boolean model**:
 - i. using AND and OR matching the query,
 - ii. get **Yes or No**

- iii. empty or huge
- b. Ranking(**more relevant** is presented first in google search)
 - i. a way of assigning a ranking **score**
 - ii. **frequency** of the terms
 - 1. One Hot Representation (only presence or absence of tokens) 表示单词是否存在[1,0,0,0,0,0]
 - iii. **Ignore the order :**
 - 1. **BOW** (bag of words model)
 - a. surprisingly effective
 - b. Can capture plural and singular words.
 - c. Can capture stop words.
 - 2. **word2vec**
 - a. $v(d)_1 = (w(t_1, d), w(t_2, d), \dots, w(t_n, d))$
 - b. $v(d)_2 = (w(t_1, d), w(t_2, d), \dots, w(t_n, d))$
 - c. a doc include the weight of each dict term
 - d. **Spare matrix** for most docs 现在是稀疏矩阵, 因为绝大部分为0
 - e.
 - iv. ignore (stop words, inflections- stemming, plurals and singular)
- 3. Term Frequency – Tf
 - a. $TF(t_1, d_1)$, the times of term1 in document1
 - b. need **normalize** frequency: frequency/number of words, 频率/字数, 因为文章有长有短
- 4. document frequency–df
 - a. **Frequent terms are less informative** than rare terms
 - b. So, Frequent terms should be **lower weights** than rare terms.
 - c. use df to capture this.
- 5. IDF - Inverse Document Frequency**
 - a. $df(t)$, the number of documents that contain t.
 - b. $df(t) \leq N$ (the num of docs)
 - c. $\log(N/df(t))$, use log to **dampen** the effect. **log可以用ln替换**
- 6. TF-IDF
 - a. TF: term Frequency ; $TF(t) = t$ 出现次数/ 术语总数
 - b. $IDF(t) = \log \frac{N}{df(t)}$ 文件总数/包t 的文档数
 - c. $TF-IDF(t, d) = \log(1 + TF(t, d)) * \log(N/df(t))$
 - d. $TF-IDF(t, d) = TF(t, d) * \log(N/DF(t))$ - alternative, 2个公式都可以, **log可以用ln替换**
 - e. Increases with the number of occurrences within a document (**文档中越多越好**)
 - f. Increases with the rarity of the term in the collection/corpus(**语料库中越少越好**)
- 7. Similarity computations
 - a. **Euclidean Distance** (Default measure metric)
 - i. $T = \{t_1, t_2, \dots, t_m\}$; $W(t, a) = \sqrt{\sum_{i=1}^m (t_i - a_i)^2}$

$$D_E(\vec{t}_a, \vec{t}_b) = \left(\sum_{t=1}^m |w_{t,a} - w_{t,b}|^2 \right)^{1/2}$$

ii.

$$\begin{aligned} & \sqrt{(27-4)^2 + (3-33)^2 + (0-33)^2 + (14-0)^2} \\ &= \sqrt{(23)^2 + (-30)^2 + (-33)^2 + (14)^2} \\ &= \sqrt{529 + 900 + 1089 + 196} \\ &= \sqrt{2714} \\ &\downarrow \\ &\approx 52.07 \end{aligned}$$

iii.

- b. **Cosine** Distance
 - c. **Jaccard** coefficient
 - i. $|\text{Intersection } (A,B)| / |\text{Union } (A,B)|$
 - ii. **0<=x<=1; 0: different; 1: same.**
 - d. **Edit** distance (Levenshtein Distance) 编辑几次
 - i. count of edits (insertion, deletion, substitution) to another sentence
 - e. **Hamming** Distance
 - i. same as Edit (每位依次对比)
 - f. **Pearson** Correlation coefficient
 - i. same as Data Mining. `np.corrcoef`
 - ii. 1: 完全的正相关; -1: 完全的负相关, 0: 没有线性相关关系
-

lecture 05 Word Embedding

- 8. Language Models
 - a. **count based** methods
 - i. word co-occurs with its neighbor
 - ii. BOW, TF, TFIDF
 - iii. **Sparse** vector
 - b. **predictive** models
 - i. predict a word from neighbors from learned small **dense** embedding vectors (原理)
 - ii. **word2vec** is efficient predictive **model** for learning word embeddings 基础
 - iii. two predictive model : **CBOW Skip-gram** (实现)

1. 组织(input 和 label), 交给word2vec训练, 得出 (input,w1, b1))

word Embedding > word2Vec > CBOW + Skip-gram

You

word Embedding 包括 word2Vec , 然后word2vec 使用了 CBOW 、 Skip-gram两个方法?

ChatGPT

是的，您理解得很正确。

Word Embedding 是一种将单词映射到低维向量空间的技术，以便在计算机中处理文本数据。

Word2Vec 是其中一种流行的 Word Embedding 方法，它通过学习单词的分布式表示来实现。而 Word2Vec 模型可以使用两种不同的方法进行训练：CBOW (Continuous Bag of Words) 和 Skip-gram。

CBOW (Continuous Bag of Words) 是一种用于训练 Word2Vec 模型的方法之一。它通过预测当前单词的上下文来学习单词的分布式表示。

Skip-gram 是另一种用于训练 Word2Vec 模型的方法。与 CBOW 相反，Skip-gram 模型的输入是当前单词的 one-hot 编码，输出是上下文单词的 one-hot 编码。

因此，Word2Vec 是 Word Embedding 的一种具体实现，而 Word2Vec 模型又可以使用 CBOW 或 Skip-gram 方法进行训练。

1. Word2Vec: Word2Vec 是一种用于学习单词的分布式表示 (word embeddings) 的技术。它可以将单词映射到低维空间中的向量，使得具有相似语义的单词在向量空间中距离较近。Word2Vec 模型可以通过两种不同的方法来训练：CBOW 和 Skip-gram。
2. CBOW (Continuous Bag of Words): CBOW 是一种用于训练 Word2Vec 模型的方法之一。它通过预测当前单词的上下文来学习单词的分布式表示。CBOW 模型的输入是上下文单词的 one-hot 编码，输出是当前单词的 one-hot 编码。CBOW 模型的目标是最大化给定上下文的条件下当前单词的概率。
3. Skip-gram: Skip-gram 是另一种用于训练 Word2Vec 模型的方法。与 CBOW 相反，Skip-gram 模型的输入是当前单词的 one-hot 编码，输出是上下文单词的 one-hot 编码。Skip-gram 模型的目标是最大化给定当前单词的条件下上下文单词的概率。
4. Word Embedding: Word Embedding 是一种将单词映射到低维向量空间的技术，以便在计算机中处理文本数据。Word2Vec、GloVe 和 FastText 等模型都是用于学习单词分布式表示的流行方法。
5. TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF 是一种用于表示文本中单词重要性的技术。它通过计算单词在文档中的频率和在整个语料库中的逆文档频率来衡量单词的重要性。TF-IDF 被广泛用于文本挖掘任务中，如文档分类、信息检索等。与 Word2Vec 不同，TF-IDF 不会学习单词的分布式表示，而是根据单词在文档集合中的出现频率和稀有程度进行加权。

9. Meaning representation

- a. Denotational . 通过 **符号或字母表** 的集合来表达意义
- b. Distributional. 意义由一个词在其周围上下文中的其他词 (**上下文**) 来决定。(目前研究内容)

10. Word Embedding

- a. used to **Rank doc, compute the difference**
- b. n-dimensional space
- c. **semantics** between words
 - i. semantically similar → nearby points
 - ii. distributional hypothesis: **words appear in the same context share semantic**
- d. **CBOW (predict target word)**
- e. **Skip-gram (predict context)**
 - i. window
 - ii. large corpus
 - iii. models semantically, related words
- f. **Train a NN on data to use the weights vector as the loop up**
 - i. **Unsupervised learning**
 - ii. NN for **word vector** representation
 - iii. **doesn't use NN in a conventional sense**
 - iv. use the **weights** as the **embedding matrix (to look up)**
 - v. 这句话指的是在训练一个神经网络，但是这个神经网络并不是用来解决传统的问题，比如图像分类或者预测房价。相反，它被用来学习词

语的向量表示，也就是把每个词都映射到一个多维空间中的点。这个神经网络会学习如何把每个词变成一个向量，让这些向量在空间中的位置反映词语之间的语义关系。

- vi. 当说到“use the weights as the embedding matrix (to look up)”时，意思是我们会用神经网络学到的参数（也就是权重）来构建一个嵌入矩阵。这个矩阵实际上就是一个词典，里面包含了每个词对应的向量表示。在训练完成后，我们可以通过查找这个矩阵，以获取每个词对应的向量表示。这样，我们就可以用这些向量来表示文本中的词语，进而用于各种自然语言处理任务，比如文本分类、情感分析等。

vii.

11. Glove

- Global Vector for word representation
- can download the **embedding to CNN models**. instead of training your own

12. dot product (点积，计算相似性)

- plant [p1,p2,p3]
- tree [t1,t2,t3]
- $p1*t1 + p2*t2 + p3*t3$, 结果越大，越相似，2个向量在方向上越相似。
- 点积（也称为内积或者点积运算）是一种用于衡量两个向量之间相似度的数学运算。在二维或三维空间中，点积的计算方法是将两个向量对应位置的元素相乘，然后将相乘的结果相加起来。在这个场景中，我们有两个向量，一个代表植物（plant）的特征，另一个代表树（tree）的特征。比如， $p1$ 、 $p2$ 、 $p3$ 分别代表了植物特征向量中的三个分量，而 $t1$ 、 $t2$ 、 $t3$ 则代表了树特征向量中的三个分量。那么，计算这两个向量的点积就是将对应位置的分量相乘，然后将相乘的结果相加起来，即 $p1*t1 + p2*t2 + p3*t3$ 。
- 点积的结果越大，表示两个向量在方向上越相似，即它们的方向更加一致。如果两个向量的方向完全相同，它们的点积结果将达到最大值。因此，在这个场景中，如果植物向量和树向量的点积结果越大，就表示这两个向量所代表的植物和树在特征上越相似。

13. 总结：

Word embedding 和 CBOW (Continuous Bag of Words) 以及 Skip-gram 是自然语言处理 (NLP) 领域中常用的技术，它们之间有着密切的关系。

CBOW 和 Skip-gram 都是用于训练词向量的神经网络模型，它们都属于词嵌入 (word embedding) 的一种实现方式。

- CBOW (Continuous Bag of Words) : CBOW 模型的目标是根据上下文单词来预测当前单词。具体来说，给定一个句子中的上下文单词（比如当前单词前后几个单词），CBOW 模型会尝试预测当前单词。这个模型的输入是上下文单词的词向量的平均值，输出是当前单词的词向量。

- Skip-gram: Skip-gram 模型的目标与 CBOW 相反, 它是根据当前单词来预测上下文单词。具体来说, 给定一个句子中的当前单词, Skip-gram 模型会尝试预测其周围的上下文单词。这个模型的输入是当前单词的词向量, 输出是上下文单词的词向量。

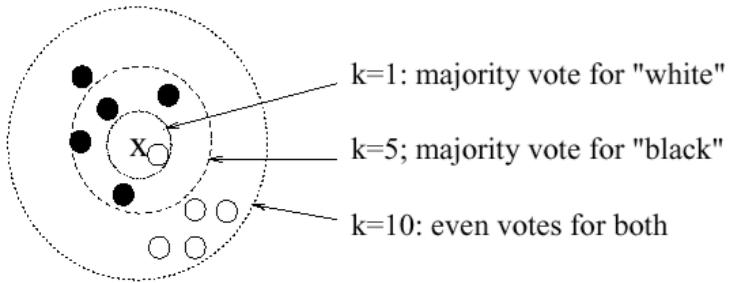
Word embedding 的目的是将单词表示为连续向量空间中的点, 以便能够更好地捕捉单词之间的语义关系。CBOW 和 Skip-gram 通过训练神经网络模型来学习这些词向量, 使得单词的向量表示能够在语义上更加丰富和准确。研究的目的在于通过这些词向量来提高自然语言处理任务的性能, 比如文本分类、情感分析、命名实体识别等。

Lecture 06 Text Categorization

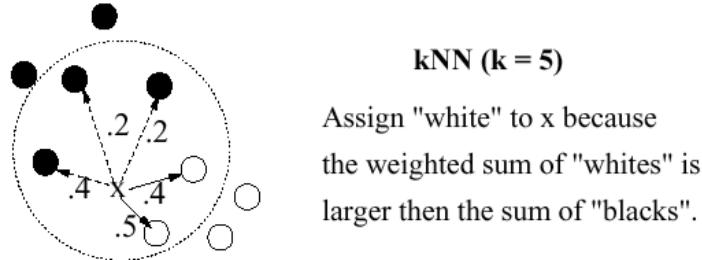
14. which models & algorithms work well for classification
 - a. first category, then similarity measure.(先分类再比较相似度)
15. Multi category VS binary category
 - a. MC can be done using iterative binary category
 - b. Text can be classified into MC **in a single step**
 - c. MC can be done using a algorithm capable of doing only binary classification
 - d. 多分类 can use 二元类的算法
16. Text Categorization Example
 - a. assign labels
 - b. category **taxonomies**(分类学) in google
 - c. web pages, Journal articles, patents archived, patient records categories, Email filter, News events filter, Blogs filter

17. Approaches

- a. KNN(k-Nearest Neighbors)
 - i. 用于分类和回归任务,
 - ii. 返回 k 个最近邻样本的平均值或加权平均值作为预测结果
 - iii. 不需要训练, 但是在预测时需要对所有训练样本进行搜索, 因此预测速度较慢。此外, KNN 对于特征空间的维度较敏感, 当特征空间的维度较高时, 其性能可能下降
 - iv. 而 K-Means 用于聚类任务, 它将样本划分为 K 个不同的簇
 - v. Majority voting schema



vi. Weighed-sum voting schema



vii. Knn for text category , steps.

1. represent doc as **points(vectors)**
2. choose **similarity measure** for pairwise documents, **any measure** can be used.
 - a. **cosine, Euclidean, Jaccard, kernel, kullback-leibler**
3. choose parameter K
4. choose **voting schema** (eg., weighted sum)
5. choose **Threshold** on score for classification.

viii. simmilarity

ix. Pros

1. robust
2. simple
3. training fast

x. Cons

1. depends on similarity measure
2. easily fooled by irrelevant attributes
3. compute expensive

b. Decision Tree

- i. Select which **attribute** to test at each **node** in the tree
- ii. **goal** is to select **most useful attribute** for classifying
- iii. **greedy** 贪婪 search ,picks the best attribute and **never looks back**
- iv. Pros
 1. **understandable** rules
 2. found the **most important feature** for classification

v. Cons

1. weak in **multi-classification** and **small training data**
2. expensive to train to pruning 修剪

1. 多分类问题:

决策树在处理多分类问题时可能会遇到困难。这是因为决策树的分裂过程需要选择最佳的特征来进行节点分裂，而在多分类情况下，可能存在多个候选特征，难以确定哪个特征最适合进行分裂。这可能导致决策树在多分类问题上表现不佳，尤其是当类别之间的边界不清晰时。

2. 小训练数据:

决策树在小型训练数据集上可能表现不佳。决策树倾向于过度拟合小型数据集，特别是当数据中存在噪声或特征之间的关系不清晰时。过度拟合会导致决策树在新数据上泛化能力较差，即使在训练数据上表现良好，也可能无法准确地对未见过的数据进行分类。

3. 训练成本:

决策树的训练成本可能相对较高，尤其是在执行修剪（pruning）操作时。修剪是一种减少决策树复杂度的技术，旨在防止过度拟合。然而，在训练过程中进行修剪需要对整个决策树进行逐级检查，以确定哪些节点可以被剪枝，这可能需要消耗大量的计算资源和时间。

综上所述，虽然决策树是一种直观且易于理解的机器学习算法，在处理多分类和小训练数据时可能表现不佳，并且训练过程可能较为昂贵。在实际应用中，这些限制可能需要通过使用其他算法或采取特定的预处理和调整步骤来克服。

c. Naive Bayes 朴素贝叶斯

i. NB classifier parameter estimation

1. $P(c_j)$, from the **frequency** of classes in the training examples.
2. $P(x_1, x_2, \dots, x_n | c_j)$. Need a very **large training** examples 很大训练集

ii. NB Assumptions

1. **Conditional independence Assumption**(条件独立假设, 核心)

具体来说，对于给定的类别 c_j ，条件独立假设表明特征 x_1, x_2, \dots, x_n 的联合概率等于各个特征单独发生的概率的乘积，即：

$$P(x_1, x_2, \dots, x_n | c_j) = P(x_1 | c_j) \times P(x_2 | c_j) \times \dots \times P(x_n | c_j)$$

这意味着在给定类别的条件下，我们可以将特征之间的关系简化为互相独立的关系。换句话说，我们假设每个特征对于类别的影响是相互独立的。

2. **ignore position** of words,
3. use **BOW** compute probability (避免考虑太多的可能性，使得模型的计算更加高效。)
4. steps:
 - a. extract **Vocabulary** (e.g., remove stop words)
 - b. calculate $P(C_j)$ for every **class**
 - c. calculate $P(X_k | C_j)$ for every **word** in every **class**

1. 提取词汇表:

- 首先, 我们从训练语料库中提取词汇表。词汇表包含了训练数据中出现过的所有不同的词汇。这个过程通常涉及将文档拆分为单词, 并去除停用词等不重要的词汇, 从而得到一个干净的词汇表。

2. 计算类别先验概率 $P(c_j)$:

- 对于每个类别 c_j , 我们需要计算它在训练数据中的先验概率。这个概率可以通过统计训练数据中属于该类别的文档数量, 并除以总文档数量来得到。这样就得到了类别 c_j 的先验概率。

3. 计算特征条件概率 $P(x_i|c_j)$:

- 对于每个类别 c_j 和词汇表中的每个词汇 x_i , 我们需要计算在给定类别下该词汇出现的条件概率。具体地, 我们统计训练数据中属于类别 c_j 的所有文档, 将它们合并成一个单一的文本 (Text), 然后计算词汇 x_i 在该文本中的出现次数 N_{ij} 。最后, 我们将 N_{ij} 除以词汇表的大小来得到条件概率 $P(x_i|c_j)$ 。

5. advantages

- a. **robust**, better than decision tree in irrelevant features
- b. good at many **equally important features**, better than decision tree
- c. good **baseline**
- d. **fast**
- e. **low storage**

d. TF-IDF

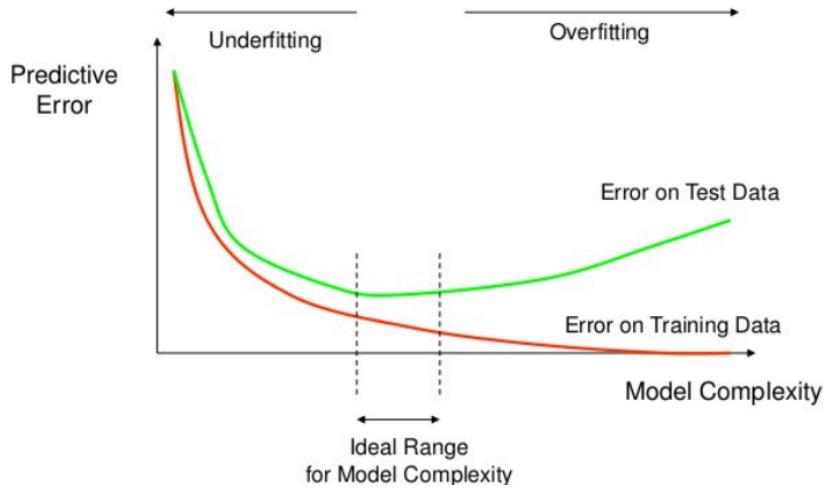
- i. query **ranking** .(sum up ti-idf for each query term)
- ii. **filter** stop word. (remove lowest tf-idf terms)
- iii. **evaluate** importance (a word for a doc in a corpus)

e. others

- i. decision trees
- ii. naive Bayes
- iii. KNN (K nearest Neighbors)
- iv. Perceptron
- v. Support Vector Machines (SVM) 属于 机器学习
- vi. CNN

18. Evaluating Results

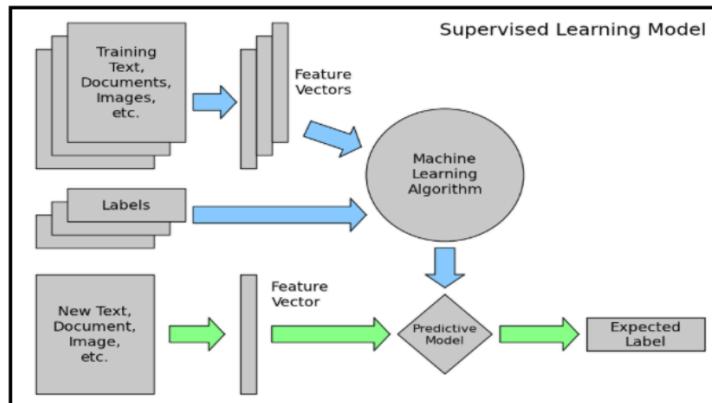
- a. Confusion matrix
- b. avoid overfitting(模型过于复杂, 数据不足, 特征选择不当, 噪声数据)
 - i. separate test data
 - 1. n-fold cross-validation (一个子集作为验证集)
 - 2. leave-one-out cross-validation(一个样本作为验证集)
 - ii. separate Train / Validation and Test



- c. compare with **baseline**
 - i. minimum performance
 - ii. competing system? dumb 愚蠢 but easy method(random, most -frequent, very simple heuristic启发式)?
 - iii. same test data**
 - iv. Gold standard , with best achieved results, **Leaderboard**

19. Deep Learning

a. supervised Learning model



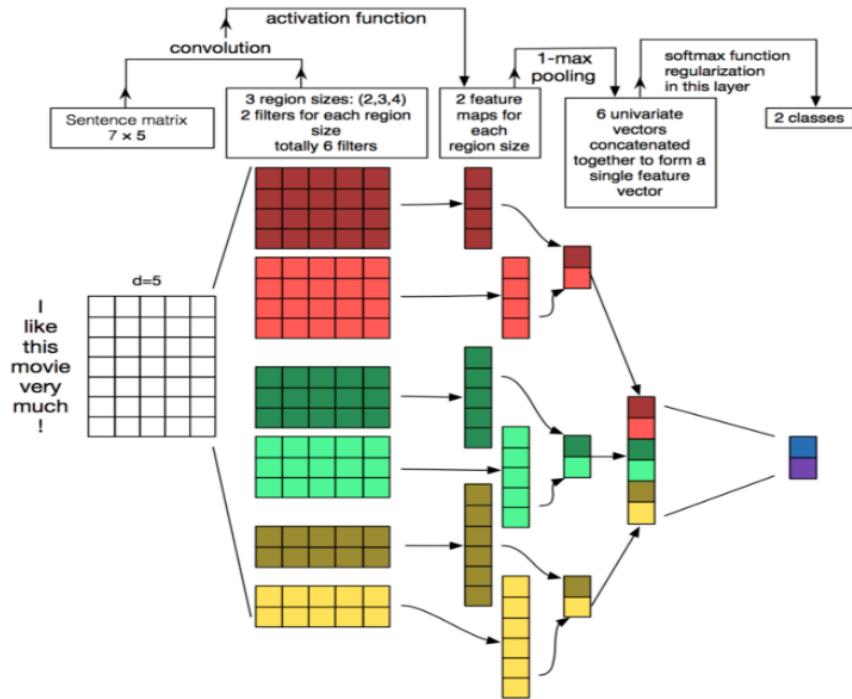
The labels vector is a **one hot** representation of the labels.

The labels vector can be any a $n \times 1$ dimension matrix where n is the number of labels.

The labels vector has to be at least a **2x1 matrix**.(Binary categorisation二分类 , multi category classification 多分类)

feature vectors has same dimensions .(简\稳健\高效)

b. CNN



i. matrix Convolution

$\begin{array}{ c c c } \hline 2 & 2 & 1 \\ \hline 3 & 1 & -1 \\ \hline 4 & 3 & 2 \\ \hline \end{array}$	$1*2 - 1*2 - 1*3 + 0*1 + 2 = -1$
--	----------------------------------

\downarrow

$\begin{array}{ c c } \hline 1 & -1 \\ \hline -1 & 0 \\ \hline \end{array}$	$1*2 - 1*2 - 1*1 + 0*-1 + 2 = 2$
---	----------------------------------

$\begin{array}{ c c } \hline -1 & \\ \hline & \\ \hline \end{array}$	$1*3 - 1*1 - 1*4 + 0*3 + 2 = 0$
--	---------------------------------

\downarrow

$\begin{array}{ c c } \hline & \\ \hline & \\ \hline \end{array}$	$1*1 - (-1)*1 - 1*3 + 0*2 + 2 = 1$
---	------------------------------------

\downarrow

Input matrix (3x3)
no padding
1 channel

Kernel (2x2)
Stride 1
Bias = 2

Feature map (2x2)

$$\begin{array}{|c|c|} \hline -1 & 2 \\ \hline 0 & 1 \\ \hline \end{array}$$

$1*2 - 1*2 - 1*3 + 0*1 + 2 = -1$

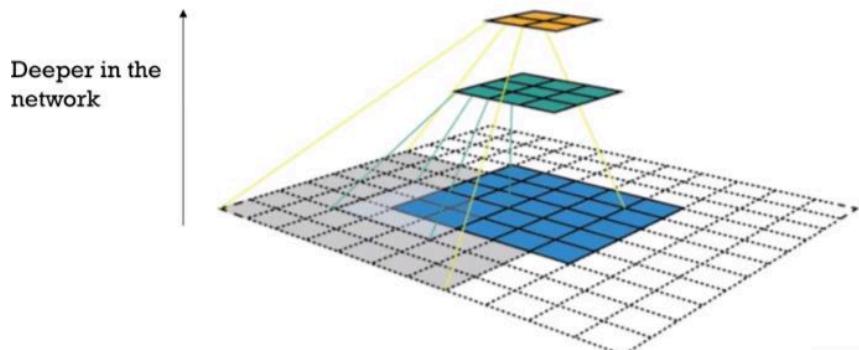
$1*2 - 1*2 - 1*1 + 0*-1 + 2 = 2$

$1*3 - 1*1 - 1*4 + 0*3 + 2 = 0$

$1*1 - (-1)*1 - 1*3 + 0*2 + 2 = 1$

1.

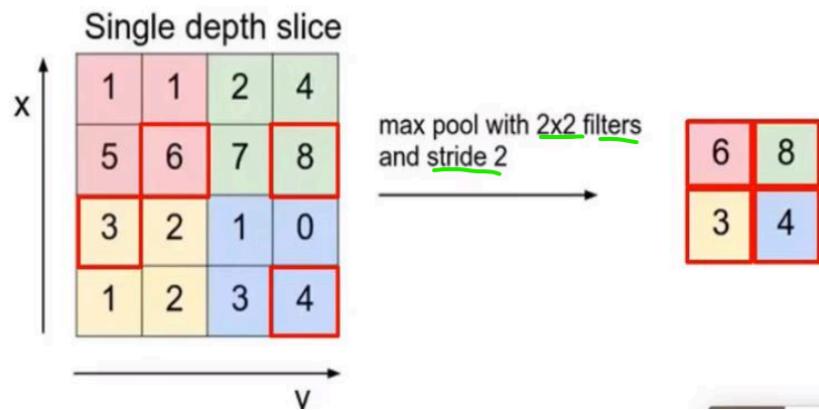
2. Convolution to reduce dimensions



3.

ii. Max pooling

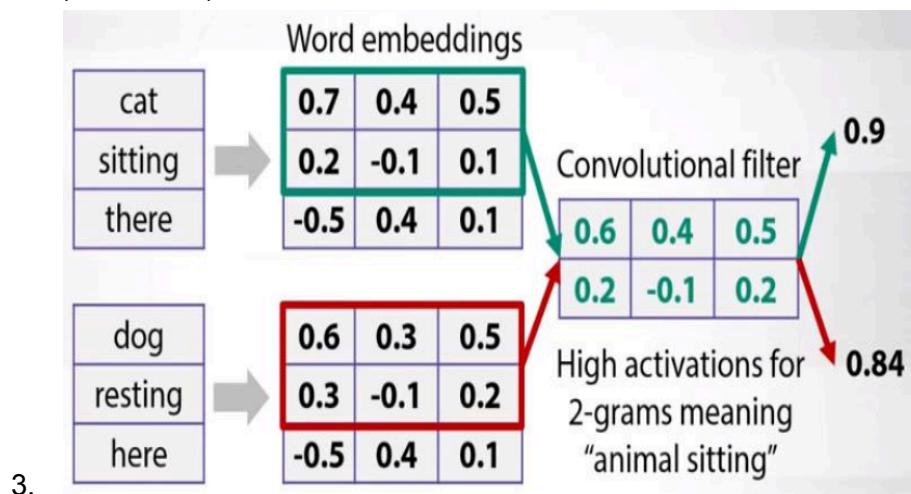
1. picks the biggest number from a window



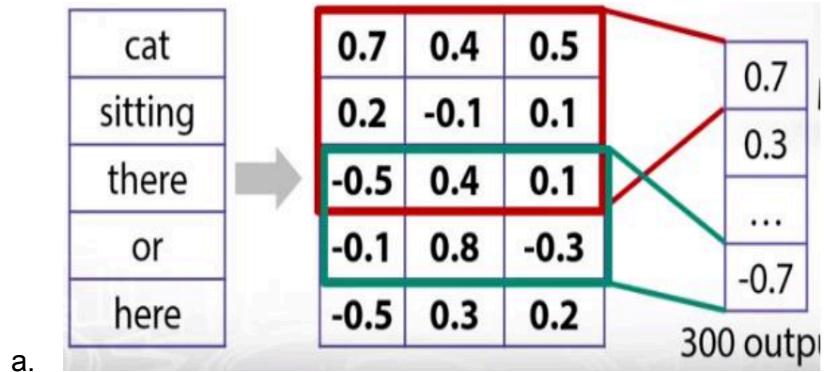
iii. matrix convolution (Text)

1. Convolve to lower dimensions

2. (mentioned)



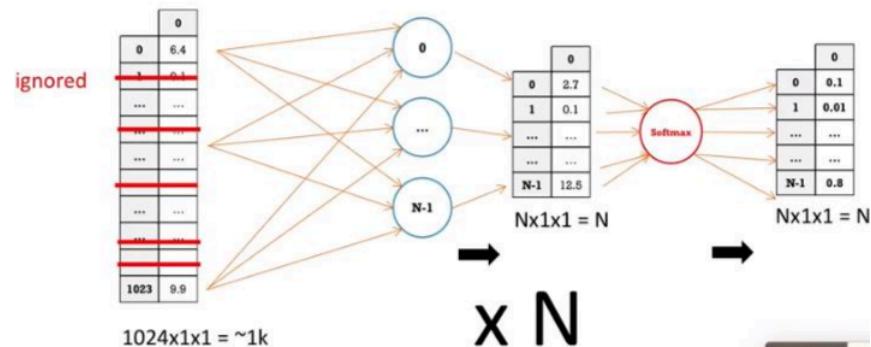
iv. Max Pooling (Text)



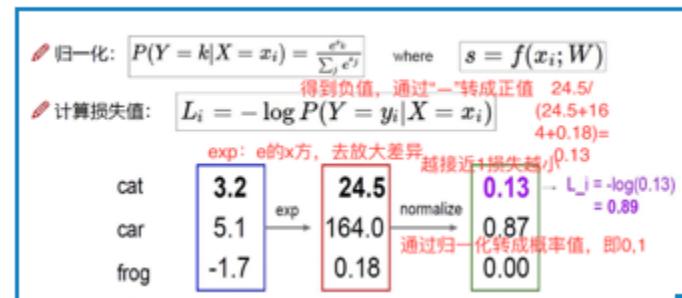
v. Fully connected neural net at the end

- flattened matrix, dropout, softmax For n categories

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{N-1} e^{z_j}}$$



softmax分类器：



Softmax 函数是一种常用的激活函数，主要用于多分类任务中的输出层。它接受一个包含多个元素的向量作为输入，并将每个元素转换为表示其在总和为1的概率分布中的相对大小的概率值。

简单来说，Softmax 函数可以将输入向量中的每个元素映射到0到1之间的值，并确保所有元素的总和为1。这使得 Softmax 函数在多分类问题中非常有用，因为它可以将模型输出的原始分数转换为各个类别的概率值，从而使得我们可以选择概率最高的类别作为最终的预测结果。

Softmax 函数的数学表达式如下：

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

其中， x_i 是输入向量中的第 i 个元素， e 是自然对数的底， $\sum_j e^{x_j}$ 是所有输入元素的指数函数之和。

因此，Softmax 函数可以将任意实数向量转换为一个概率分布，这使得它在多分类问题中非常有用。

20. Summary

a. Term

- i. algorithm term
 - 1. word embedding
 - 2. n-grams
 - 3. sliding window (固定大小的窗口，用于提取特征或计算统计量)
 - 4. strides (在卷积操作中，核在输入数据上滑动的步长，影响输出特征图的尺寸。)
 - 5. kernels (积操作中使用的滤波器，用于从输入数据中提取特征)
 - 6. max pooling
 - 7. activation function(输入信号转换为输出信号的函数，常用的包括ReLU、Sigmoid和Tanh等)
 - 8. flattened matrix
- ii. DeepLearning term
 - 1. Fully connected NN
 - 2. dropout (正则化技术，通过在训练过程中随机将部分神经元的输出置零来减少过拟合)
 - 3. softmax function (多分类问题的激活函数，将神经网络输出转换为表示概率分布的向量)
 - 4. back propagation
 - 5. loss function

b. Approach

- i. machine learning
 - 1. SVM(support vector machines)
 - 2. KNN(K-nearest neighbors)
 - 3. Decision Tree
- ii. Deep Learning
 - 1. LTSM(Long Short-Term Memory)

- a. LSTM代表“长短期记忆”(Long Short-Term Memory)，是一种用于处理序列数据的循环神经网络(RNN)架构。与标准的RNN相比，LSTM具有更强大的记忆能力，能够更有效地处理长序列数据并捕捉序列中的长期依赖关系。
 - b. LSTM通过使用称为“门”的结构来实现记忆和控制信息流。它包括三个门：遗忘门(Forget Gate)、输入门(Input Gate)和输出门(Output Gate)。这些门控制着信息的流动，以及何时保留、删除或输出信息。
2. RNN
 3. CNN
 4. Perceptron
-

Lecture 07 Information Extraction (NER)

21. NER(Named Entity Recognition)

- a. person name; org; location; Date; Email; Web address; ...
- b. it is a Classification problem
- c. very similar to other NLP tasks, such as POS tagging

22. Knowledge NER

- a. (precise\small training data \ expensive dev \ domain dependent\ changes over time\hard maintain)
- b. Rule based NER
 - i. regular expression
 - ii. customer rule (eg. capitalized word + {city, center, river} Location, Ex. New Zealand)
 - iii. Context patterns
 - 1. [PERSON] earned [MONEY] ; eg. Frank earned \$20
- c. difficult with Rule based
 - i. new proper name emerge (movie; books; singer; restaurants)
 - ii. No lexicon 词典 contains all proper names
 - iii. same entity has multiple variants(总理\工党领袖\雅顿)
 - iv. proper names are ambiguous(person\location)
- d. Decision Tree
 - i. requires identification of attributes to classify.(需要识别属性)

23. Learning Systems

- a. (high recall \ no developer \ no expert\ annotations are cheap\ **large training data**)

	data	methods	example
supervised	labeled training data	Hidden Markov Models KNN Decision Trees AdaBoost SVM NN	NE recognition POS Tagging Parsing
unsupervised	labels must be automatically discovered	clustering	NE disambiguation 消歧义, text classification
Semi-supervised 半监督	small percentage is labeled	bootstrapping ,(Using techniques such as regular expressions to identify and label data and then feeding the data into a learning system) active learning, co-training, self-training	NE recognition, POS tagging, Parsing

24. Features for NE Detection

- a. The **more useful** feature, the **more powerful** your learner gets
- b. contextual 上下文 ngram?
 - i. current word W0
 - ii. words around W0 in [-3,...,+3] window
 - iii. eg. Adam Smith works for ibm in London
- c. Orthographic 规范和约定
 - i. initial-caps, all-caps, all-digits, roman-number, URL, punctuation-mark, acronym 缩写, contains-hyphen 连字符
 - ii. eg. Adam Smith works for IBM in London
- d. part of speech tag (when available)
- e. BOW
- f. Trigger word
 - i. for person (Mr, Miss, Dr, PhD)
 - ii. for location(city,street)
 - iii. for org(Ltd.,co.)
- g. Gazetteers 地名
 - i. geographical

- ii. first name
- iii. surname
- iv. company names

25. Machine Learning NER

- a. NED(Named Entity Detection) 实体探测
 - i. Identify named entities using BIO Tags
 - ii. Both machine and Deep learning
 - iii. 3 categories
 - iv. 2 steps (打标, 提取)

1. (B beginning an entity开始, I continues the entity 延续, O word outside the entity外围)

Adam_B-PER Smith_I-PER works_O for_O IBM_B-ORG,_O London_B-LOC._O

2. extract continuous B & I (Adam Smith, IBM, London)

- b. NEC(Classify into a predefined set of categories) 归类
 - i. person name
 - ii. org
 - iii. location
 - iv. miscellaneous 杂类 (movies,sport event, etc)

26. Process

import nltk nltk.download('punkt') nltk.download('averaging_perceptron_tagger') nltk.download('maxent_ne_chunker') nltk.download('words')	Jacinda Ardern is the Prime Minister of New Zealand but Roenzo isn't. Please correct the result.	corpus
Sentence= nltk.sent_tokenize(text)	["Jacinda Ardern is the Prime Minister of New Zealand but Roenzo isn't.", 'Please correct the result.]	句子
word_tokenize(Sentence)	['Jacinda', 'Ardern', 'is', 'the', 'Prime', 'Minister', 'of', 'New', 'Zealand', 'but', 'Roenzo', 'is', "n't", '.']	词 token
pos_tag(tokens)	[('Jacinda', 'NNP'), ('Ardern', 'NNP'), ('is', 'VBZ'), ('the', 'DT'), ('Prime', 'NNP'), ('Minister', 'NNP'), ('of', 'IN'), ('New', 'NNP'), ('Zealand', 'NNP'), ('but', 'CC'), ('Roenzo', 'NNP'), ('is', 'VBZ'), ("n't", 'RB'), ('.', '.')]	POS
ne_chunk(pos_tags)	(S (PERSON Jacinda/NNP)	Parsing

<pre>or ne_chunk_sents(tagged_sentences)</pre>	<pre>(ORGANIZATION Ardern/NNP) is/VBZ the/DT Prime/NNP Minister/NNP of/IN (GPE New/NNP Zealand/NNP) but/CC (PERSON Roenzo/NNP) is/VBZ n't/RB ./.)</pre>	
<pre># Print all entity names print(entity_names) # Print unique entity names print (set(entity_names))</pre>		

高级类库, 一步完成

```
# # Using Spacy NER
import spacy

nlp = spacy.load('en_core_web_sm') # Install en_core_web_sm
if required.
doc = nlp(sample)

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
#Grant Robertson 101 116 PERSON
#12 146 148 CARDINAL
#today 188 193 DATE
#seven-day 199 208 DATE
```

trick

27. multiple choice : 2 answer

28. <<Introduction to Information Retrieval – Manning>
><https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>