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

AUT

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 boarder line 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.

AUT

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

AUT

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

AUT

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?

AUT

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)

AUT

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.

AUT

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

AUT

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

AUT

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

AUT

Rhetorical Structure Theory

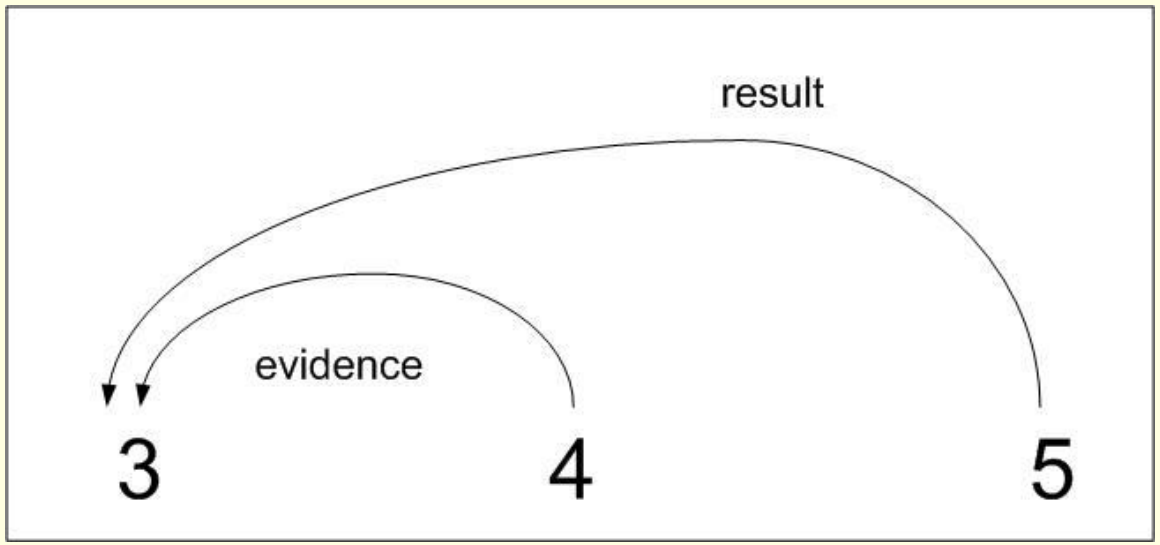
- Assumption: Clauses in well-formed text are related via predefined rhetorical relations
- **Evidence**: a claim information intended to increase the readers' belief in the claim

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.

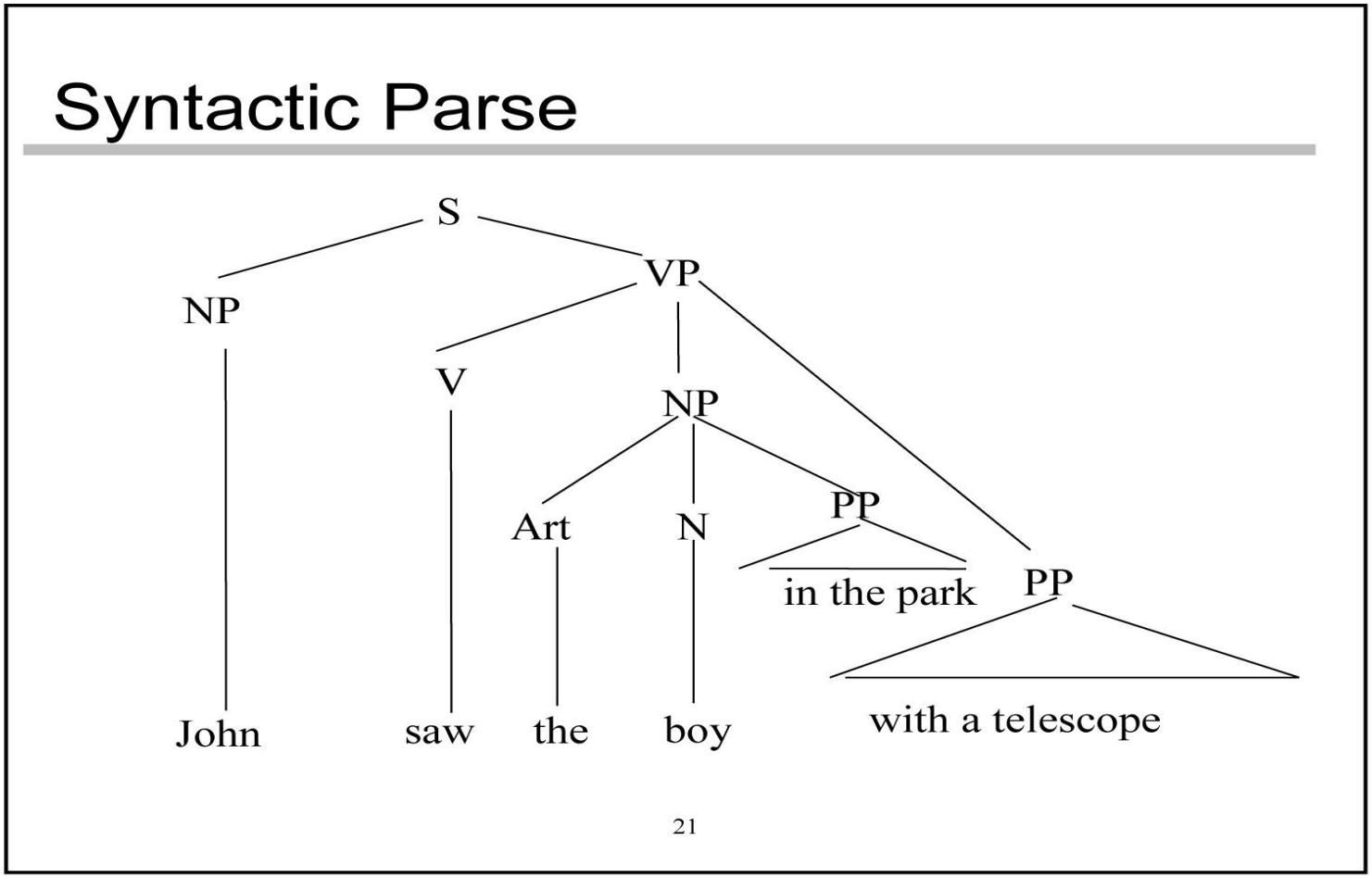


Basic levels of language processing

- 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!

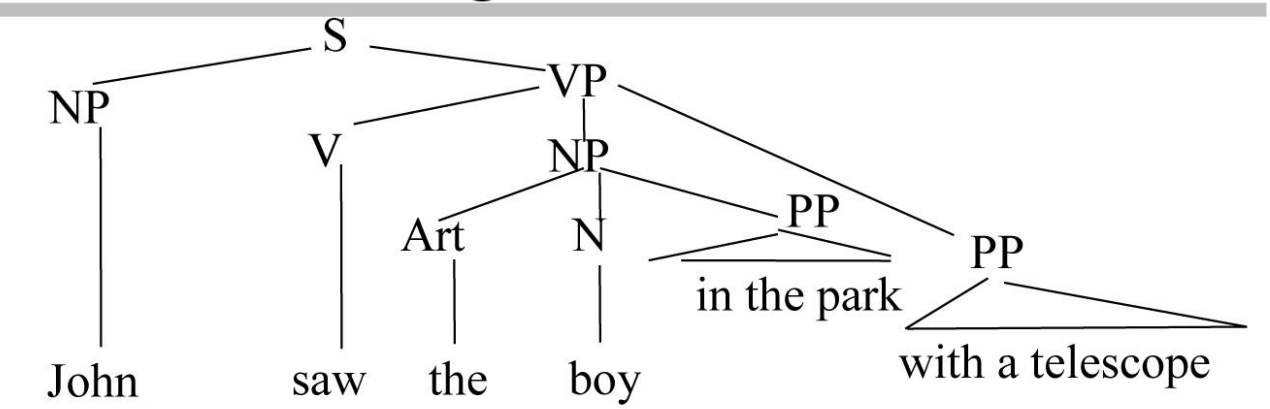
Syntactic structure of a sentence

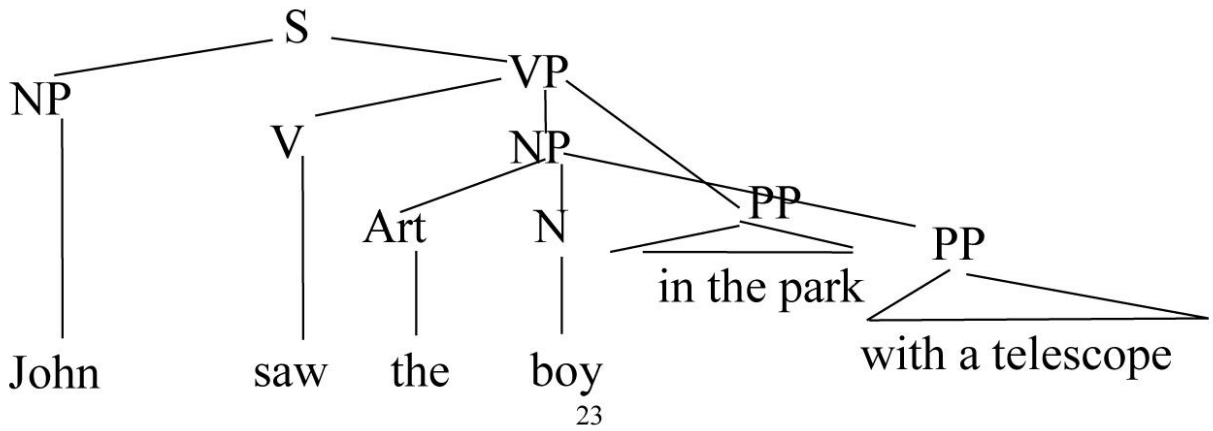
- John saw the boy in the park with a telescope



Two possible parses

Structural ambiguities





23

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

AUT

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

AUT

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 *there*	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)

29

POS tags

PoS Tagging

25.	TO	*to*	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-3rd ps.sing. present	43.	)	Right bracket character
32.	VBZ	Verb, 3rd ps.sing. present	44.	"	Straight double quote
33.	WDT	*wh*-determiner	45.	‘	Left open single quote
34.	WP	*wh*-pronoun	46.	“	Left open double quote
35.	WP$	Possesive *wh*-pronoun	47.	’	Right close single quote
36.	WRB	*wh*-adverb	48.	”	Left close double quote

30

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 *wh*-word or *wh*-phrase
8.	SINV	Declarative sentence with subject-aux inversion
9.	SQ	Subconstituent of SBARQ excluding *wh*-word or *wh*-phrase
10.	VP	Verb phrase
11.	WHADVP	*wh*-adverb phrase
12.	WHNP	*wh* -noun phrase
13.	WHPP	*wh* -prepositional phrase
14.	X	Constituent of unknown or uncertain category
	Null elements	
1.	*	"Understood" subject of infinitive or imperative
2.	0	Zero variant of *that* in subordinate clauses
3.	T	Trace-marks position where moved *wh*-constituent is interpreted
4.	NIL	Marks position where preposition is interpreted in pied-piping contexts

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
                                       (ADVP (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

AUT

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
  - $P(w_n|w_1,w_2\ldots w_{n-1})$
- 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(*the* | *its water is so transparent that*)
- By definition that's :

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

How would you get this?

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

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.

AUT

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

AUT

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.

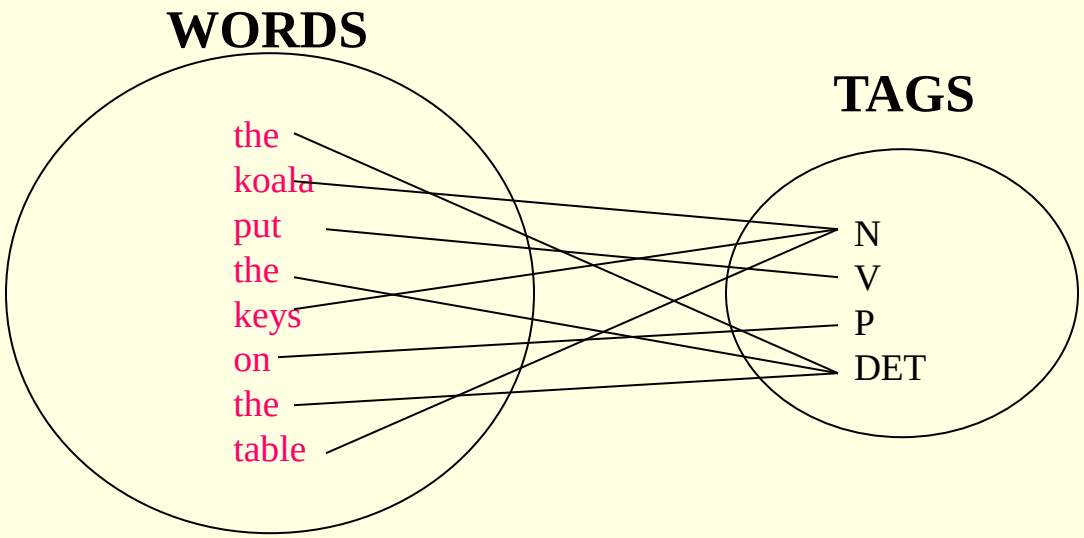
AUT

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

AUT

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.

AUT

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.

AUT

- 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**.

AUT

- 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 $\Rightarrow$ 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?

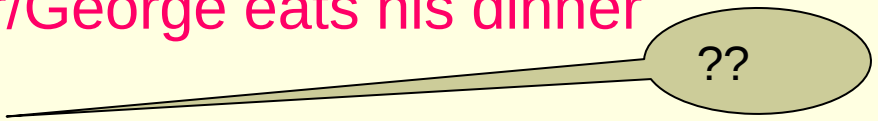
- Nouns denote people, places and things and can be preceded by articles? But…
  - My ***typing*** is very bad.
  - *The ***Mary*** loves John.
- Verbs are used to refer to actions, processes, states
  - But some are closed class and some are open

  I ***will have emailed*** everyone by noon.
- Adverbs modify actions
  - Is Monday a temporal adverbial or a noun?

AUT

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

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***

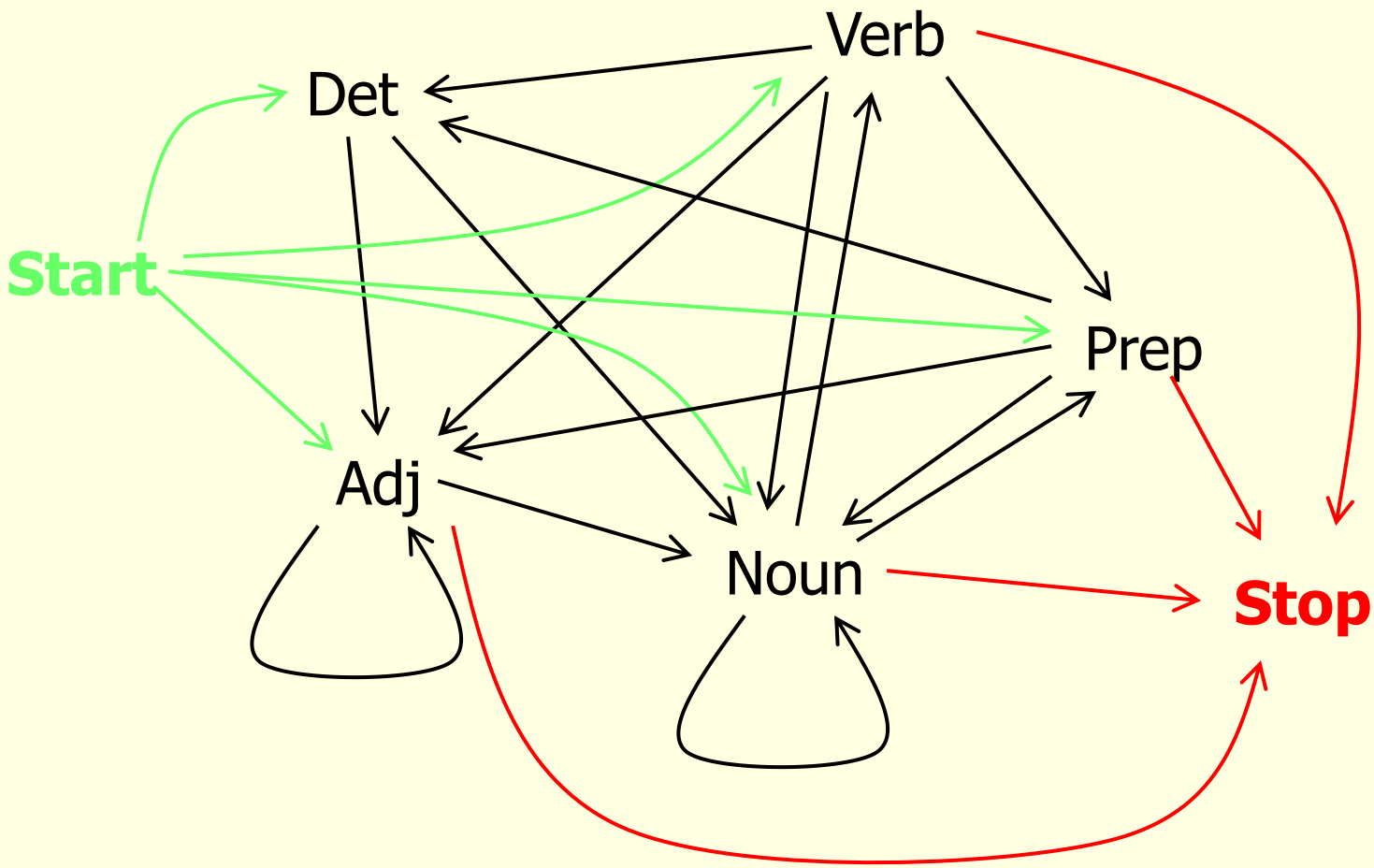
Tagging Whole Sentences with POS is Hard

- 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

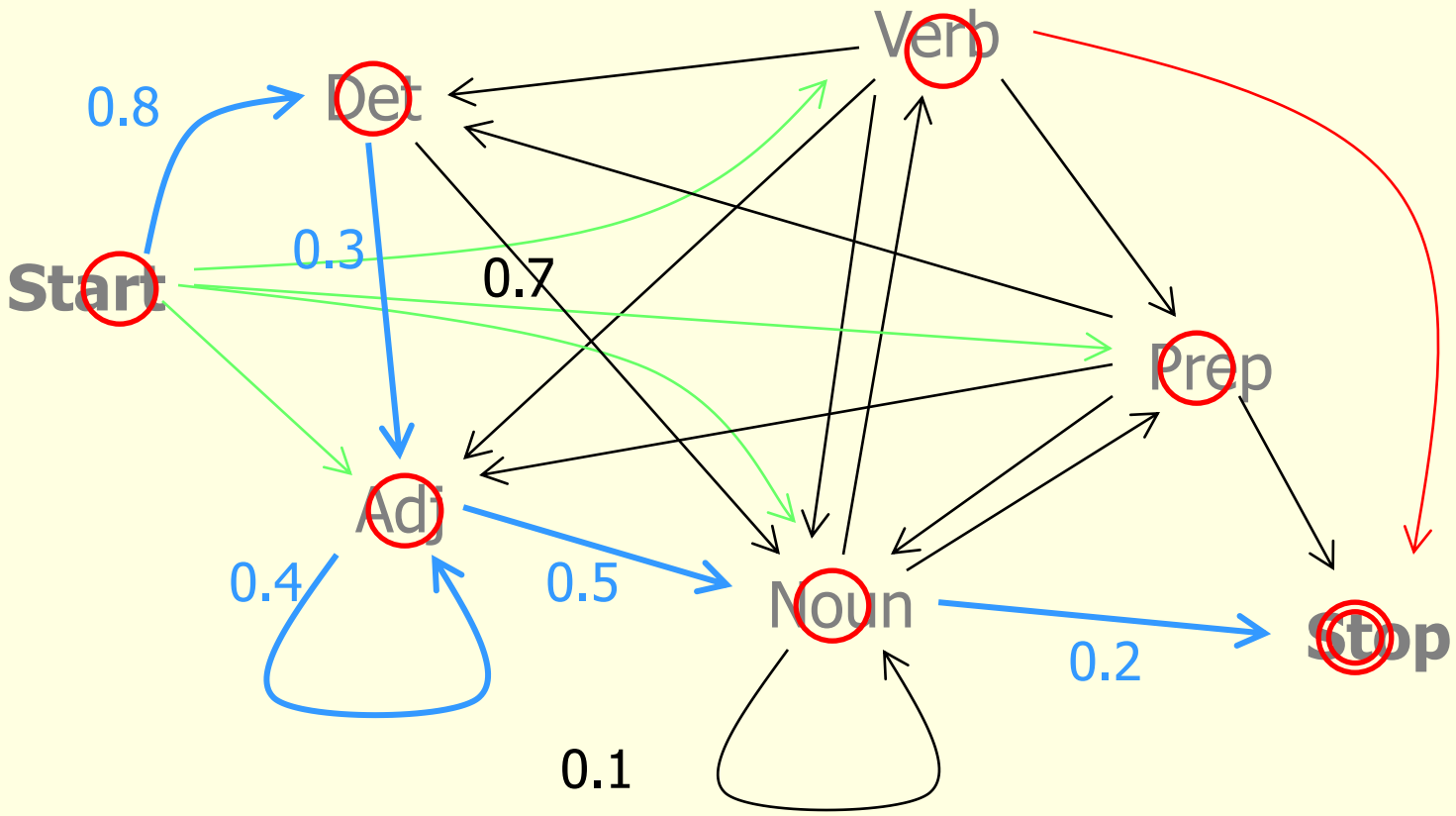
- For tokens w1, ..., wn, find the most probable corresponding sequence of possible tags t1, ..., tn
- We assume that *probable* means something like “most frequently observed in some manually tagged corpus of words”.
- Penn Treebank II (a common training corpus)
  - 1 million words from the Wall Street Journal
  - Tagged for POS (and other attributes)

Markov Model (bigrams)



Markov Model as a FSM

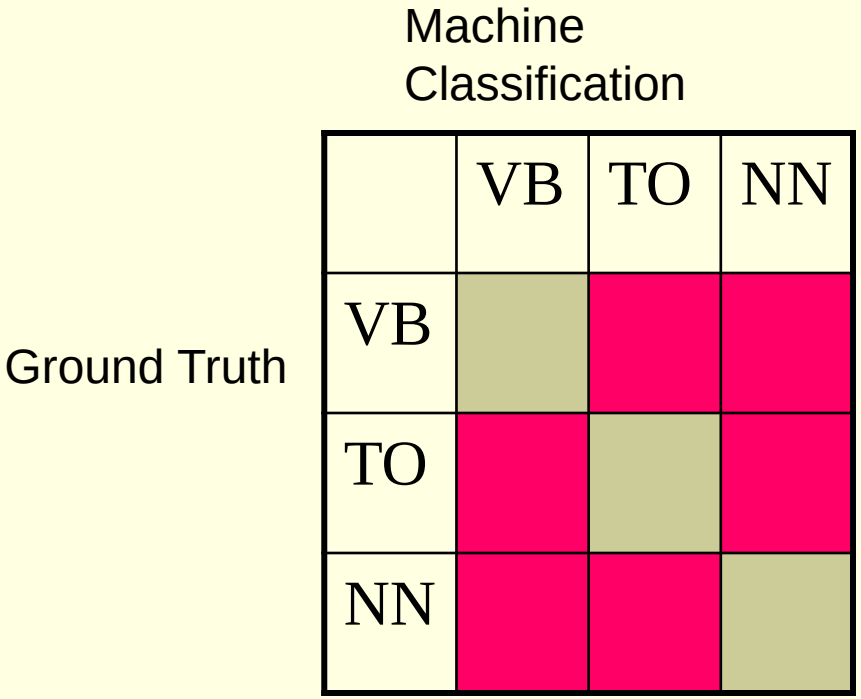
p(tag seq)



Start Det Adj Adj Noun **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?



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

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

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

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

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

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

AUT

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

AUT

Multi Category Evaluation

		Machine Computation		
		VB	TO	NN
Ground Truth	VB	420	20	300
	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?

AUT

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

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...

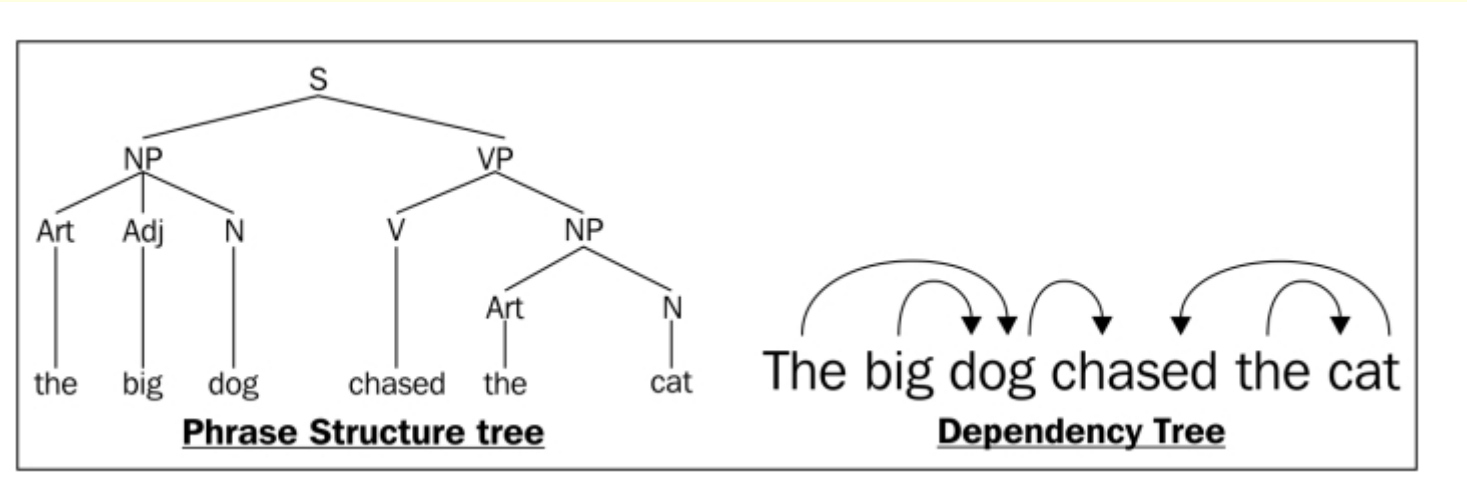
```
VP → VBD PP
VP → VBD PP PP
VP → VBD PP PP PP
VP → VBD PP PP PP PP
```

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

AUT

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 is presented first as in a google search.
  - **What are the possibilities for ranking?**

AUT

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}

AUT

Vector Space Representation

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

AUT

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) = (**w**(t1, d), **w**(t2, d), . . . , **w**(tn, d))
- Sparse matrix for most documents
- Question: does this representation allow to compute the similarity between documents ?
- Similarity between vectors ?
  - inner product V(d1).V(d2)
- 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?**

AUT

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

AUT

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*/df*t*) instead of *N*/df*t* 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		

AUT

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:

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

- **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:

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

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

AUT

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?

AUT

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) 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) <= d(x,y) + d(y,z).

AUT

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

$$D_E(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{1/2}$$

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

Cosine Distance

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

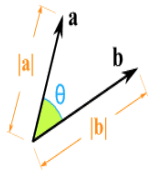
$$SIM_C(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|}$$

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

a · b

This means the Dot Product of **a** and **b**

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



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

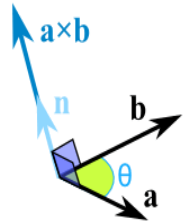
Where:
|**a**| is the magnitude (length) of vector **a**
|**b**| is the magnitude (length) of vector **b**
θ is the angle between **a** and **b**

So we multiply the length of **a** times the length of **b**, then multiply by the cosine of the angle between **a** and **b**

WE CAN CALCULATE THE CROSS PRODUCT THIS WAY:

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

- |**a**| is the magnitude (length) of vector **a**
- |**b**| is the magnitude (length) of vector **b**
- θ is the angle between **a** and **b**
- **n** is the unit vector at right angles to both **a** and **b**



So the **length** is: the length of **a** times the length of **b** times the sine of the angle between **a** and **b**,

Then we multiply by the vector **n** to make sure it heads in the right **direction** (at right angles to both **a** and **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.

$$SIM_J(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{\left|\overrightarrow{t_a}\right|^2 + \left|\overrightarrow{t_b}\right|^2 - \overrightarrow{t_a} \cdot \overrightarrow{t_b}}.$$

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

AUT

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 $T = \{t_1, \ldots, t_m\}$, a commonly used form is

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

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 $\overrightarrow{t_a} = \overrightarrow{t_b}$. 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 $SIM_P < 0$.

AUT

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**.

AUT

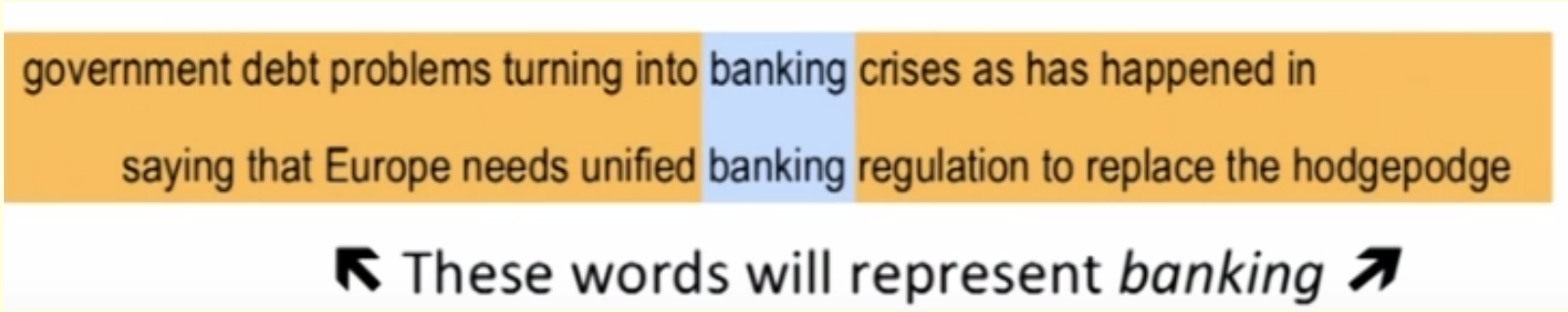
Language Models

- 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.

AUT

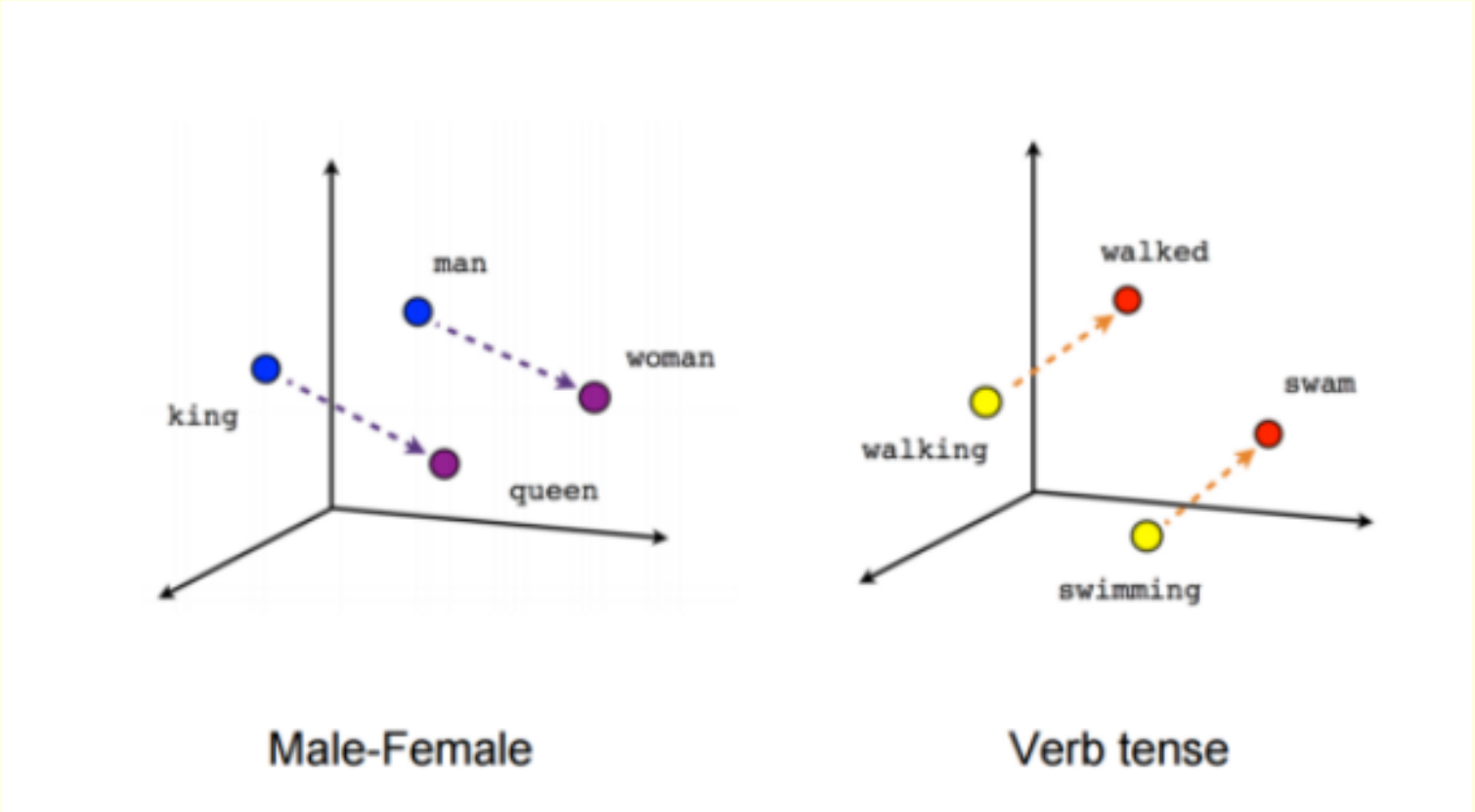
Two forms of meaning representation

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



Word Embedding

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



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.

AUT

Skip-gram model

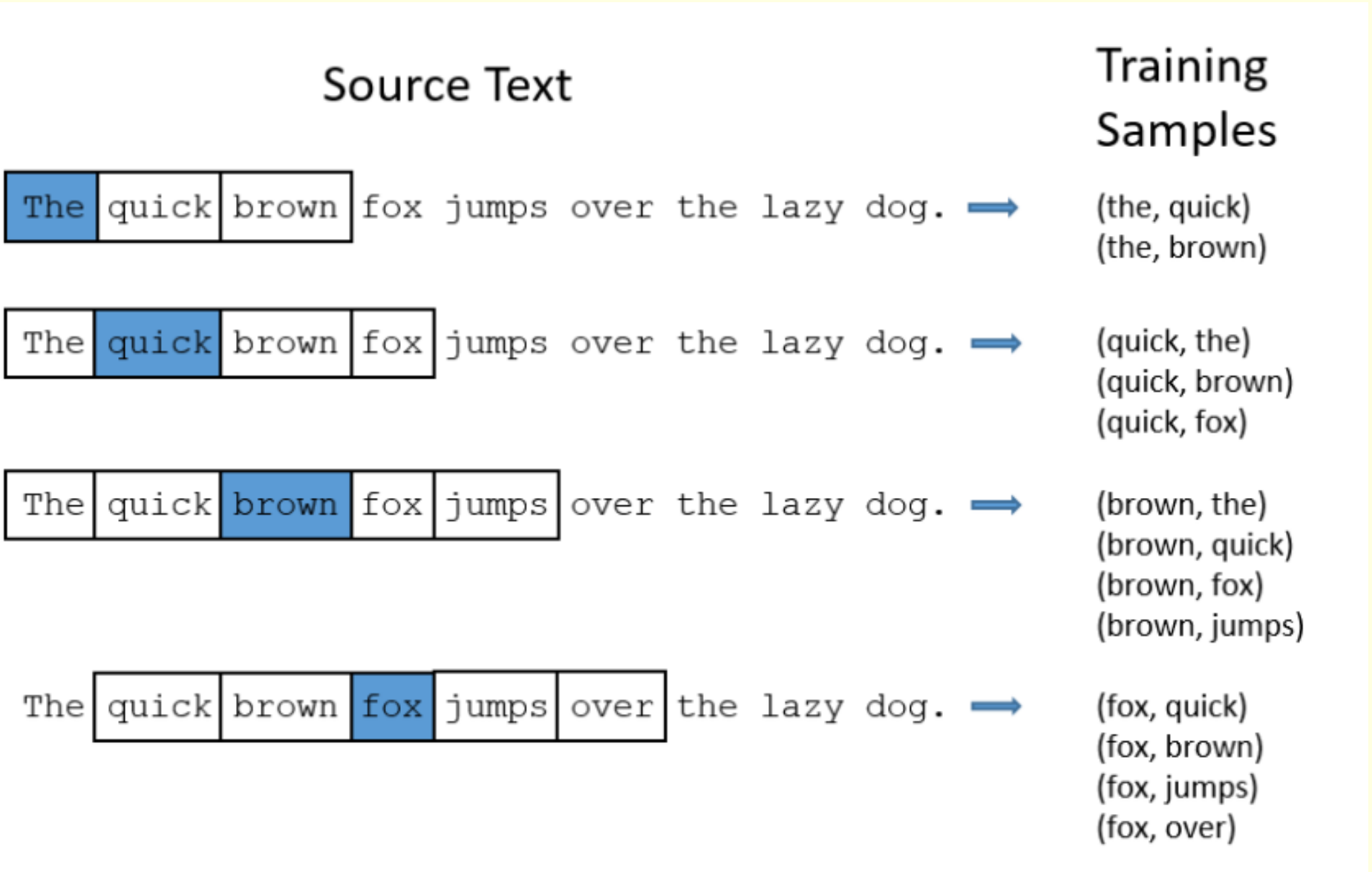
- “ *the quick brown fox jumped over the lazy dog*”
- Window - words to the left and to the right
  - Focus word: *quick*
  - Context words: the and brown for window size of 1.
  - Words: ***the*** and ***brown***
  - Whole dataset: *(quick, the), (quick, brown), (brown, quick), (brown, fox), ….*
- The 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.

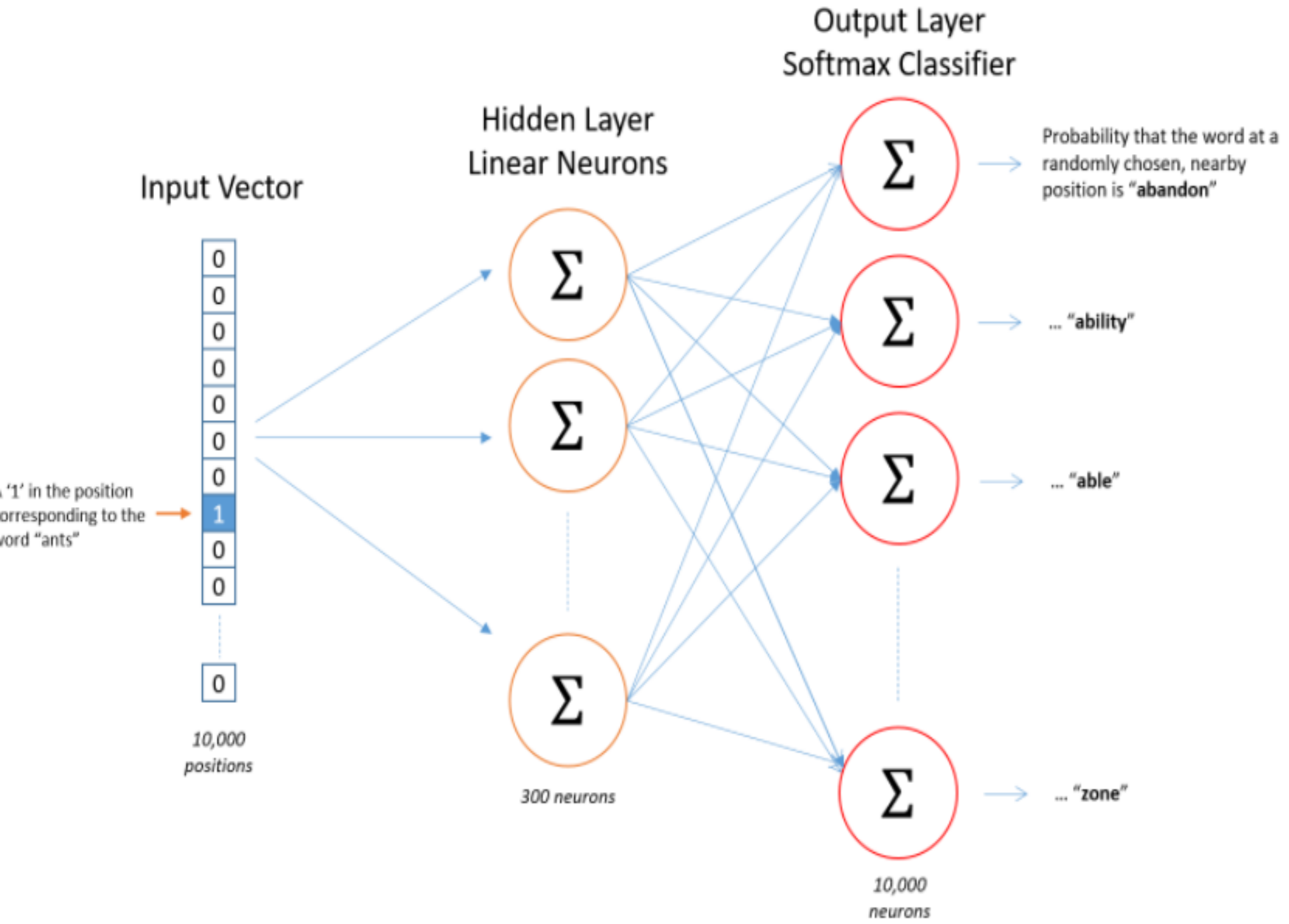
AUT

Skip-Gram model

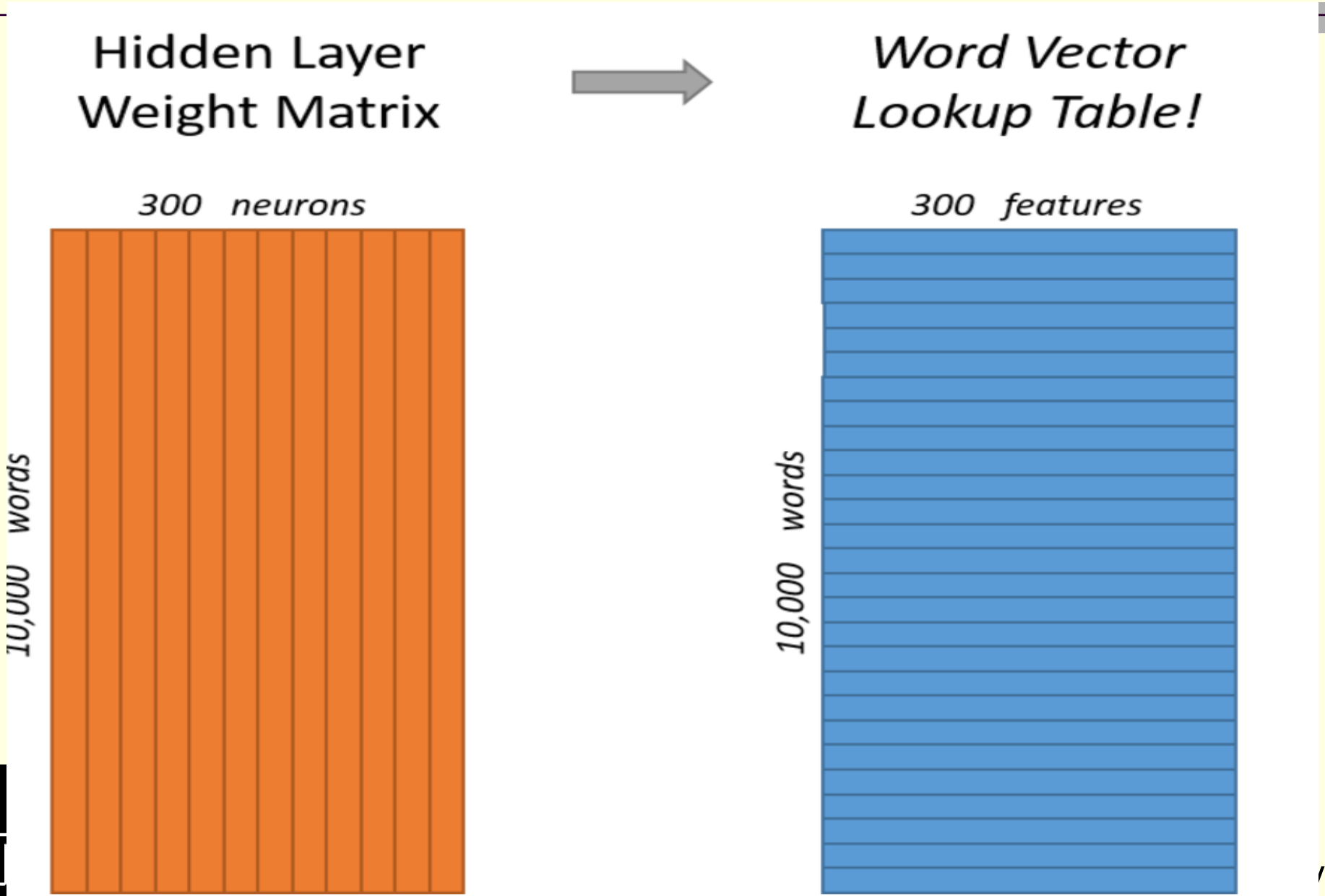
- Train a NN to on data and use the weights vector as the look up.
- Same principle applies for CBOW but in inverse.

An example for window size 2




Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
'1' in the position corresponding to the word "ants"
0
0
0
0
0
0
1
0
0
0
10,000 positions
300 neurons
10,000 neurons

The matrix functions as 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

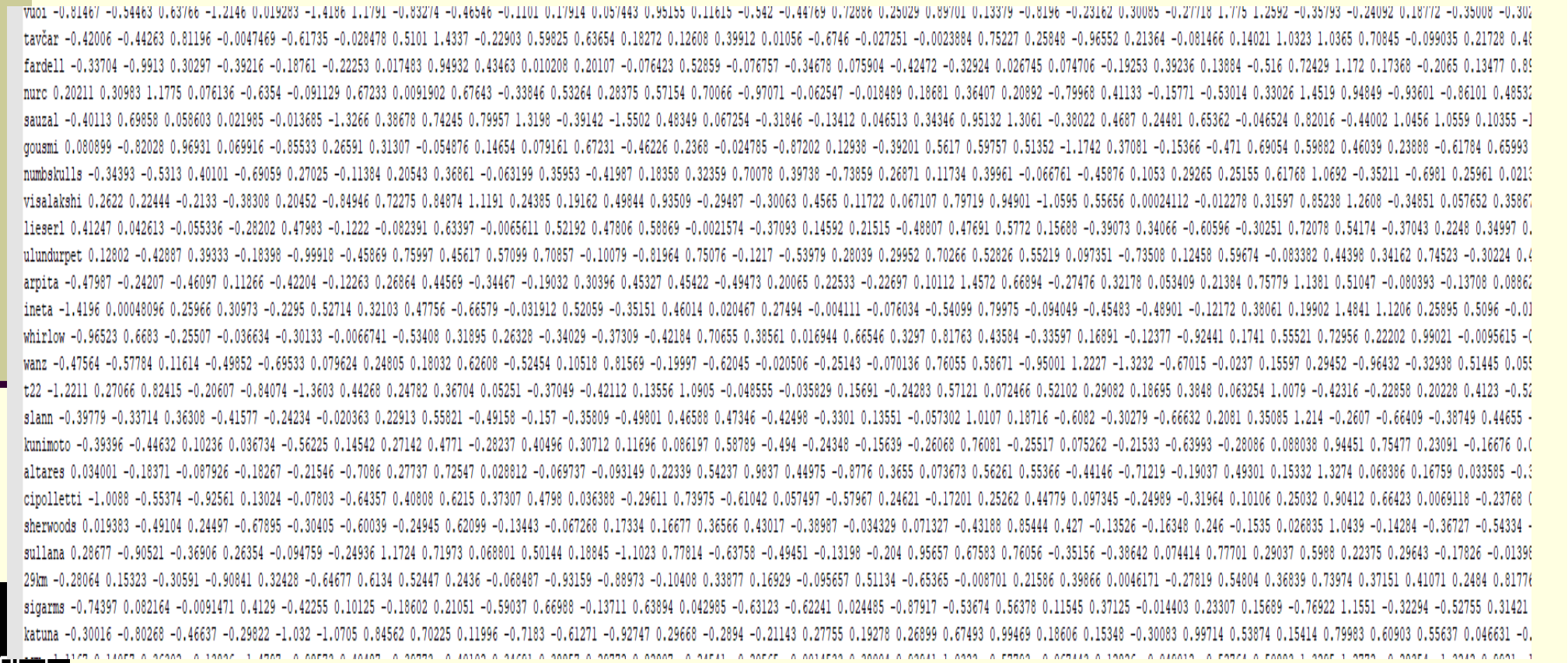
$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

GloVe: Global Vectors for Word Representation

- Developed by Stanford as open source project.
- Trained on Wikipeidia
- 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



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 Leaning
  - Lab on Deep Learning and conventional algorithms.

AUT

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”

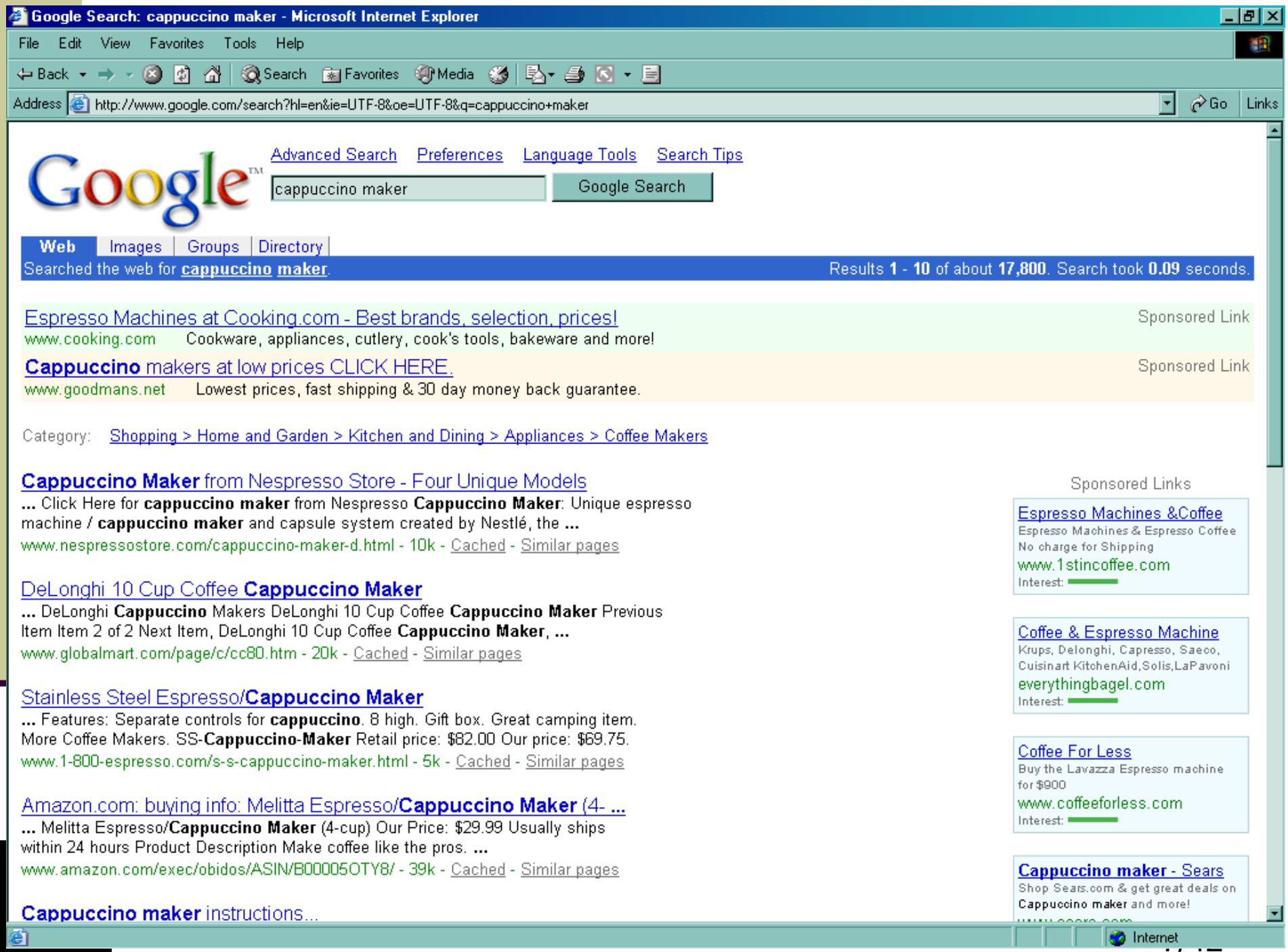
  e.g., “spam” : “not-spam”

  e.g., “contains adult language” :“doesn’t”

AUT

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”


Google Search: cappuccino maker - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=cappuccino+maker
Go Links
Google
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
... 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
DeLonghi 10 Cup Coffee Cappuccino Maker
... DeLonghi Cappuccino Makers DeLonghi 10 Cup Coffee Cappuccino Maker Previous Item 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
... 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- ...
... 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...
Sponsored Links
Espresso Machines &Coffee
Espresso Machines & Espresso Coffee No charge for Shipping
www.1stincoffee.com
Interest:
Coffee & Espresso Machine
Krups, Delonghi, Capresso, Saeco, Cuisinart KitchenAid,Solis,LaPavoni
everythingbagel.com
Interest:
Coffee For Less
Buy the Lavazza Espresso machine for $900
www.coffeeforless.com
Interest:
Cappuccino maker - Sears
Shop Sears.com & get great deals on Cappuccino maker and more!
Internet

Yahoo! Shopping - Search - cappuccino maker - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

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

YAHOO! SHOPPING

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 r Search

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

Done Internet

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, ...).

AUT

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.

AUT

Approaches to Text Categorization

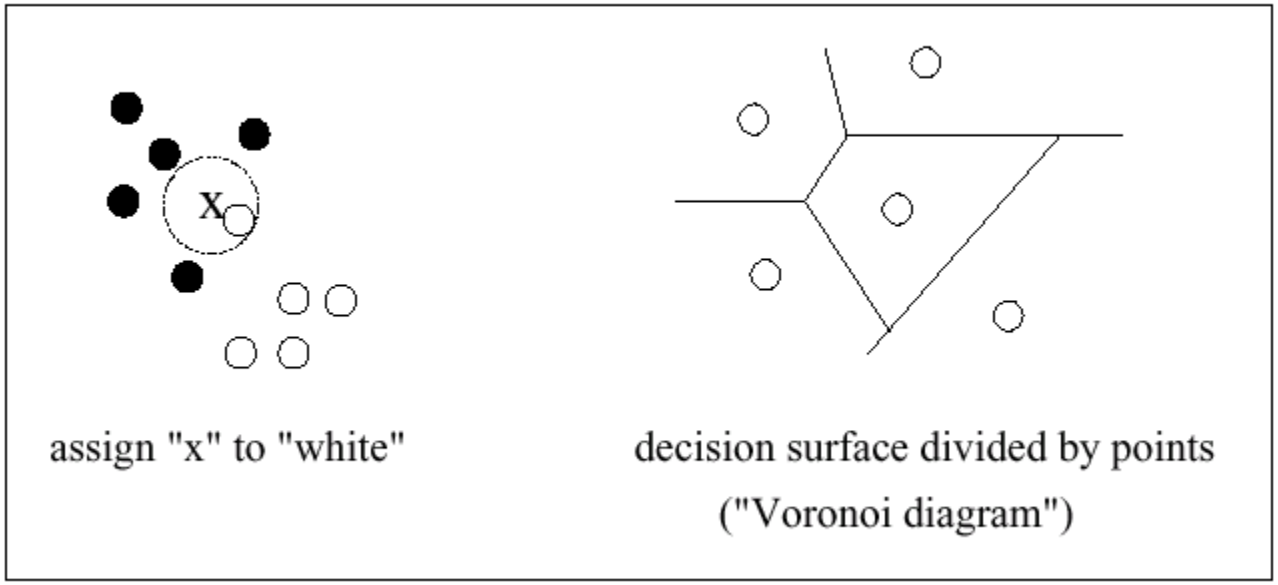
(emphasizing the “k-Nearest Neighbor approach” and Naïve Bayes approach)

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)

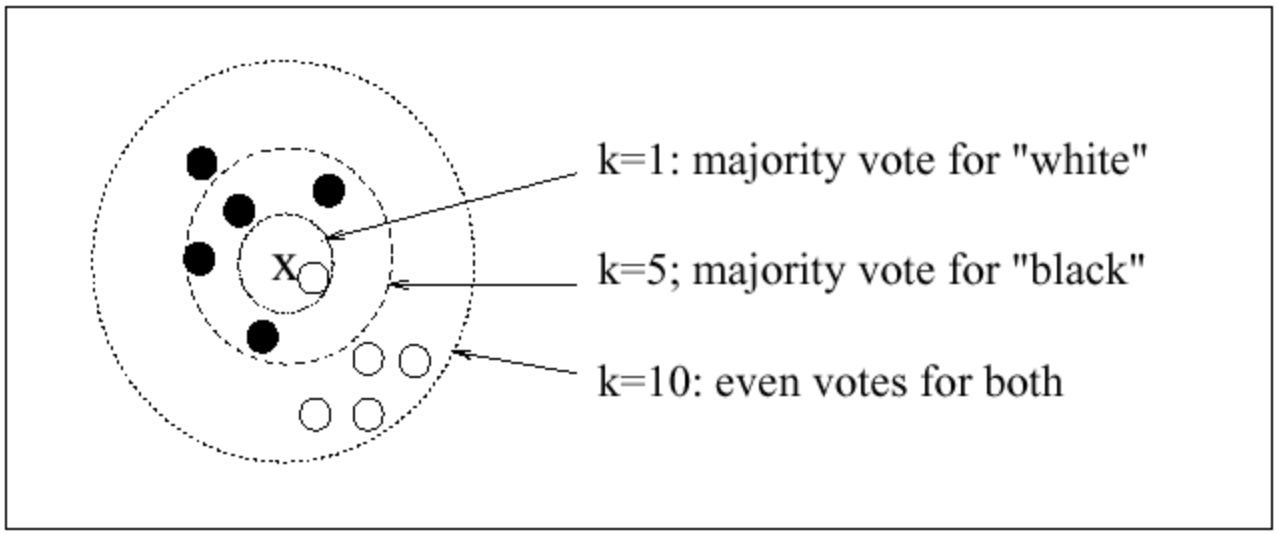
1-NN: assign "x" (new point) to the class of it nearest neighbor



assign "x" to "white"

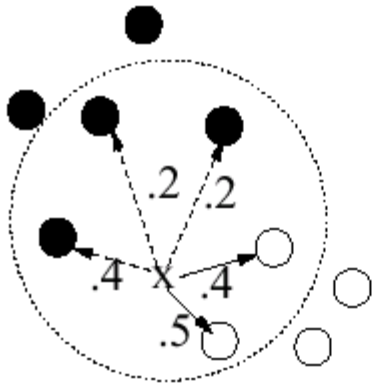
decision surface divided by points ("Voronoi diagram")

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 then 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)

AUT

NB Classifier parameter estimation

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots..x_n \mid c_j)$
  - 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

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

- 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 occurences 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

AUT

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 ti-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…

AUT

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

F_1 = F-value or F-score

A = Accuracy

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

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

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

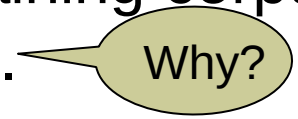
$$A = \frac{TP+TN}{TP+FP+FN+TN}$$

What about more than 2 classes?

AUT

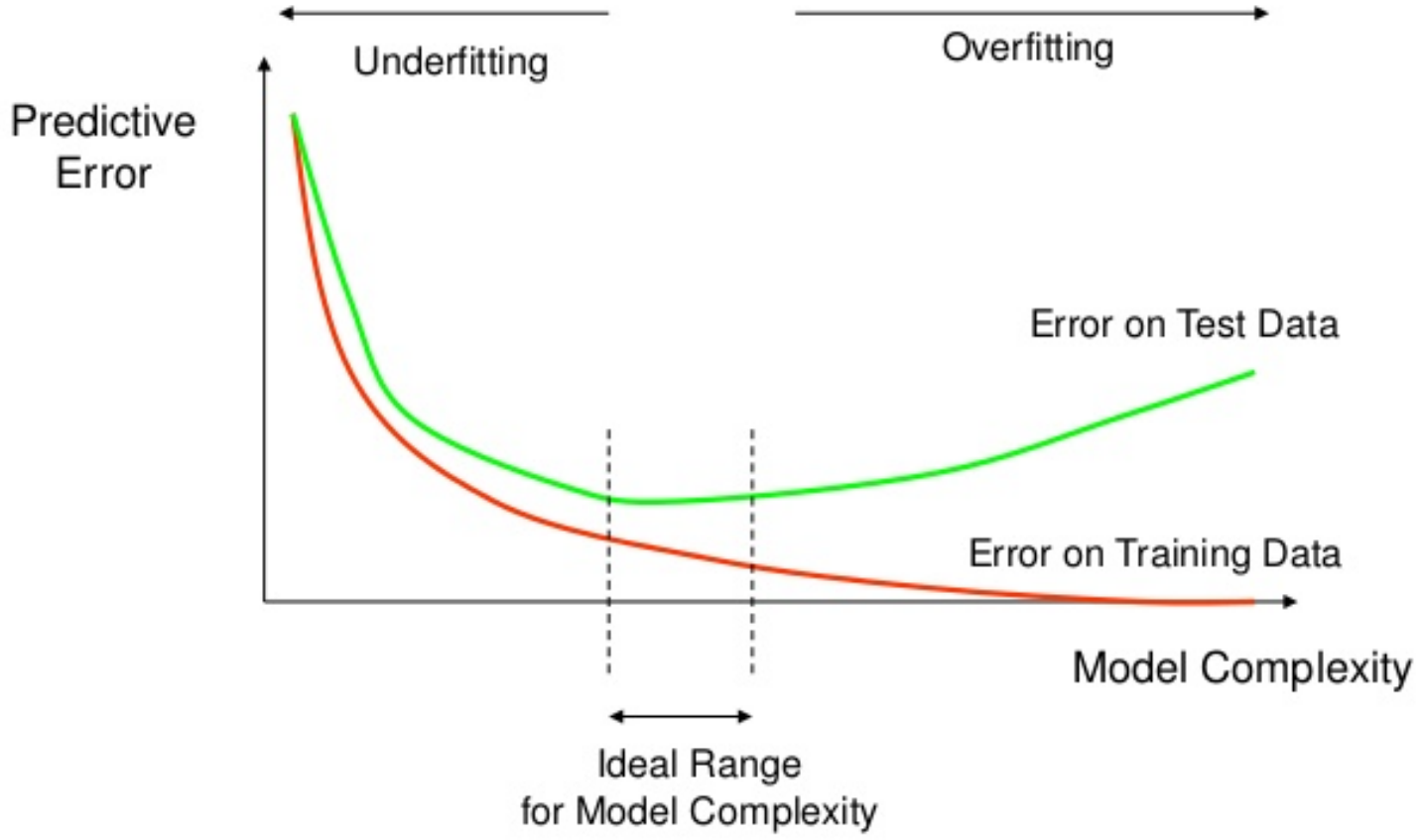
Evaluating Results

- Results on training corpus might not be mirrored in the real world.
- 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.



AUT

Overfitting/Underfitting



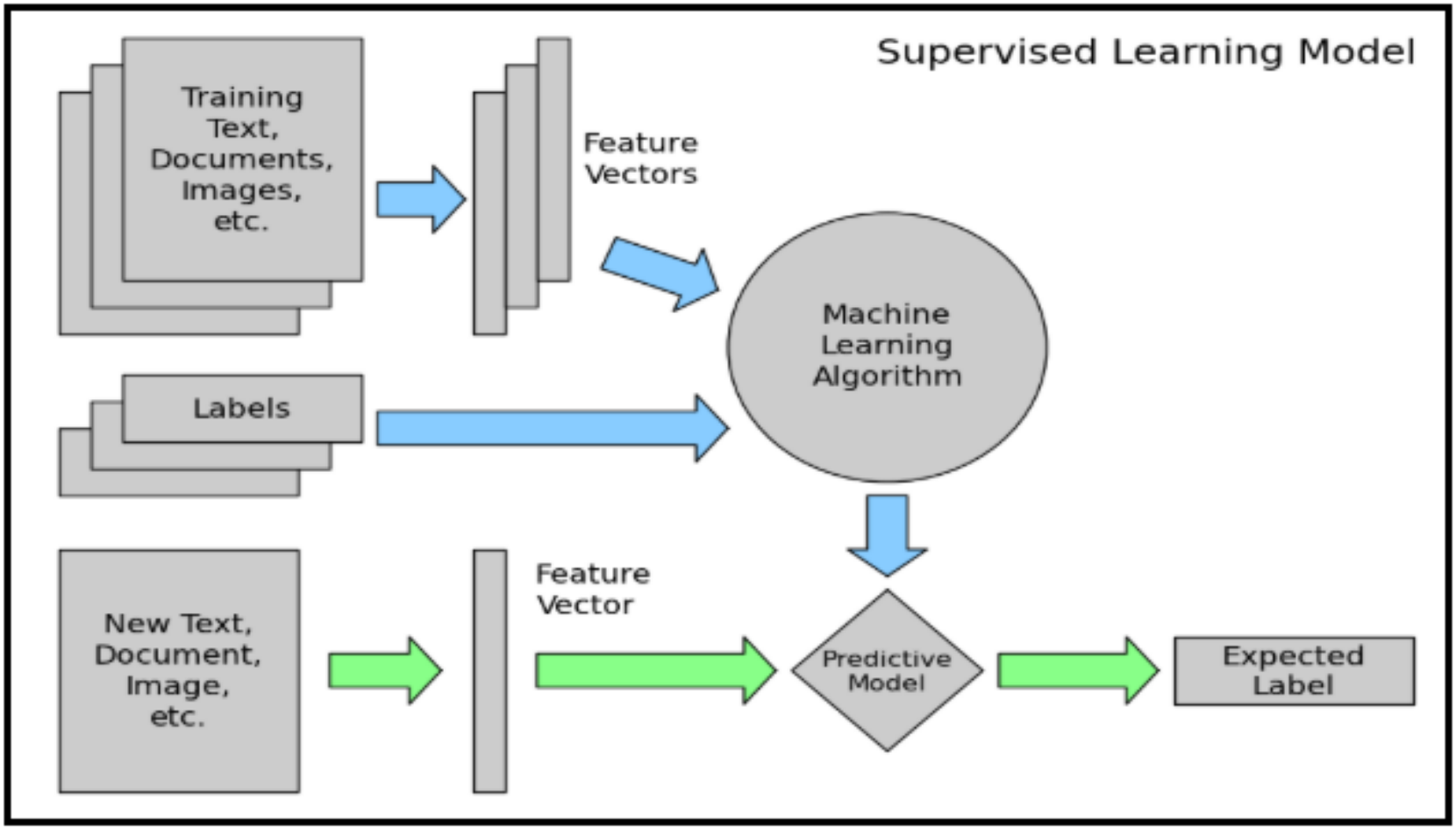
Baseline performance

- **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

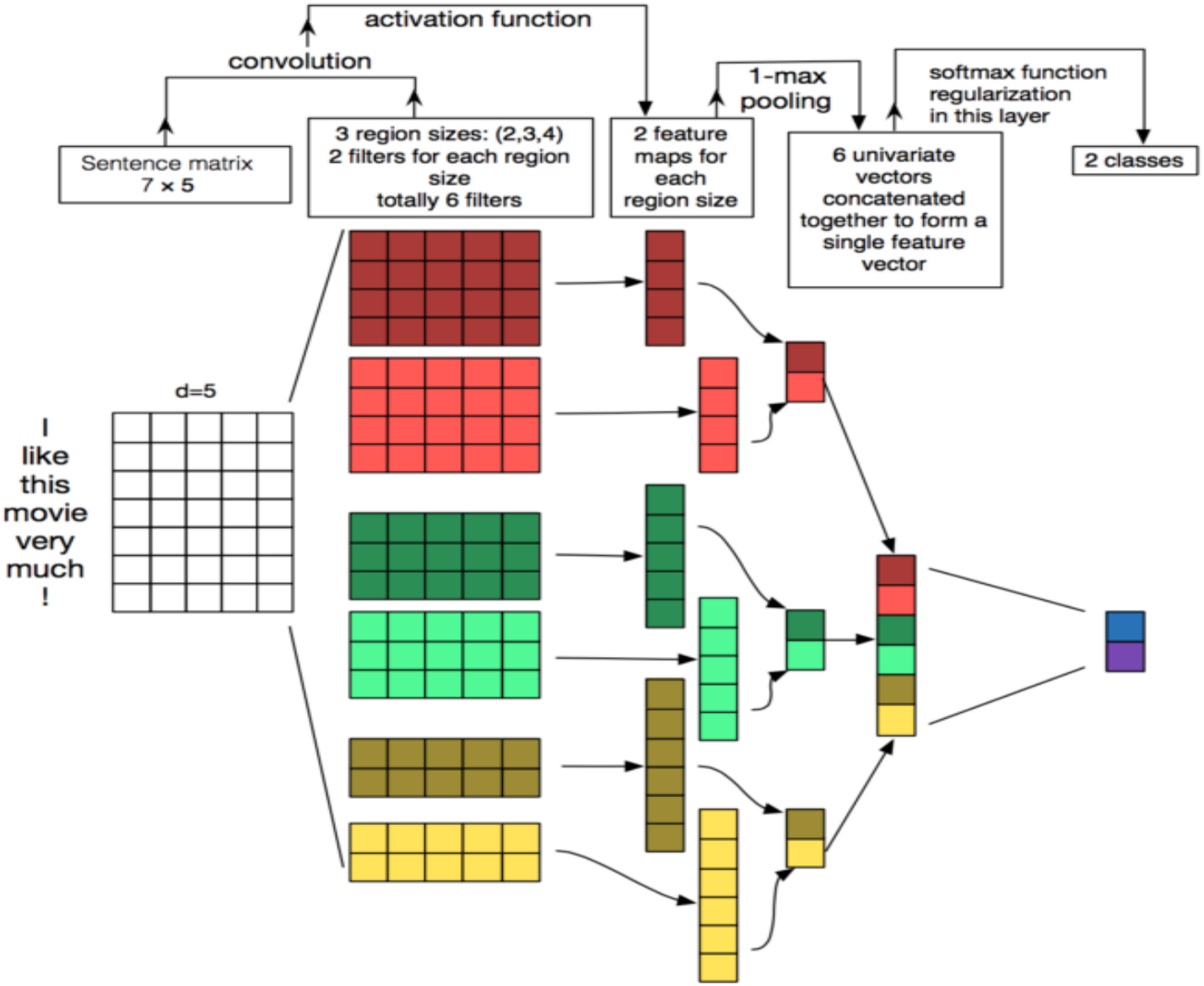
AUT

Deep Learning

Classification using Deep Learning

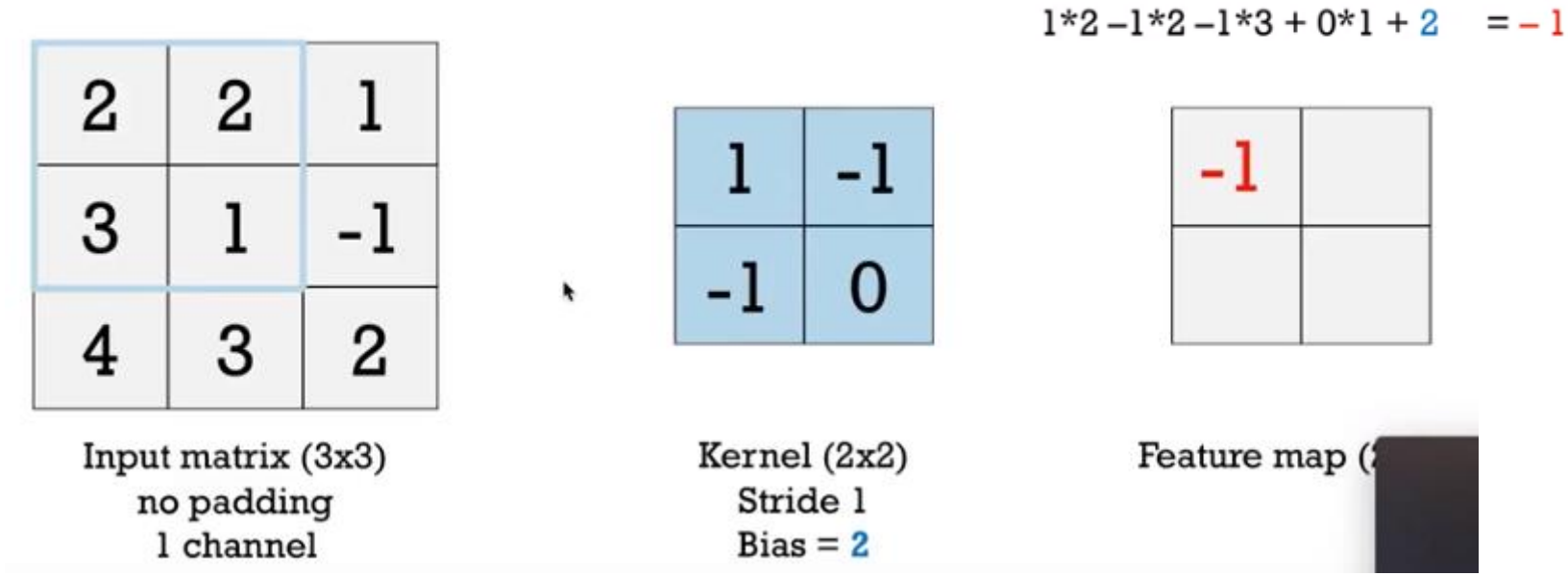


Convolutional Neural Network (CNN)



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**.



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.

$1*2 - 1*2 - 1*3 + 0*1 + 2 = -1$
$1*2 - 1*2 - 1*1 + 0*-1 + 2. = 2$
$1*3 - 1*1 - 1*4 + 0*3 + 2 = 0$
$1*1 - (-1)*1 - 1*3 + 0*2 + 2 = 1$

2	2	1
3	1	-1
4	3	2

Input matrix (3x3)
no padding

1	-1
-1	0

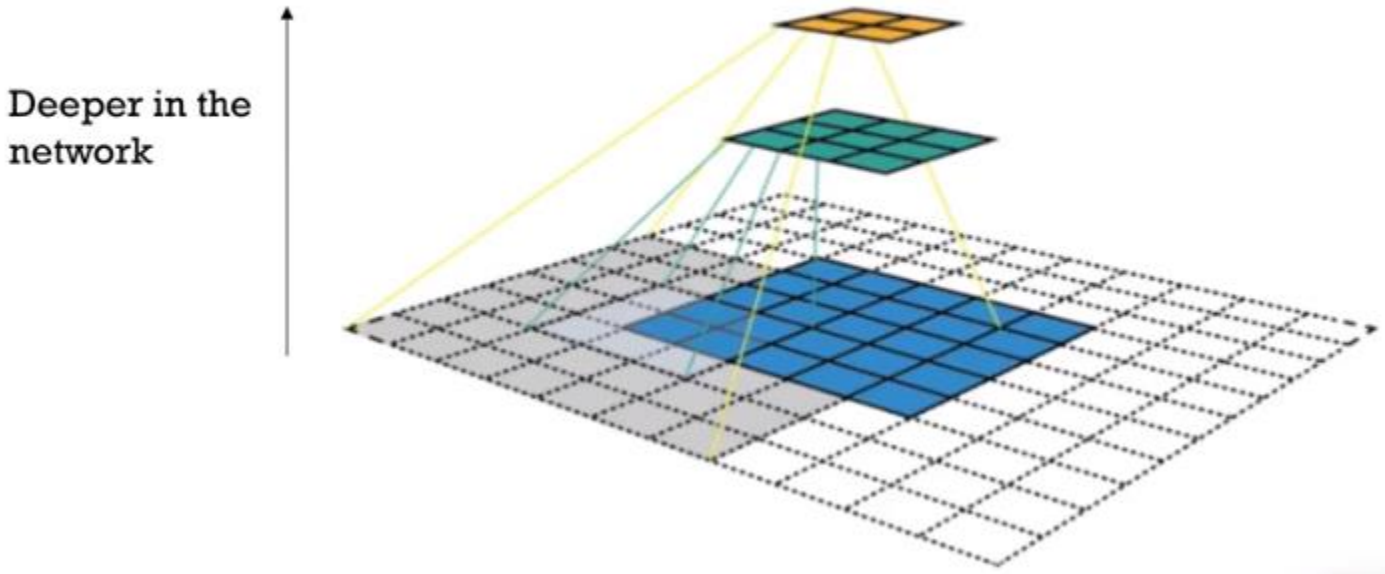
Kernel (2x2)
Stride 1

-1	2
0	1

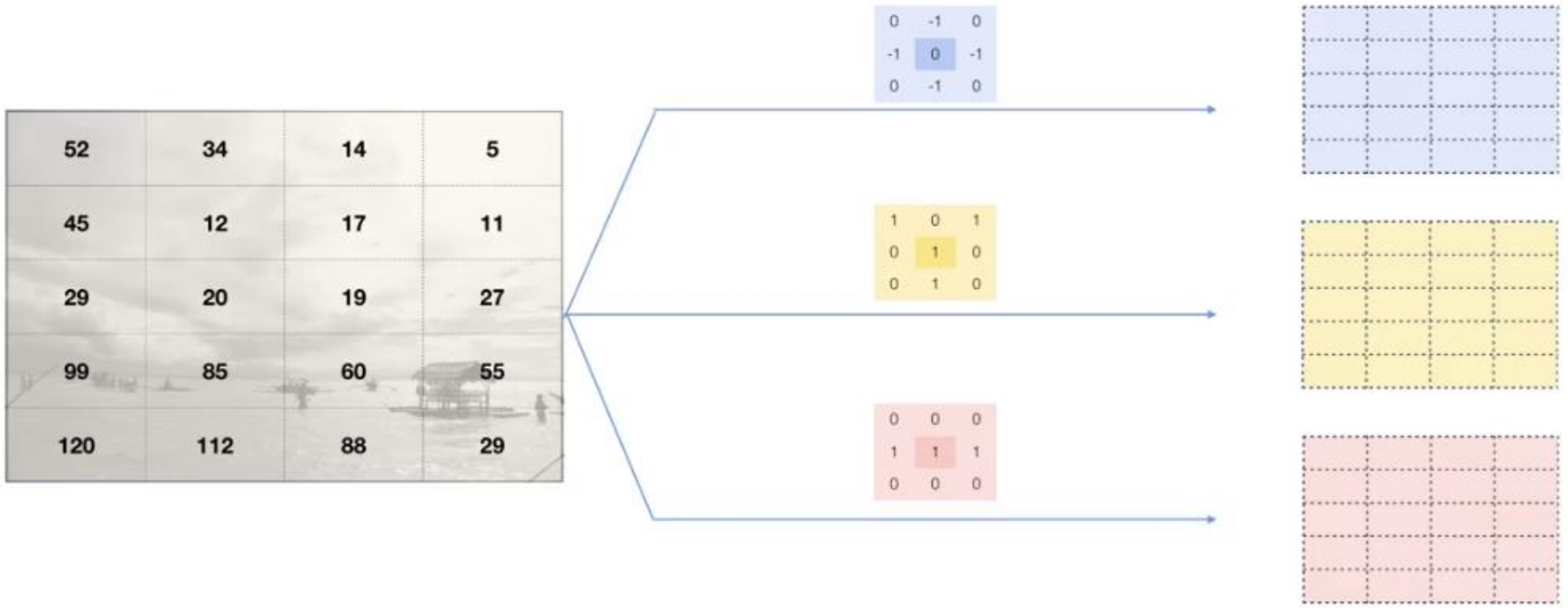
Feature map (

Convolution to reduce dimensions

- To prevent combinatorial explosion.

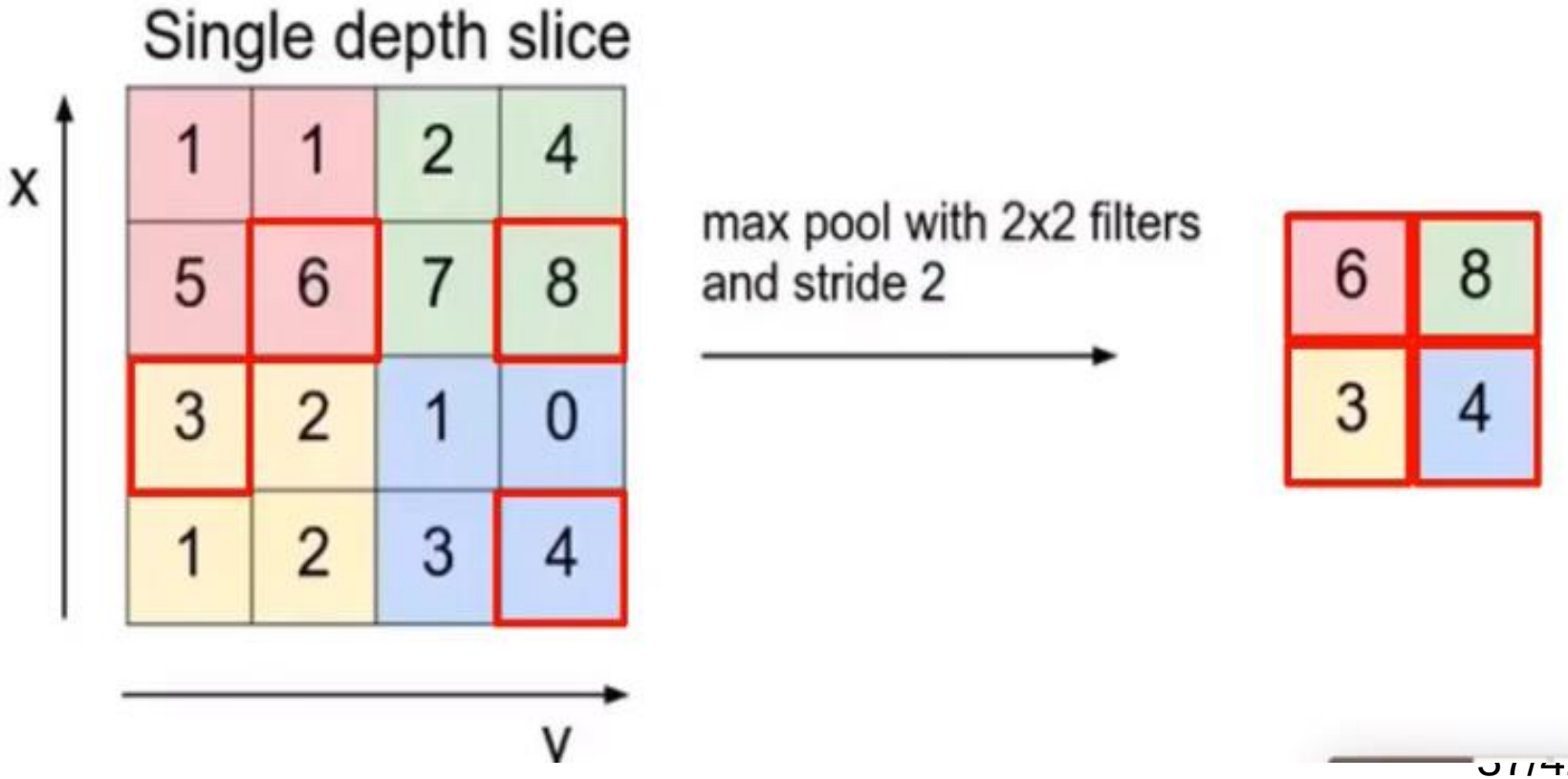


3 convolutions with different kernels

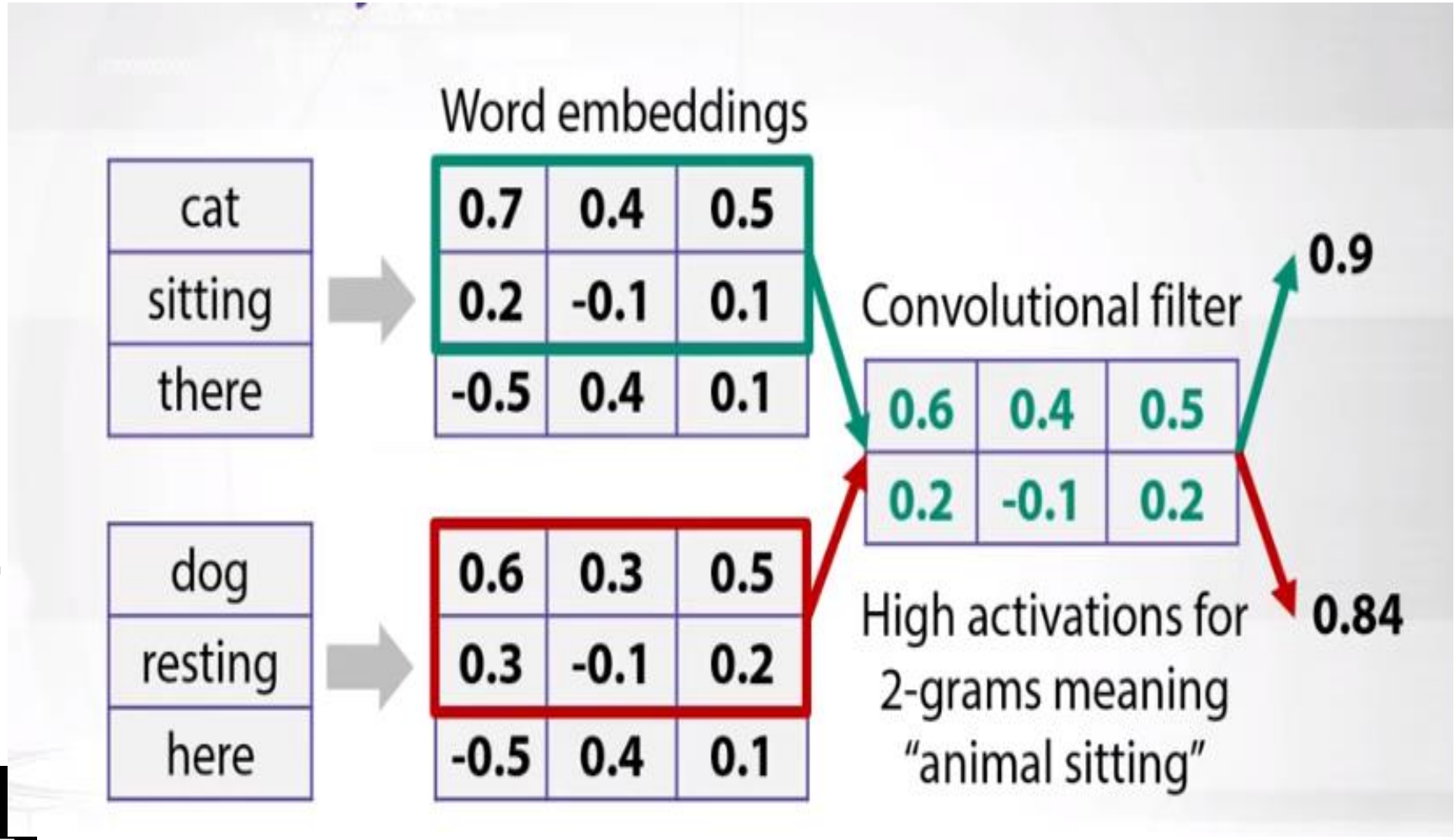


Max pooling

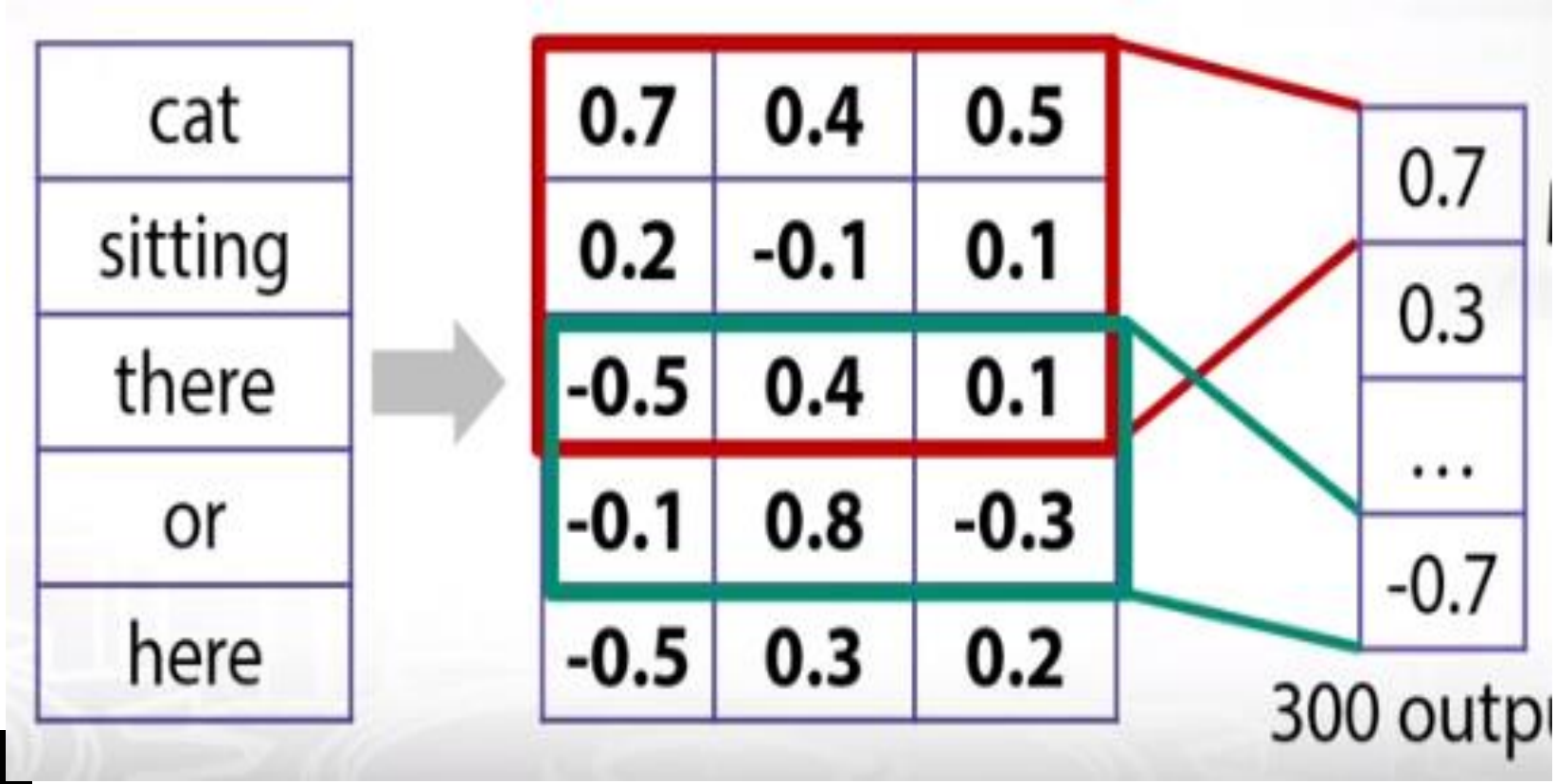
- Picks the biggest number from a window/region



Convolve to lower dimensional matrix



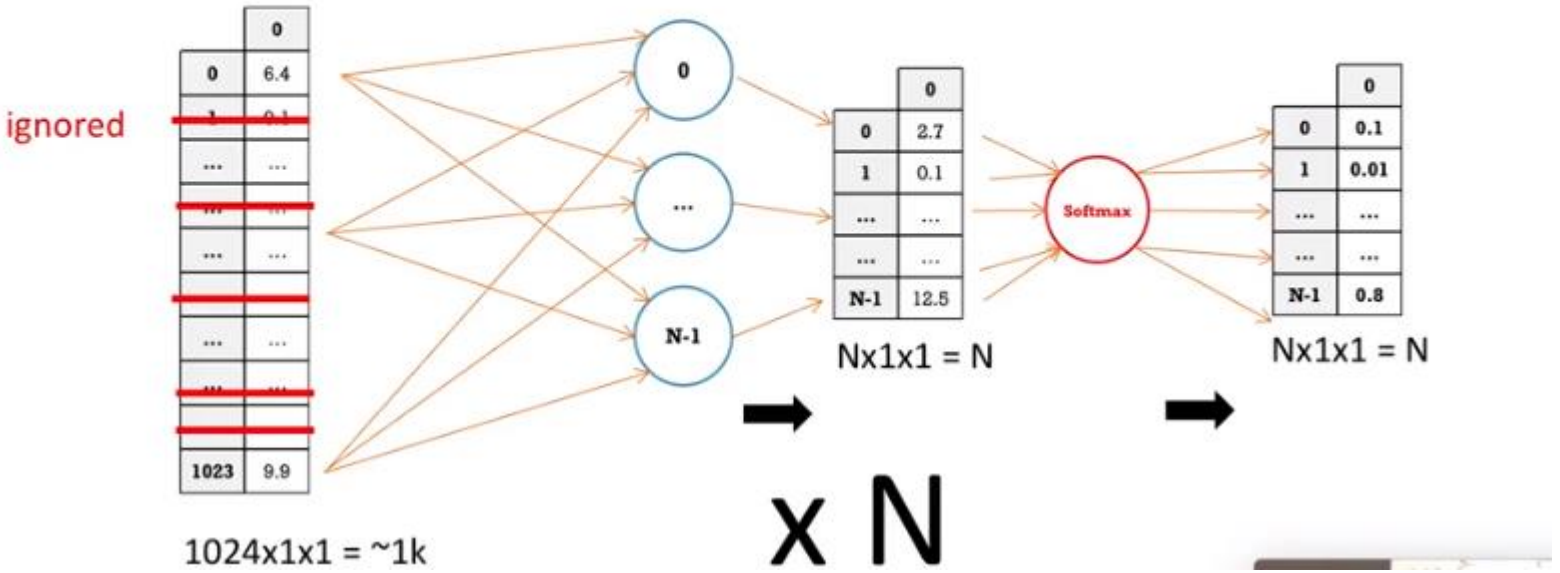
Max pooling


cat
sitting
there
or
here
0.7 0.4 0.5
0.2 -0.1 0.1
-0.5 0.4 0.1
-0.1 0.8 -0.3
-0.5 0.3 0.2
0.7
0.3
...
-0.7
300 outp

Fully connected neural net at the end.

- Flattened matrix, Dropout, softmax for n categories

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



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/

AUT

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

AUT

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.

AUT

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

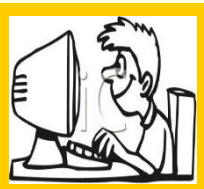
<PER>**Mr John Minto** </PER> taught COMP814 in <DATE>**semester 2 2023**</DATE>.
<PER>**Mr John Minto** </PER> also stood for the elections in <LOC>**Auckland**</LOC>.
<ORG>**Fisher and Pykal** </ORG> bought <ORG>**Haier**</ORG>.

- 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:
  - …

AUT

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

AUT

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

AUT

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

AUT

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}[-.\ ()]){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

AUT

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

AUT

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*

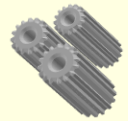
  May the *person* vs. May the *month*

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

AUT

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, …

AUT

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

$$\mathrm{x} = (x_1, x_2, \ldots \mathrm{x}_n)$$

$$\mathrm{y} = (y_1, y_2, \ldots y_n)$$

calculate the Euclidean distance as

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

AUT

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:

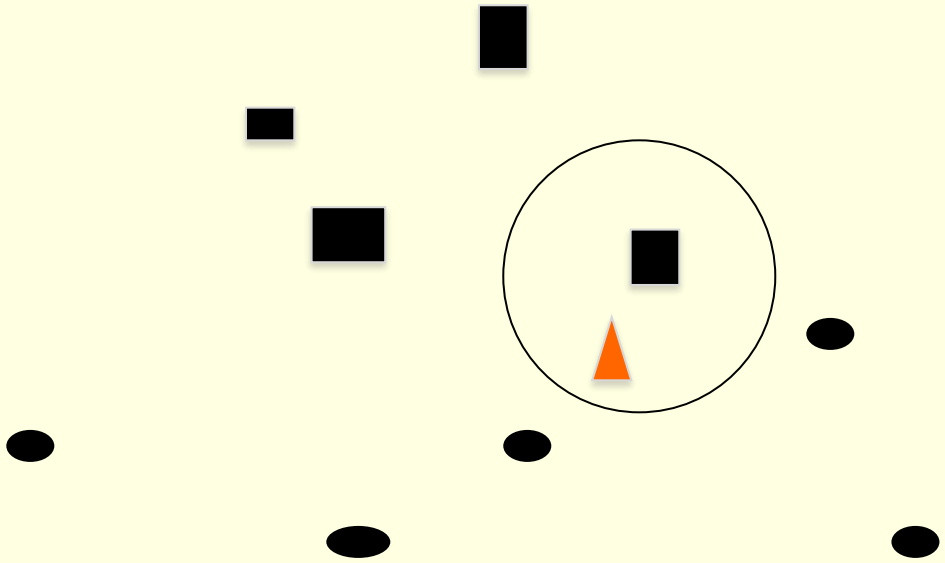
$$d(JK, NP) = \sqrt[2]{(1^2 + 0 + 1^2 + 1^2)} = 1.73$$

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

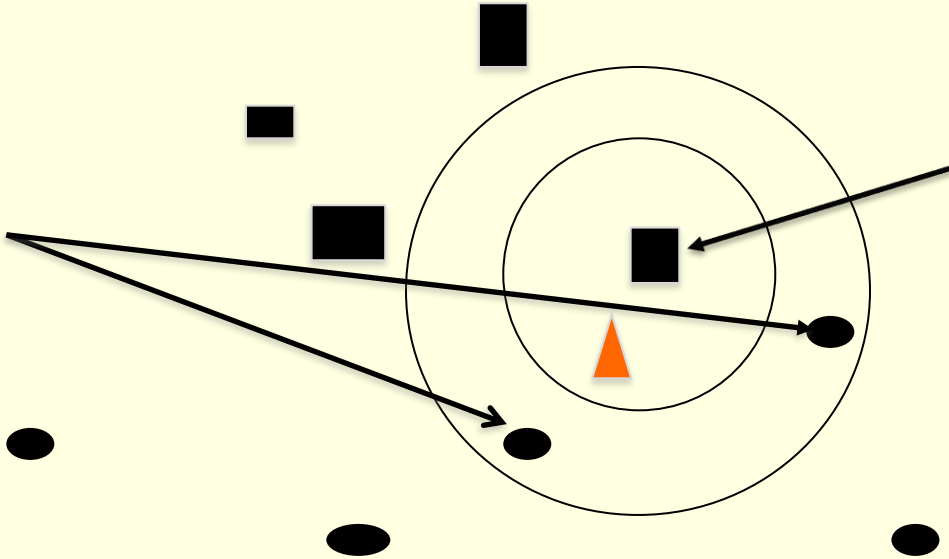
$$d(JK, PC) = \sqrt[2]{(0 + 0 + 0 + 0)} = 0$$

AUT

1-Nearest Neighbor



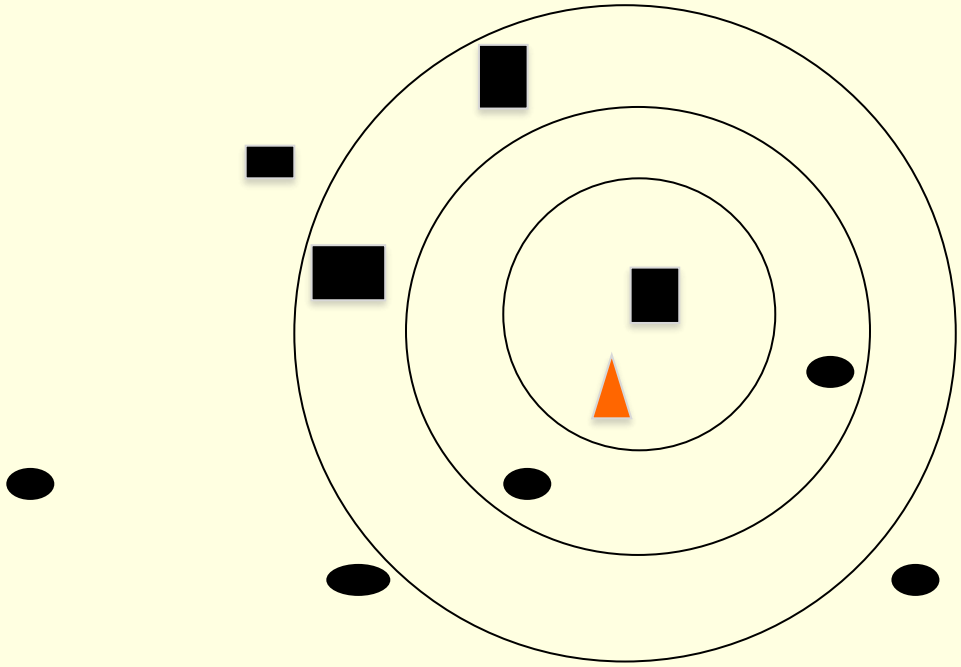
3-Nearest Neighbor



choose the category of the majority of the neighbors

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

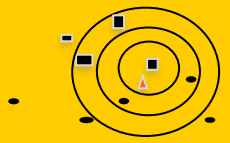
5-Nearest Neighbor



the value of *k* is typically odd to avoid ties

k Nearest Neighbours

Pros

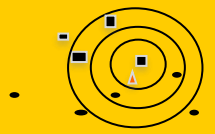


+ robust

+ 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

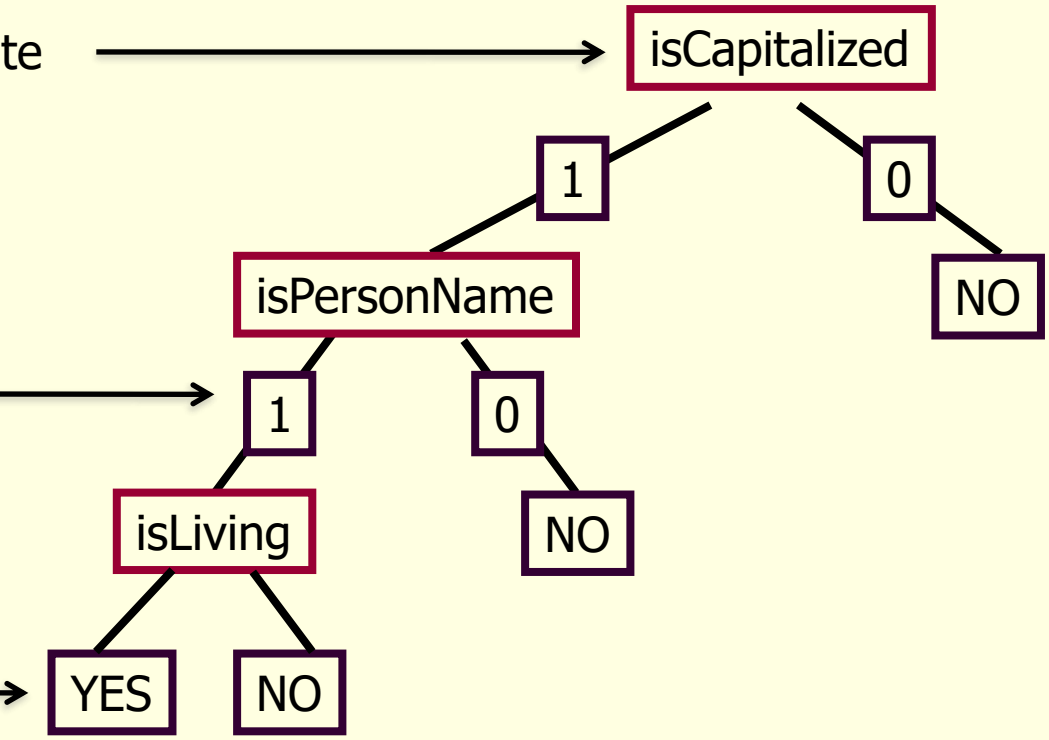
An Example

	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,…,+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)**
  - *initial-caps* *all-caps* *all-digits*
  - *roman-number* *contains-dots* *contains-hyphen*
  - *acronym* *lonely-initial* *punctuation-mark*
  - *single-char* *functional-word** *URL*
- **Word-Type Patterns:**
  - *functional* *lowercased* *quote*
  - *capitalized* *punctuation mark* *other*
- **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 *

AUT

Features for NE Classificaton (4)

- **Contextual**
  - current word W_0
  - words around W_0 in [-3,…,+3] window
- **Part-of-speech tag** (when available)
- **Bag-of-Words**
  - words in [-5,…,+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

AUT

Machine Learning NER

Adam_B Smith_I works_O for_O IBM_B ,_O London_B ._O

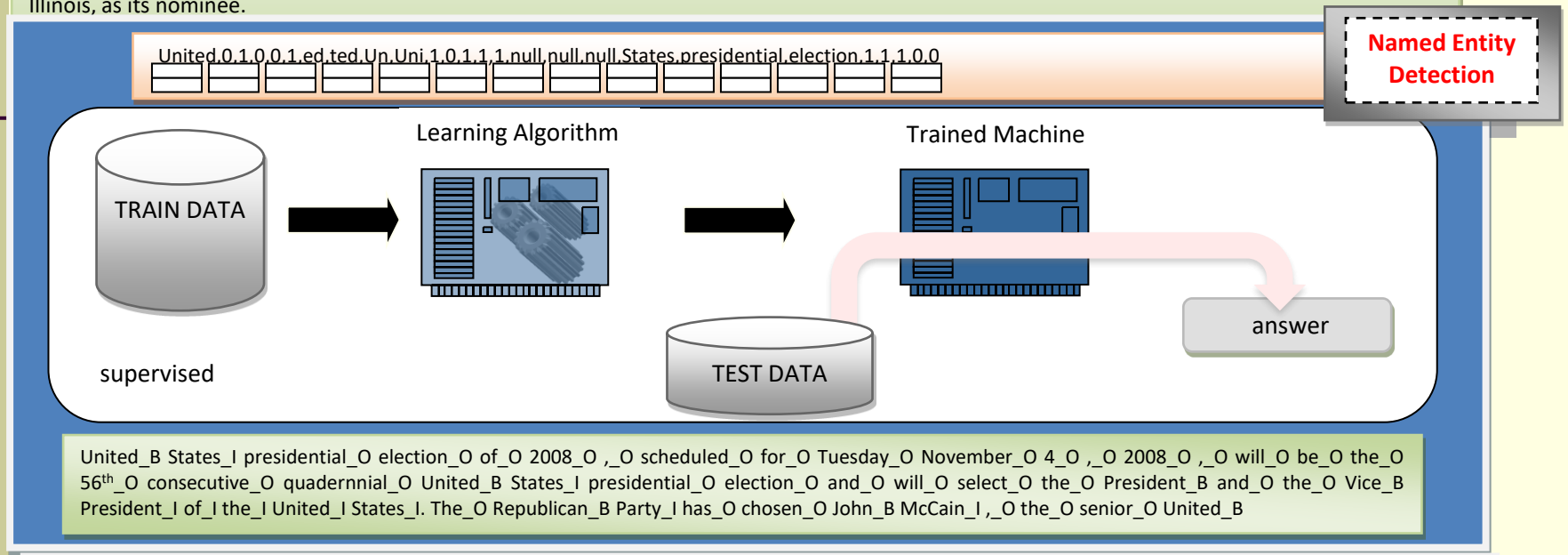
- **NED(Named Entity Detection)**
  - Identify named entities using BIO tags
  - B beginning of an entity
  - I continues the entity
  - O word outside the entity

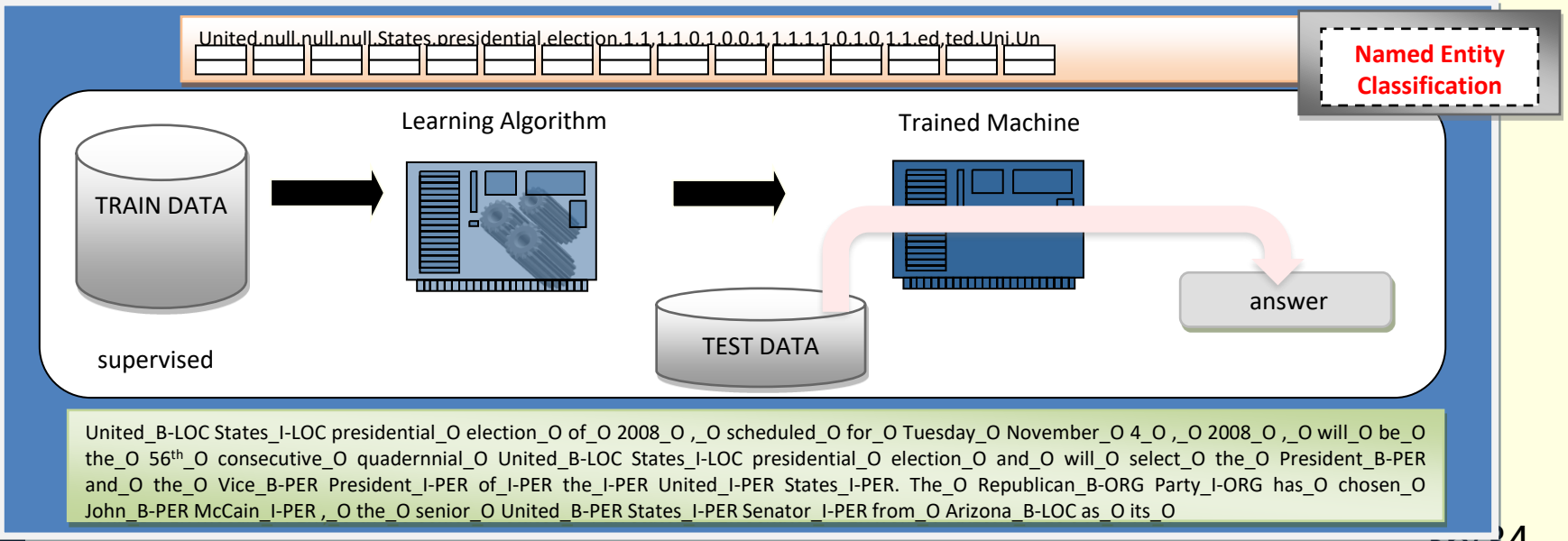
Machine Learning NER

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.





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

Demo Code

deptokens.csv

sampleText2.txt