INTRODUCTION TO DATA SCIENCE

JMCT (SLIDES BY JOHN P DICKERSON)

Lecture #14 – 10/15/2020
Lecture #15 – 10/20/2020

CMSC320
Tuesdays & Thursdays
5:00pm – 6:15pm
(… or anytime on the Internet)

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

PRECURSOR TO NATURAL LANGUAGE PROCESSING

Mechanical translation **started in the 1930s**

- Largely based on dictionary lookups

Georgetown-IBM Experiment:

- Translated 60 Russian sentences to English

- Fairly basic system behind the scenes

- Highly publicized, system ended up spectacularly failing

Funding dried up; not much research in "mechanical translation" until the 1980s …

STATISTICAL NATURAL LANGUAGE PROCESSING

Pre-1980s: primarily based on sets of hand-tuned rules

Post-1980s: introduction of machine learning to NLP

- Initially, decision trees learned what-if rules automatically

- Then, hidden Markov models (HMMs) were used for part of speech (POS) tagging

- Explosion of statistical models for language

- Recent work focuses on purely unsupervised or semi-supervised learning of models

We'll cover some of this in the machine learning lectures!

IBM Watson

NLP IN DATA SCIENCE

In Mini-Project #1, you used `requests` and `BeautifulSoup` to scrape structured data from the web

Lots of data come as unstructured free text: ???????????

- Facebook posts

- Amazon Reviews

- Wikileaks dump

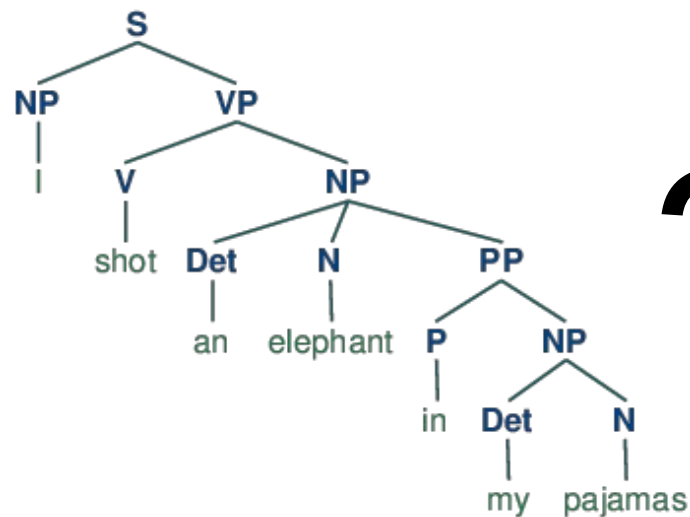**Data science: want to get some meaningful information from unstructured text**

- Need to get some level of understanding what the text says

UNDERSTANDING LANGUAGE IS HARD

One morning I shot an elephant in my pajamas.

How he got into my pajamas, I'll never know.

Groucho Marx

UNDERSTANDING LANGUAGE IS HARD

The Winograd Schema Challenge:

• Proposed by Levesque as a complement to the Turing Test

Formally, need to pick out the antecedent of an ambiguous pronoun:

The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.

Terry Winograd

Levesque argues that understanding such sentences requires more than NLP, but also commonsense reasoning and deep contextual reasoning

UNDERSTANDING LANGUAGE IS HARD?

> I haven't played it that much yet, but it's shaping to be one of the greatest games ever made! It exudes beauty in every single pixel of it. It's a masterpiece. 10/10
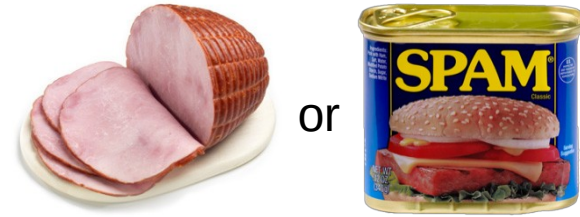
fabchan, March 3, 2017, Metacritic

> a horrible stupid game,it's like 5 years ago game,900p 20~30f, i don't play this **** anymore it's like someone give me a **** to play ,no this time sorry,so Nintendo go f yourself pls

Nsucks7752, March 6, 2017, Metacritic

Perhaps we can get some signal (in this case, sentiment) without truly understanding the text …

"SOME SIGNAL"

Possible signals ?????????

President Trump during a meeting in the Roosevelt Room of the White House last week. Al Drago/The New York Times

"SOME SIGNAL"

What type of article is this?

- Sports

- Political

- Dark comedy

What entities are covered?

- And are they covered with positive or negative sentiment?

Possible signals ????????

ASIDE: TERMINOLOGY

Documents: groups of free text

- Actual documents (NYT article, journal paper)

- Entries in a table

Corpus: a collection of documents

Terms: individual words

- Separated by whitespace or punctuation

NLP TASKS

Syntax: refers to the grammatical structure of language

- The rules via which one forms sentences/expressions

Semantics: the study of meaning of language

John is rectangular and a rainbow.

- Syntactically correct

- Semantically meaningless

SYNTAX

Tokenization

- Splitting sentences into tokens

Lemmatization/Stemming

- Turning "organizing" and "organized" into "organiz"

Morphological Segmentation

- How words are formed, and relationships of different parts
- Easy for English, but other languages are difficult

Part-of-speech (POS) Tagging

- Determine whether a word is a noun/adverb/verb etc.

Parsing

- Create a "parse tree" for a sentence

SEMANTICS: INFORMATION EXTRACTION

What is IE?

Unstructured Web Text → **Structured Sequences**

The second sign of the Zodiac is Taurus.

Strokes are the third most common cause of death in America today.

No study would be complete without mentioning the largest rodent in the world, the Capybara.

Sign of the Zodiac:
1. Aries
2. Taurus
3. Gemini...

Most Common Cause of Death in America:
1. Heart Disease
2. Cancer
3. Stroke...

Largest rodent in the world:
1. Capybara
2. Beaver
3. Patagonian Cavies

SEMANTICS: NAMED ENTITY RECOGNITION

Identifying key entities in text

In 1917, Einstein applied the general theory of relativity to model the large-scale structure of the universe. He was visiting the United States when Adolf Hitler came to power in 1933 and did not go back to Germany, where he had been a professor at the Berlin Academy of Sciences. He settled in the U.S., becoming an American citizen in 1940. On the eve of World War II, he endorsed a letter to President Franklin D. Roosevelt alerting him to the potential development of "extremely powerful bombs of a new type" and recommending that the U.S. begin similar research. This eventually led to what would become the Manhattan Project. Einstein supported defending the Allied forces, but largely denounced using the new discovery of nuclear fission as a weapon. Later, with the British philosopher Bertrand Russell, Einstein signed the Russell-Einstein Manifesto, which highlighted the danger of nuclear weapons. Einstein was affiliated with the Institute for Advanced Study in Princeton, New Jersey, until his death in 1955.

Tag colours:
LOCATION TIME PERSON ORGANIZATION MONEY PERCENT DATE

SEMANTICS: SENTIMENT ANALYSIS

Deciding if reviews/opinions are positive or negative

Heavily used by ad industry today

⭐☆☆☆☆ **An extremely versatile machine!**, November 22, 2006

By **Dr. Nickolas E. Jorgensen "njorgens3"**

This review is from: Cuisinart DGB-600BC Grind & Brew, Brushed Chrome (Kitchen)

This coffee-maker does so much! It makes weak, watery coffee! It grinds beans if you want it to! It inexplicably floods the entire counter with half-brewed coffee when you aren't looking! Perhaps it could be used to irrigate crops... It is time-consuming to clean, but in fairness I should also point out that the stainless-steel thermal carafe is a durable item that has withstood being hurled onto the floor in rage several times. And if all these features weren't enough, it's pretty expensive too. If faced with the choice between having a car door repeatedly slamming into my genitalia and buying this coffee-maker, I'd unhesitatingly choose the Cuisinart! The coffee would be lousy, but at least I could still have children...

SEMANTICS: MACHINE TRANSLATION

Translating from one language to another

Simple substitution of words doesn't work very well
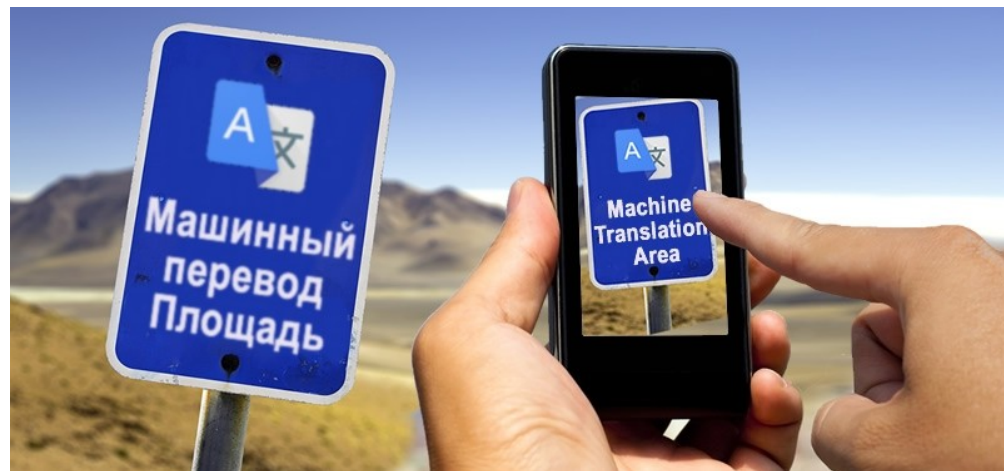
(Image: Google)

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

SEMANTICS: MACHINE TRANSLATION

0> how long before the next flight to Alice Springs?

1> wie lang vor dem folgenden Flug zu Alice Springs?

2> how long before the following flight to Alice jump?

3> wie lang vor dem folgenden Flug zu Alice springen Sie?

4> how long before the following flight to Alice do you jump?

5> wie lang, bevor der folgende Flug zu Alice tun, Sie springen?

6> how long, before the following flight to Alice does, do you jump?

7> wie lang bevor der folgende Flug zu Alice tut, tun Sie springen?

8> how long before the following flight to Alice does, do you jump?

9> wie lang, bevor der folgende Flug zu Alice tut, tun Sie springen?

10> how long, before the following flight does to Alice, do do you jump?

11> wie lang bevor der folgende Flug zu Alice tut, Sie tun Sprung?

12> how long before the following flight does leap to Alice, does you?

22

SEMANTICS: QUESTION ANSWERING

Answer questions posed a user with specific answers

SEMANTICS: QUESTION ANSWERING

Answer questions posed a user with specific answers

Factoid questions

- Who wrote "The Universal Declaration of Human Rights"?
- How many calories are there in two slices of apple pie?
- What is the average age of the onset of autism?
- Where is Apple Computer based?

Complex (narrative) questions:

- In children with an acute febrile illness, what is the efficacy of acetaminophen in reducing fever?
- What do scholars think about Jefferson's position on dealing with pirates?

SEMANTICS: SPOKEN DIALOGUE SYSTEMS
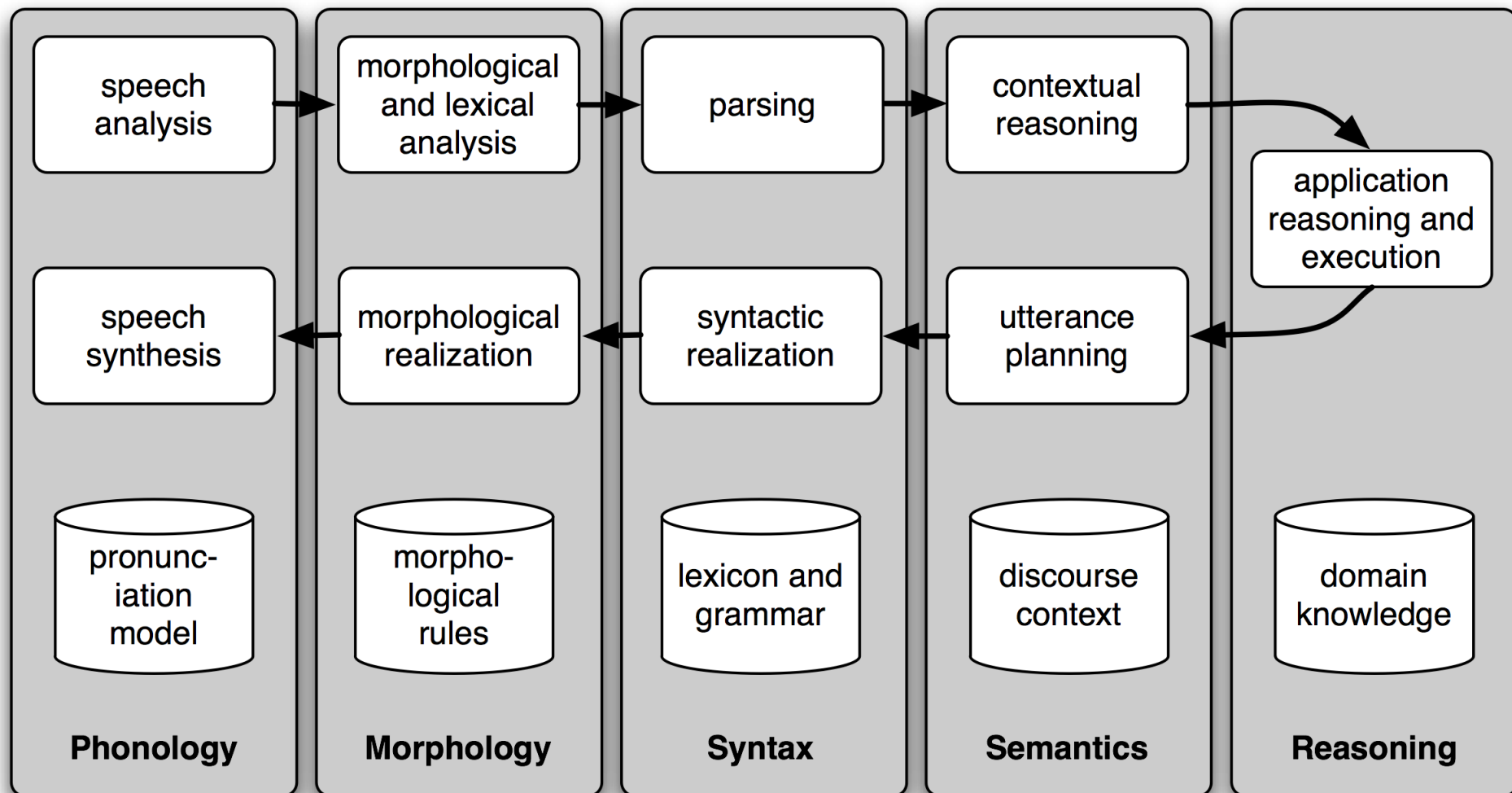
SEMANTICS: TEXTUAL ENTAILMENT

Given two text fragments, determine if one being true entails the other, entails the other's negation, or allows the other to be either true or false

TEXT	HYPOTHESIS	ENTAILMENT
Eyeing the huge market potential, currently led by Google, Yahoo took over search company Overture Services Inc last year.	• Yahoo bought Overture.	• TRUE
Microsoft's rival Sun Microsystems Inc. bought Star Office last month and plans to boost its development as a Web-based device running over the Net on personal computers and Internet appliances.	• Microsoft bought Star Office.	• FALSE
The National Institute for Psychobiology in Israel was established in May 1971 as the Israel Center for Psychobiology by Prof. Joel.	• Israel was established in May 1971.	• FALSE

SEMANTICS: DOCUMENT SUMMARIZATION

Quite a few tools out there today… e.g., SMMRY

OTHER TASKS

Speech Recognition

Caption Generation

Natural Language Generation

Optical Character Recognition

Word Sense Disambiguation

- serve: help with food or drink; hold an office; put ball into play

…

Doing all of these for many different languages
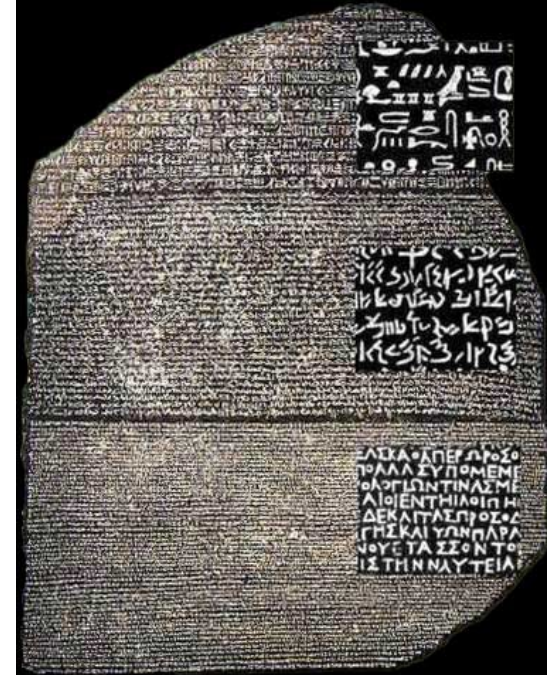
SEMANTICS: TEXT CLASSIFICATION

Is it spam?

Who wrote this paper? (Author identification)

- https://en.wikipedia.org/wiki/The_Federalist_Papers#Authorship

- https://www.uwgb.edu/dutchs/pseudosc/hidncode.htm

¡Identificación del idioma!

Sentiment analysis

What type of document is this?

When was this document written?

Readability assessment
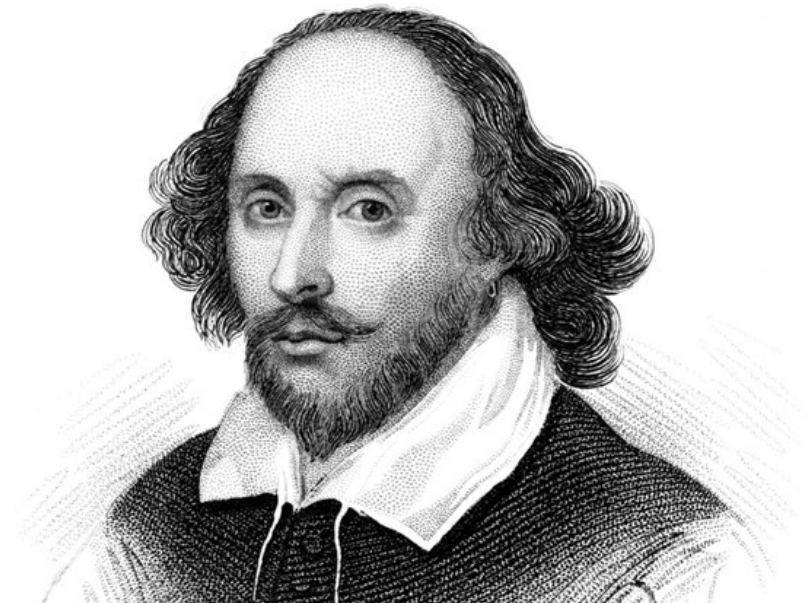
TEXT CLASSIFICATION

Input:

- A document w

- A set of classes $Y = \{y_1, y_2, \ldots, y_J\}$

Output:

- A predicted class $y \in Y$

(You will spend much more time on classification problems throughout the program, this is just a light intro!)

TEXT CLASSIFICATION

Hand-coded rules based on combinations of terms (and possibly other context)

If email w:

- Sent from a DNSBL (DNS blacklist)   **OR**

- Contains "Nigerian prince"   **OR**

- Contains URL with Unicode   **OR** …

Then: y_w = spam

Pros: ?????????

- Domain expertise, human-understandable

Cons: ?????????

- Brittle, expensive to maintain, overly conservative

TEXT CLASSIFICATION

Input:

- A document w

- A set of classes $Y = \{y_1, y_2, \ldots, y_J\}$

- A training set of m hand-labeled documents
    $\{(w_1, y_1), (w_2, y_2), \ldots, (w_m, y_m)\}$

Output:

- A learned classifier $w \rightarrow y$

This is an example of supervised learning

REPRESENTING A DOCUMENT "IN MATH"

Simplest method: **bag of words**
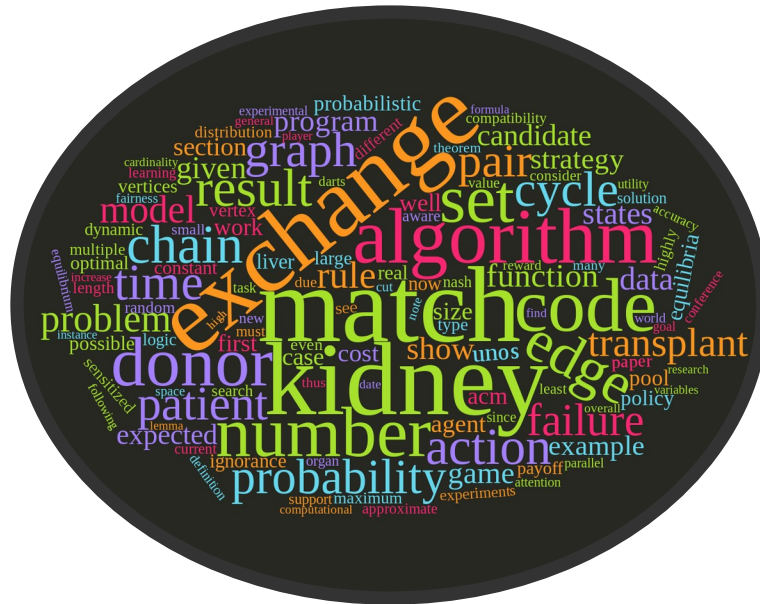
Represent each document as a vector of word frequencies

- Order of words does not matter, just #occurrences

BAG OF WORDS EXAMPLE

the quick brown fox jumps over the lazy dog

I am he as you are he as you are me

he said the CMSC320 is 189 more CMSCs than the CMSC131

	the	CMSC320	you	he	I	quick	dog	me	CMSCs	...	than
Document 1	2	0	0	0	0	1	1	0	0		0
Document 2	0	0	2	2	1	0	0	1	0	...	0
Document 3	2	1	0	1	0	0	0	0	1		1

TERM FREQUENCY

Term frequency: the number of times a term appears in a specific document

- tf_{ij}: frequency of word j in document i

This can be the raw count (like in the BOW in the last slide):

- $tf_{ij} \in \{0,1\}$ if word j appears or doesn't appear in doc i

- $\log(1 + tf_{ij})$ – reduce the effect of outliers

- $tf_{ij} / \max_j tf_{ij}$ – normalize by document i's most frequent word

What can we do with this?

- Use as features to learn a classifier w ✉ y …!

DEFINING FEATURES FROM TERM FREQUENCY

Suppose we are classifying if a document was written by The Beatles or not (i.e., binary classification):

- Two classes $y \in Y = \{ 0, 1 \} = \{$ not_beatles, beatles $\}$

Let's use tf$_{ij} \in$ {0,1}, which gives:

	the	CMSC320	you	he	I	quick	dog	me	CMSCs	...	than
$x_1^\top =$	1	0	0	0	0	1	1	0	0		0
$x_2^\top =$	0	0	1	1	1	0	0	1	0	...	0
$x_3^\top =$	1	1	0	1	0	0	0	0	1		1

$y_1 = 0$

$y_2 = 1$

$y_3 = 0$

Then represent documents with a feature function:

$\mathbf{f}(\mathbf{x}, y = \text{not\_beatles} = 0) = \quad [\mathbf{x}^\top, \mathbf{0}^\top, 1]^\top$

$\mathbf{f}(\mathbf{x}, y = \text{beatles} = 1) = \quad [\mathbf{0}^\top, \mathbf{x}^\top, 1]^\top$

LINEAR CLASSIFICATION

We can then define weights *θ* for each feature

> *θ* = { <CMSC320, not_beatles> = +1,
> <CMSC320, beatles> = -1,
> <walrus, not_beatles> = -0.3,
> <walrus, beatles> = +1,
> <the, not_beatles> = 0,
> <the, beatles>, 0, … }

Write weights as vector that aligns with feature mapping

Score *ψ* of an instance *x* and class *y* is the sum of the weights for the features in that class:

> $\boldsymbol{\psi}_{xy} = \Sigma\ \theta_n\ f_n(\mathbf{x},\ y)$
>
> $\quad\quad = \boldsymbol{\theta}^\top\ \mathbf{f}(\mathbf{x},\ y)$

LINEAR CLASSIFICATION

We have a feature function f(x, *y*) and a score $\boldsymbol{\psi}_{xy} = \boldsymbol{\theta}^{\top} f(\mathbf{x}, y)$

And return the class with highest score!

Compute the score of the document for that class

$$\hat{y} = \arg\max_{y} \theta^{\mathsf{T}} \mathbf{f}(\mathbf{x}, y)$$

For each class y ∈ { not_beatles, beatles }

(… and also this whole "linear classifier" thing.)

Where did these weights come from? We'll talk about this in the ML lectures …

EXPLICIT EXAMPLE

We are interested in classifying documents into one of two classes $y \in Y = \{ 0, 1 \} = \{$ hates_cats, likes_cats$\}$

Document 1: I like cats

Document 2: I hate cats

	_	like	hate	cats
$x_1^\top =$	1	1	0	1
$x_2^\top =$	1	0	1	1

$y_1 = ?$

$y_2 = ?$

Now, represent documents with a feature function:

$\mathbf{f}(\mathbf{x}, y = \text{hates\_cats} = 0) = [\mathbf{x}^\top, \mathbf{0}^\top, 1]^\top$

$\mathbf{f}(\mathbf{x}, y = \text{likes\_cats} = 1) = [\mathbf{0}^\top, \mathbf{x}^\top, 1]^\top$

EXPLICIT EXAMPLE

$f(\mathbf{x}, y = 0) = [\mathbf{x}^\top, 0^\top, 1]^\top$
$f(\mathbf{x}, y = 1) = [0^\top, \mathbf{x}^\top, 1]^\top$
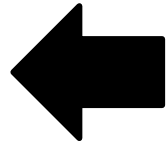
	_	like	hate	cats
$x_1^\top =$	1	1	0	1
$x_2^\top =$	1	0	1	1

$y_1 = ?$

$y_2 = ?$

y=0: hates_cats *y=1: likes_cats* (1)

	_	like	hate	cats	_	like	hate	cats	⋮
$f(\mathbf{x_1}, y = \text{hates\_cats} = 0) =$	1	1	0	1	0	0	0	0	1
$f(\mathbf{x_1}, y = \text{likes\_cats} = 1) =$	0	0	0	0	1	1	0	1	1
$f(\mathbf{x_2}, y = \text{hates\_cats} = 0) =$	1	0	1	1	0	0	0	0	1
$f(\mathbf{x_2}, y = \text{likes\_cats} = 1) =$	0	0	0	0	1	0	1	1	1

EXPLICIT EXAMPLE

Now, assume we have weights θ for each feature

θ = { <I, hates_cats> = 0, <I, likes_cats> = 0,

<like, hates_cats> = -1, <like, likes_cats> = +1,

<hate, hates_cats> = +1, <hate, likes_cats> = -1,

<cats, hates_cats> = -0.1, <cats, likes_cats = +0.5>}

Write weights as vector that aligns with feature mapping:

	y=0: hates_cats				*y=1: likes_cats*			(1)
Parameter vector $\theta^\top$ =	0	-1	1	-0.1	0	1	-1	0.5	1
	I	like	hate	cats	I	like	hate	cats	:
f($\mathbf{x}_1$, y = hates_cats = 0) =	1	1	0	1	0	0	0	0	1
f($\mathbf{x}_1$, y = likes_cats = 1) =	0	0	0	0	1	1	0	1	1
f($\mathbf{x}_2$, y = hates_cats = 0) =	1	0	1	1	0	0	0	0	1
f($\mathbf{x}_2$, y = likes_cats = 1) =	0	0	0	0	1	0	1	1	1

EXPLICIT EXAMPLE

Score $\boldsymbol{\psi}$ of an instance _x_ and class _y_ is the sum of the weights for the features in that class:

$$\boldsymbol{\psi}_{xy} = \Sigma \, \theta_n \, f_n(\mathbf{x}, y)$$

$$= \boldsymbol{\theta}^{\mathrm{T}} \, \mathbf{f}(\mathbf{x}, y)$$

Let's compute $\boldsymbol{\psi}_{x1, y=hates\_cats}$ ...

- $\boldsymbol{\psi}_{x1, y=hates\_cats} = \boldsymbol{\theta}^{\mathrm{T}} \, \mathbf{f}(\mathbf{x_1}, y = \text{hates\_cats} = 0)$

- $= 0*1 + -1*1 + 1*0 + -0.1*1 + 0*0 + 1*0 + -1*0 + 0.5*0 + 1*1$

- $= -1 - 0.1 + 1 = \textbf{-0.1}$

1	I
1	like
0	hate
1	cats
0	I
0	like
0	hate
0	cats
1	–

hates_cats   *likes_cats*   (1)

$\boldsymbol{\theta}^{\mathrm{T}} =$

0	-1	1	-0.1	0	1	-1	0.5	1

●

$f(\mathbf{x_1}, y = 0)$

42

EXPLICIT EXAMPLE

Saving the boring stuff:

- $\boldsymbol{\psi}_{x1,y=hates\_cats} = -0.1$; $\boldsymbol{\psi}_{x1,y=likes\_cats} = +2.5$

  Document 1: I like cats

- $\boldsymbol{\psi}_{x2,y=hates\_cats} = +1.9$; $\boldsymbol{\psi}_{x2,y=likes\_cats} = +0.5$

  Document 2: I hate cats

We want to predict the class of each document:

$$\hat{y} = \arg\max_{y} \theta^{\mathsf{T}} \mathbf{f}(\mathbf{x}, y)$$

Document 1: argmax{ $\boldsymbol{\psi}_{x1,y=hates\_cats}$, $\boldsymbol{\psi}_{x1,y=likes\_cats}$ } ????????

Document 2: argmax{ $\boldsymbol{\psi}_{x2,y=hates\_cats}$, $\boldsymbol{\psi}_{x2,y=likes\_cats}$ } ????????

INVERSE DOCUMENT FREQUENCY

Recall:

- tf_{ij}: frequency of word *j* in document *i*

Any issues with this ??????????

- Term frequency gets overloaded by common words

Inverse Document Frequency (IDF): weight individual words negatively by how frequently they appear in the corpus:

$$\text{idf}_j = \log\left(\frac{\#\text{documents}}{\#\text{documents with word } j}\right)$$

IDF is just defined for a word j, not word/document pair j, i

INVERSE DOCUMENT FREQUENCY

	the	CMSC320	you	he	I	quick	dog	me	CMSCs	...	than
Document 1	2	0	0	0	0	1	1	0	0		0
Document 2	0	0	2	2	1	0	0	1	0	...	0
Document 3	2	1	0	1	0	0	0	0	1		1

$$\text{idf}_{\text{the}} = \log\left(\frac{3}{2}\right) = 0.405 \qquad \text{idf}_{\text{you}} = \log\left(\frac{3}{1}\right) = 1.098$$

$$\text{idf}_{\text{CMSC320}} = \log\left(\frac{3}{1}\right) = 1.098 \qquad \text{idf}_{\text{he}} = \log\left(\frac{3}{2}\right) = 0.405$$

TF-IDF

How do we use the IDF weights?

Term frequency inverse document frequency (TF-IDF):

- TF-IDF score: tf_{ij} x idf_j

	the	CMSC320	you	he	I	quick	dog	me	CMSCs	...	than
Document 1	0.8	0	0	0	0	1.1	1.1	0	0		0
Document 2	0	0	2.2	0.8	1.1	0	0	1.1	0	…	0
Document 3	0.8	1.1	0	0.4	0	0	0	0	1.1		1.1

This ends up working better than raw scores for classification and for computing similarity between documents.

TOKENIZATION

First step towards text processing

For English, just split on non-alphanumerical characters

- Need to deal with cases like: I'm, or France's, or Hewlett-Packard
- Should "San Francisco" be one token or two?

Other languages introduce additional issues

- L'ensemble → one token or two?
- German noun compounds are not segmented
    - Lebensversicherungsgesellschaftsangestellter
- Chinese/Japanese more complicated because of white spaces

OTHER BASIC TERMS

Lemmatization

- Reduce inflections or variant forms to base form
    - am, are, is → be
    - car, cars, car's, cars' → car
- the boy's cars are different colors → the boy car be different color

Morphology/Morphemes

- The small meaningful units that make up words
- Stems: The core meaning-bearing units
- Affixes: Bits and pieces that adhere to stems
    - Often with grammatical functions

STEMMING

Reduce terms to their stems in information retrieval

Stemming is crude chopping of affixes

- language dependent
- e.g., **automate(s)**, **automatic**, **automation** all reduced to **automat**.

for example compressed and compression are both accepted as equivalent to compress.

➡️

for exampl compress and compress ar both accept as equival to compress

NLP IN PYTHON

Two majors libraries for performing basic NLP in Python:

- Natural Language Toolkit (NLTK): started as research code, now widely used in industry and research

- Spacy: much newer implementation, more streamlined

Pros and cons to both:

- NLTK has more "stuff" implemented, is more customizable

  - This is a blessing and a curse

- Spacy is younger and feature sparse, but can be much faster

- Both are Anaconda packages
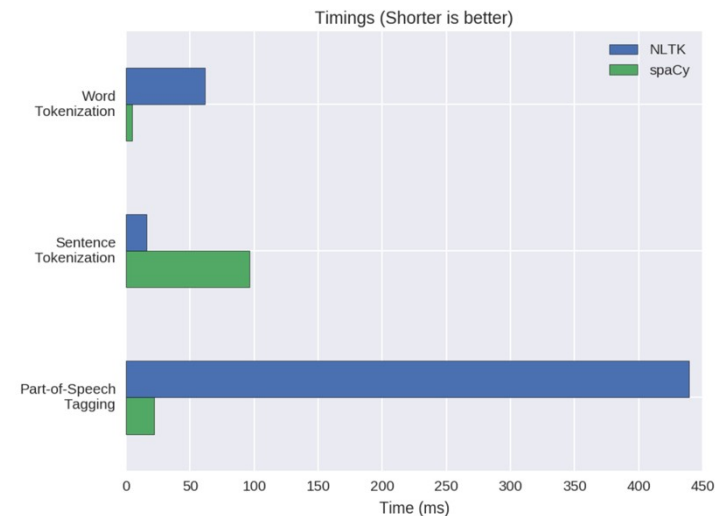
NLTK EXAMPLES

```
import nltk

# Tokenize, aka find the terms in, a sentence
sentence = "A wizard is never late, nor is he early.
He arrives precisely when he means to."
tokens = nltk.word_tokenize(sentence)
```

```
LookupError:
**********************************************************************
  Resource 'tokenizers/punkt/PY3/english.pickle' not found.
  Please use the NLTK Downloader to obtain the resource:  >>>
  nltk.download()
  Searched in:
    - '/Users/spook/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************
```


Fool of a Took!

NLTK EXAMPLES

Corpora are, by definition, large bodies of text

- **NLTK relies on a large corpus set to perform various functionalities; you can pick and choose:**

```
# Launch a GUI browser of available corpora
nltk.download()
```
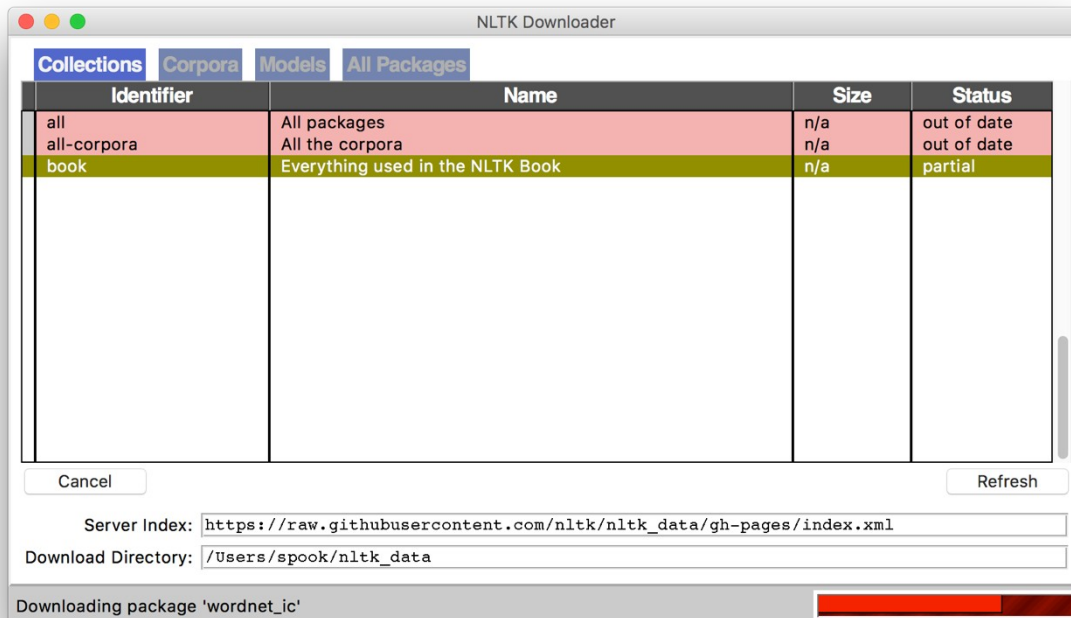


```
# Or download
everything at once!
nltk.download("all")
```

NLTK EXAMPLES

ptb	Penn Treebank	6.1 KB	not installed
punkt	Punkt Tokenizer Models	13.0 MB	installed
qc	Experimental Data for Question Classification	132.5 KB	not installed

```
import nltk

# Tokenize, aka find the terms in, a sentence
sentence = "A wizard is never late, nor is he early.
He arrives precisely when he means to."
tokens = nltk.word_tokenize(sentence)
```

```
['A', 'wizard', 'is', 'never', 'late', ',', 'nor',
'is', 'he', 'early', '.', 'He', 'arrives',
'precisely', 'when', 'he', 'means', 'to', '.']
```

(This will also tokenize words like "o'clock" into one term, and "didn't" into two term, "did" and "n't".)

53

NLTK EXAMPLES

```
# Determine parts of speech (POS) tags
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

```
[('A', 'DT'), ('wizard', 'NN'), ('is', 'VBZ'),
('never', 'RB'), ('late', 'RB'), (',', ','), ('nor',
'CC'), ('is', 'VBZ'), ('he', 'PRP'), ('early', 'RB')]
```

Abbreviation	POS
DT	Determiner
NN	Noun
VBZ	Verb (3rd person singular present)
RB	Adverb
CC	Conjunction
PRP	Personal Pronoun

Full list: https://cs.nyu.edu/grishman/jet/guide/PennPOS.html

NLTK EXAMPLES

```
# Find named entities & visualize
entities =
nltk.chunk.ne_chunk( nltk.pos_tag( nltk.word_tokeni
ze("""
    The Shire was divided into four quarters, the Farthings already referred
to. North, South, East, and West; and these again each into a number of
folklands, which still bore the names of some of the old leading families,
although by the time of this history these names were no longer found only in
their proper folklands. Nearly all Tooks still lived in the Tookland, but
that was not true of many other families, such as the Bagginses or the Boffins.
Outside the Farthings were the East and West Marches: the Buckland (see
beginning of Chapter V, Book I); and the Westmarch added to the Shire in S.R.
1462.
    """)))
entities.draw()
```

ORGANIZATION   ..   Outside IN   the DT   ORGANIZATION   were VBD   the DT   GPE   and CC   LOCATION   ::   the DT   GPE

Boffins NNP   Farthings NNS   East NNP   West NNP   Marches NNP   Buckland NNP
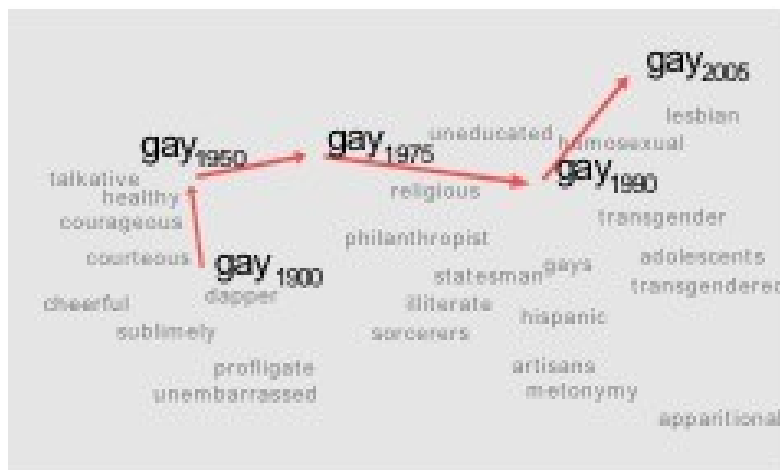
BRIEF ASIDE: VECTOR SEMANTICS OF DOCS/TERMS

"fast" is similar to "rapid"

"tall" is similar to "height"

Question answering:

Q: "How tall is Mt. Everest?"
Candidate A: "The official height of Mount Everest is 29029 feet"

[Kulkarni, Al-Rfou, Perozzi, Skiena 2015]

Many thanks to Dan Jurafsky here!

DISTRIBUTIONAL MODELS OF MEANING

Distributional models of meaning
= vector-space models of meaning
= vector semantics

Intuitions: Zellig Harris (1954):

- "oculist and eye-doctor … occur in almost the same environments"
- "If A and B have almost identical environments we say that they are synonyms."

Firth (1957):

- "You shall know a word by the company it keeps!"

[DJ]

INTUITION OF DISTRIBUTIONAL WORD SIMILARITY

A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.

From context words humans can guess tesgüino means

- an alcoholic beverage like beer

Intuition for algorithm:

- Two words are similar if they have similar word contexts.

[DJ]

FOUR KINDS OF VECTOR MODELS

Sparse vector representations

- Mutual-information weighted word co-occurrence matrices

Dense vector representations:

- Singular value decomposition (and Latent Semantic Analysis)
- Neural-network-inspired models (skip-grams, CBOW)
- Brown clusters
  - Won't go into these much – basically, classify terms into "word classes" using a particular clustering method
  - Hard clustering due to Brown et al. 1992, embed words in some space and cluster. Generally, better methods out there now …

[DJ]

SHARED INTUITION

Model the meaning of a word by embedding in a vector space.

The meaning of a word is a vector of numbers

- Vector models are also called "embeddings".

Contrast: word meaning is represented in many computational linguistic applications by a vocabulary index ("word number 545")

REMINDER: TERM-DOCUMENT MATRIX

Each cell: count of term t in a document d: $tf_{t,d}$:

- Each document is a count vector in $\mathbb{N}^v$: a column below

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

[DJ]

REMINDER: TERM-DOCUMENT MATRIX

Two documents are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

[DJ]

THE WORDS IN A TERM-DOCUMENT MATRIX

Each word is a count vector in $\mathbb{N}^D$: a row below

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

[DJ]

THE WORDS IN A TERM-DOCUMENT MATRIX

Two words are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

[DJ]

TERM-CONTEXT MATRIX FOR WORD SIMILARITY

Two words are similar in meaning if their context vectors are similar

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

[DJ]

THE WORD-WORD OR WORD-CONTEXT MATRIX

Instead of entire documents, use smaller contexts

- Paragraph
- Window of 4 words

A word is now defined by a vector over counts of context words

- **Instead of each vector being of length _D_**

- **Each vector is now of length |_V_|**

The word-word matrix is |_V_|x|_V_|, not _D_x_D_

[DJ]

WORD-WORD MATRIX
SAMPLE CONTEXTS 7 WORDS

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0		2	1	0	1	0
information	0		1	6	0	4	0

...   ...

WORD-WORD MATRIX

We showed only 4x6, but the real matrix is 50,000 x 50,000

- So it's very sparse
    - Most values are 0.
- That's OK, since there are lots of efficient algorithms for sparse matrices.

The size of windows depends on your goals

- The shorter the windows , the more syntactic the representation
    - 1-3 very syntacticy
- The longer the windows, the more semantic the representation
    - 4-10 more semanticy

MEASURING SIMILARITY

Given 2 target words v and w

- **Need a way to measure their similarity.**

Most measure of vectors similarity are based on the:

- **Dot product or inner product from linear algebra**

$$\text{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- High when two vectors have large values in same dimensions.
- Low (in fact 0) for orthogonal vectors with zeros in complementary distribution

69

PROBLEM WITH DOT PRODUCT

$$\text{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

Dot product is longer if the vector is longer. Vector length:

$$|\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

Vectors are longer if they have higher values in each dimension

That means more frequent words will have higher dot products

That's bad: we don't want a similarity metric to be sensitive to word frequency

[DJ]

SOLUTION: COSINE

Just divide the dot product by the length of the two vectors!

$$\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$$

This turns out to be the cosine of the angle between them!

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos \theta$$

$$\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} = \cos \theta$$

SIMILARITY BETWEEN DOCUMENTS

Given two documents x and y, represented by their TF-IDF vectors (or any vectors), the cosine similarity is:

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\mathsf{T}\mathbf{y}}{|\mathbf{x}| \times |\mathbf{y}|}$$

Formally, it measures the cosine of the angle between two vectors x and y:

- $\cos(0^\circ) = 1$, $\cos(90^\circ) = 0$??????????

Similar documents have high cosine similarity; dissimilar documents have low cosine similarity.
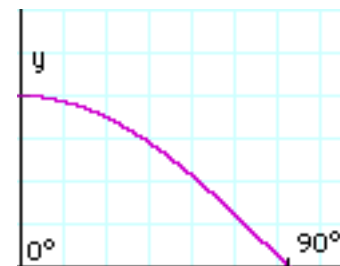
EXAMPLE

	large	data	computer
apricot	2	0	0
digital	0	1	2
information	1	6	1

$$\cos(\vec{v}, \vec{w}) = \frac{\vec{v} \bullet \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|} \bullet \frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

Which pair of words is more similar?

cosine(apricot,information) =

$$\frac{\frac{2+0+0}{\sqrt{2+0+0}\ \sqrt{1+36+1}}} \quad \frac{2}{\sqrt{2}\sqrt{38}} = .23$$

cosine(digital,information) =

$$\frac{\frac{0+6+2}{\sqrt{0+1+4}\ \sqrt{1+36+1}}} \quad = \frac{8}{\sqrt{38}\sqrt{5}} = .58$$

cosine(apricot,digital) =

$$\frac{\frac{0+0+0}{\sqrt{1+0+0}\ \sqrt{0+1+4}}} \quad = 0$$

73

(MINIMUM) EDIT DISTANCE

How similar are two strings?

Many different distance metrics (as we saw earlier when discussing entity resolution

- Typically based on the number of edit operations needed to transform from one to the other

Useful in NLP context for spelling correction, information extraction, speech recognition, etc.

BRIEF ASIDE: N-GRAMS

n-gram: Contiguous sequence of n tokens/words etc.

- Unigram, bigram, trigram, "four-gram", "five-gram", …

Figure 1 *n*-gram examples from various disciplines

Field	Unit	Sample sequence	1-gram sequence	2-gram sequence	3-gram sequence
Vernacular name			unigram	bigram	trigram
Order of resulting **Markov model**			0	1	2
Protein sequencing	amino acid	… Cys-Gly-Leu-Ser-Trp …	…, Cys, Gly, Leu, Ser, Trp, …	…, Cys-Gly, Gly-Leu, Leu-Ser, Ser-Trp, …	…, Cys-Gly-Leu, Gly-Leu-Ser, Leu-Ser-Trp, …
DNA sequencing	base pair	…AGCTTCGA…	…, A, G, C, T, T, C, G, A, …	…, AG, GC, CT, TT, TC, CG, GA, …	…, AGC, GCT, CTT, TTC, TCG, CGA, …
Computational linguistics	character	…to_be_or_not_to_be…	…, t, o, _, b, e, _, o, r, _, n, o, t, _, t, o, _, b, e, …	…, to, o_, _b, be, e_, _o, or, r_, _n, no, ot, t_, _t, to, o_, _b, be, …	…, to_, o_b, _be, be_, e_o, _or, or_, r_n, _no, not, ot_, t_t, _to, to_, o_b, _be, …
Computational linguistics	word	… to be or not to be …	…, to, be, or, not, to, be, …	…, to be, be or, or not, not to, to be, …	…, to be or, be or not, or not to, not to be, …

LANGUAGE MODELING

Assign a probability to a sentence

- Machine Translation:
  - P(**high** winds tonite) > P(**large** winds tonite)
- Spell Correction
  - The office is about fifteen **minuets** from my house
    - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
- Speech Recognition
  - P(I saw a van) >> P(eyes awe of an)
- + Summarization, question-answering, etc., etc.!!

LANGUAGE MODELING

Goal: compute the probability of a sentence or sequence of words:

- $P(W) = P(w_1,w_2,w_3,w_4,w_5\ldots w_n)$

Related task: probability of an upcoming word:

- $P(w_5|w_1,w_2,w_3,w_4)$

A model that computes either of these:

- $P(W)$ or $P(w_n|w_1,w_2\ldots w_{n-1})$ is called a language model.

(We won't talk about this much further in this class.)

SIMPLEST CASE:
UNIGRAM MODEL

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

...ne automatically generated sentences from a unigram model

    fifth, an, of, futures, the, an, incorporated, a, a,
    the, inflation, most, dollars, quarter, in, is, mass

    thrift, did, eighty, said, hard, 'm, july, bullish

    that, or, limited, the

BIGRAM MODEL

Condition on the previous word:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

```
texaco, rose, one, in, this, issue, is, pursuing, growth, in,
a, boiler, house, said, mr., gurria, mexico, 's, motion,
control, proposal, without, permission, from, five, hundred,
fifty, five, yen

outside, new, car, parking, lot, of, the, agreement, reached

this, would, be, a, record, november
```

N-GRAM MODELS

We can extend to trigrams, 4-grams, 5-grams

In general this is an insufficient model of language

- because language has long-distance dependencies:

- **"The computer which I had just put into the machine room on the fifth floor crashed."**

But we can often get away with N-gram models

MOVING ON …

Words words words!

- Free text and natural language processing in data science

- Bag of words and TF-IDF

- N-Grams and language models

- Information extraction & sentiment mining

Thanks to Amol Deshpande (UMD)

INFORMATION EXTRACTION (IE)

Information extraction (IE) systems

- Find and understand limited relevant parts of texts
- Gather information from many pieces of text
- Produce a structured representation of relevant information:
  - relations (in the database sense), a.k.a.,
  - a knowledge base
- Goals:
  - Organize information so that it is useful to people
  - Put information in a semantically precise form that allows further inferences to be made by computer algorithms

INFORMATION EXTRACTION (IE)

IE systems extract clear, factual information

- Roughly: Who did what to whom when?

E.g.,

- Gathering earnings, profits, board members, headquarters, etc. from company reports
  - The headquarters of BHP Billiton Limited, and the global headquarters of the combined BHP Billiton Group, are located in Melbourne, Australia.
  - headquarters("BHP Biliton Limited", "Melbourne, Australia")
- Learn drug-gene product interactions from medical research literature

LOW-LEVEL INFORMATION EXTRACTION

Is now available and popular in applications like Apple or Google mail, and web indexing

The Los Altos Robotics Board of Directors is having a potluck dinner Friday January 6, 2012 ... and the upcoming Botball and FRC (MVHS ... agle Strike Robotics) seasons. You are ... of these dinners three years back and it was a ...

> Create New iCal Event…
> Show This Date in iCal…
>
> Copy
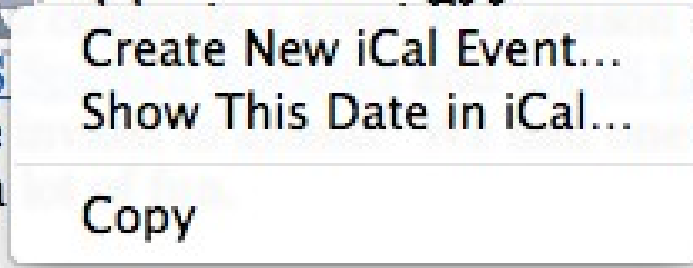
Often seems to be based on regular expressions and name lists

LOW-LEVEL INFORMATION EXTRACTION

WHY IS IE HARD ON THE WEB?

A book, Not a toy

Title

Need this price

WHY DOESN'T TEXT SEARCH (IR) WORK?

What you search for in real estate advertisements:

Town/suburb. You might think easy, but:

- Real estate agents: Coldwell Banker, Mosman
- Phrases: Only 45 minutes from Parramatta
- Multiple property ads have different suburbs in one ad

Money: want a range not a textual match

- Multiple amounts: was $155K, now $145K
- Variations: offers in the high 700s [but not rents for $270]

Bedrooms: similar issues: br, bdr, beds, B/R

NAMED ENTITY RECOGNITION (NER)

A very important sub-task: find and classify names in text

In `1917`, `Einstein` applied the general theory of relativity to model the large-scale structure of the universe. He was visiting the `United States` when `Adolf Hitler` came to power in `1933` and did not go back to `Germany`, where he had been a professor at the `Berlin Academy of Sciences`. He settled in the `U.S.`, becoming an American citizen in `1940`. On the eve of World War II, he endorsed a letter to President `Franklin D. Roosevelt` alerting him to the potential development of "extremely powerful bombs of a new type" and recommending that the `U.S.` begin similar research. This eventually led to what would become the `Manhattan` Project. Einstein supported defending the Allied forces, but largely denounced using the new discovery of nuclear fission as a weapon. Later, with the British philosopher `Bertrand Russell`, `Einstein` signed the `Russell`-`Einstein Manifesto`, which highlighted the danger of nuclear weapons. Einstein was affiliated with the `Institute for Advanced Study` in `Princeton, New Jersey`, until his death in `1955`.

Tag colours:
`LOCATION` `TIME` `PERSON` `ORGANIZATION` `MONEY` `PERCENT` `DATE`

NAMED ENTITY RECOGNITION (NER)

The uses:

- Named entities can be indexed, linked off, etc.
- Sentiment can be attributed to companies or products
- A lot of IE relations are associations between named entities
- For question answering, answers are often named entities.

Concretely:

- Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters' OpenCalais, Evri, AlchemyAPI, Yahoo's Term Extraction, …
- Apple/Google/Microsoft/… smart recognizers for document content