GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 15:NLP

Language is Complex

- A uniquely human trait
- Perception, comprehension, production
- Vocalization, manual production (signing), written
- Phonology - organization of sounds in language
- Morphology - structure of language
- Syntax - rules that govern the structure
- Semantics - meaning
- Vocabulary - the set of words available

Some Questions NLP Looks To Answer

- What are common patterns that occur in language use?
- What kinds of things do people say/write?
- What do these utterances say/ask about the world?
- What is a ***grammatical*** utterance?
  - We all speak English, but how many of us know and understand the ‘rules’, or ‘grammar’ of the language?

Are These Grammatical Utterances?

- John I believe Sally said Bill believed Sue saw
- What did Sally whisper that she had secretly read
- John wants very much for himself to win
- Those are the books you should read before it becomes too difficult to talk about
- Who did Joe think said John saw him
- The children read Mary’s stories about each other

CHOMSKY AND NORVIG

Noam Chomsky

An American linguist, often described as the ‘father of modern linguistics’. Currently Professor Emeritus at MIT.

In 2005 he was voted the world’s top public intellectual.

Peter Norvig

An American computer scientist, whose PhD thesis was entitled “A Unified Theory of Inference for Text Understanding”

Was head of the computational sciences division at NASA Ames Research Center, now Director of Research at Google

CHOMSKY AND NORVIG

At the Brains, Minds and Machines symposium, held during MIT's 150th birthday party:

Prof Chomsky derided researchers in machine learning who use purely statistical methods to produce behavior that mimics something in the world, but who don't try to understand the meaning of that behavior

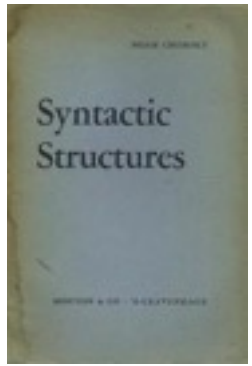
To Chomsky, building such models is like studying the dance made by a bee returning to the hive, and producing a statistically based simulation of such a dance - without attempting to understand why the bee behaved that way

CHOMSKY AND NORVIG - http://norvig.com/chomsky.html

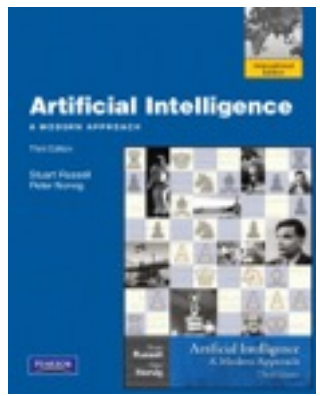
Noam Chomsky:	[...] there's been a lot of work on trying to apply statistical models to various linguistic problems. I think there have been some successes, but a lot of failures. There is a notion of success ... which I think is novel in the history of science. It interprets success as approximating unanalyzed data
Peter Norvig:	Science is a combination of gathering facts and making theories; neither can progress on its own. I think Chomsky is wrong to push the needle so far towards theory over facts; in the history of science, the laborious accumulation of facts is the dominant mode, not a novelty. The science of understanding language is no different than other sciences in this respect.

CHOMSKY AND NORVIG

Noam Chomsky



Peter Norvig



Chomsky and Norvig

Generative: seeks to describe language model of the mind for which real-world data (e.g. text) provide only indirect evidence

Empiricist: interested in describing language as it actually occurs (ignoring the underlying language models of the mind)

→ Statistical NLP models

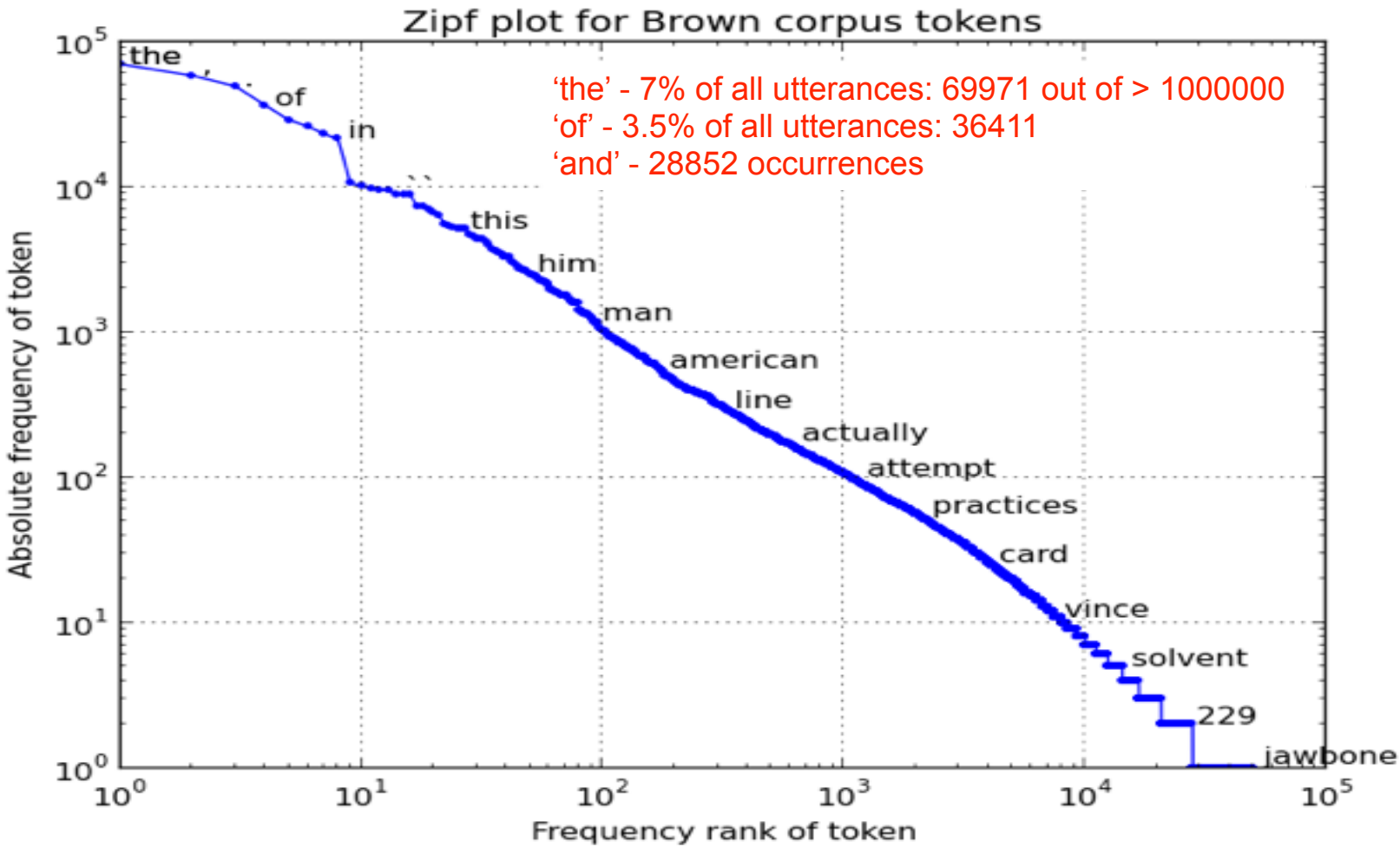
Q: What is a Corpus?

A: A large and structured set of texts

e.g. the ***Brown Corpus***, contains 500 samples of English-language text, totaling roughly one million words, compiled from works published in the United States in 1961

$\text{freq} \propto 1 / \text{rank}$

- Given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table
- The most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc


Zipf plot for Brown corpus tokens
‘the’ - 7% of all utterances: 69971 out of > 1000000
‘of’ - 3.5% of all utterances: 36411
‘and’ - 28852 occurrences
the
of
in
this
him
man
american
line
actually
attempt
practices
card
vince
solvent
229
jawbone
Absolute frequency of token
Frequency rank of token
10^0
10^1
10^2
10^3
10^4
10^5

KEY CONCEPTS - HOMOGRAPHS

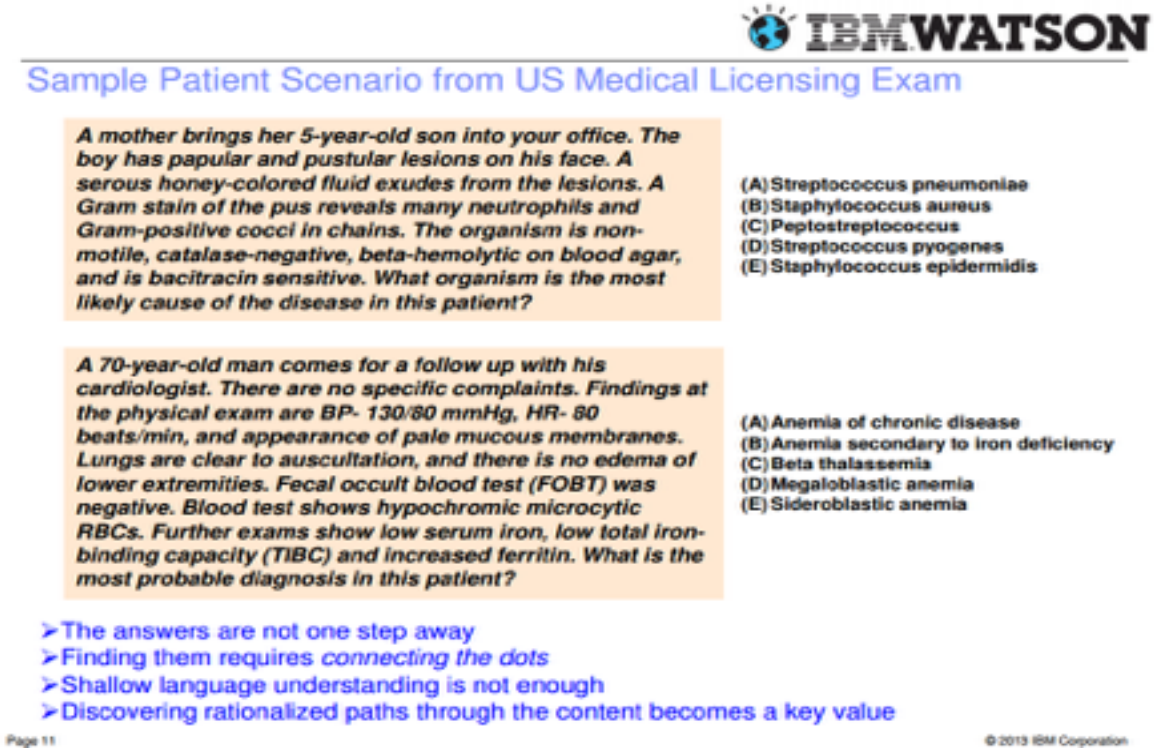
homographs—identically spelled words with multiple meanings:

- “the spirit is willing, but the flesh is weak”

- *translated to Russian and back*:
  “the vodka is agreeable, but the meat is spoiled”

KEY CONCEPTS - NLP IS HARD!

source: *New York Knowledge Engineering Meetup* (08/04/2014): I.B.M. Watson present and future



KEY CONCEPTS - NLP TASKS

Low Level	High Level
Lexical parsing *Morphological (word) segmentation* *Optical character recognition (OCR)* *Part-of-speech (POS) tagging* *Sentence boundary disambiguation* *Speech/phoneme segmentation*	*Automatic summarization* *Discourse analysis* *Machine translation* *Named entity recognition (NER)* *Natural language generation* *Natural language understanding* *Sentiment analysis* *Speech recognition* *Topic segmentation and recognition* *Word sense disambiguation*

KEY CONCEPTS - TOOLS - PYTHON NLP PACKAGES

	Pros	Cons
NLTK*:** ***Natural Language Toolkit	• *Well-documented* • *Active dev community* • *Lots of features*	• *Unsuitable for high-performance applications*
Gensim*:** ***Topic Modeling for Humans	• *Built-in distributed computing support* • *Can index datasets larger than RAM* • *Great docs, tutorials*	• *Narrow application focus* • *Smaller support community (than NLTK)*
Sklearn	• *Part of familiar set of tools for Machine Learning* • *Good lib to explore small datasets*	• *Limited set of NLP features* • *"Blows up with memory errors much sooner than other libs"*
corenlp*:** ***Wrapper for Stanford Core NLP	*Stanford Core NLP ...* • *Written in Java* • *Gold standard for serious NLP work*	• *Python wrapper around a package written in Java* • *Relatively little support*

PRACTICAL NLP

Key Concepts - Common Techniques Working With Text

- Convert to lowercase
- Remove punctuation
- Word Tokenization
- Stemming/Lemmatization: normalize word forms
- Word Tagging
- Name Entity Recognition
- Filtering stop-words
- Vectorization (n-grams and bag of words)
- Term-frequency Inverse Document Frequency (TF-IDF)

KEY CONCEPTS - WORD TOKENIZATION

- Tokenization is the act of splitting text into words
- Separating on whitespace is easy, but often results in poor results because of case, punctuation and contractions
- Conversion to lowercase and removal of punctuation is self evident

KEY CONCEPTS - WORD TOKENIZATION

- Standard approaches to the identification of an end of a sentence:
    - A period ends a sentence
    - If the preceding token is in a hand-compiled list of abbreviations then it does NOT end a sentence
    - If the next token is capitalized then the sentence ended
- Overall has an accuracy of about 95%

KEY CONCEPTS - STEMMING/LEMMATIZATION

- The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

- For example:

  am, are, is -> be
  car, cars, car's, cars' -> car

KEY CONCEPTS - STEMMING/LEMMATIZATION

- The result of this mapping of text might be something like:

  the boy's cars are different colors -> the boy car be differ color

- Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

- Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .

KEY CONCEPTS - STEMMING/LEMMATIZATION

- It is often useful to strip away conjugations and other modifiers

science, scientist -> scion

swim, swimming, swimmer -> swim

- The resulting text, while unreadable, retains semantic content

KEY CONCEPTS - WORD TAGGING

- Tagging or POS-tagging is the association between a word and it's Part Of Speech (POS).
- Apply POS-tags to each word
- Verbs, Nouns, Adjectives, etc…
- NLTK has pos_tag function
- Detailed
  - 6 verb tags, VB (verb), VBD (past tense), VBG (gerund), VBN (past participle), VBP (singular past tense), VBZ (third person singular present tense)
  - DT - determiner, NN - noun, IN - conjunction, etc

KEY CONCEPTS - NAME ENTITY RECOGNTION

- The goal of NER is to recognize tokens or words associated with people, organizations, and locations.

NLTK ne_chunk function

- returns PERSON, GPE (Geo political entity)

KEY CONCEPTS - FILTERING STOP-WORDS

- Some words are so common that they provide little useful information to a statistical language model
- For example: “the”, “is”, “it”, “not”
- Different languages have different stop words
- Any group of words could be chosen as the stop words for a given purpose
- Identification and removal of stop words:
  - Lookup each word in a list
  - Assume words and terms with the highest document frequency as stop words. This is tunable by setting a threshold

KEY CONCEPTS - VECTORIZING DOCUMENTS

Turn a document into a numerical feature vector

Measure aspects of words in a document to obtain a numerical measure.

KEY CONCEPTS - TF-IDF

- Term Frequency - Inverse Document Frequency
- Stop words are commonly occurring words across all documents - so not terribly useful
- Want commonly occurring words that appear in few documents

t = single term, d = single document, D = all documents in the corpus

- Term Frequency = frequency of word, term, n-gram in a documents
  - $TF(t, d) = N_{term\_occurrences\_in\_doc}$

KEY CONCEPTS - TF-IDF

- Document Frequency = documents containing the term as a proportion of the total number of documents
  - $DF(t, D) = N_{documents\_containing\_term}/N_{documents}$
- Inverse Document Frequency = a measure of the commonality of the term both within a given document, but across the documents in the corpus
  - $TF\text{-}IDF(t, d, D) = (N_{term}/N_{terms\_in\_document})*Log(N_{documents}/N_{documents\_containing\_term})$
  - TF-IDF is larger for words that occur more frequently in a document, but occur in fewer documents overall

TF-IDF example:

doc1	
term	*count*
this	*1*
is	*1*
a	*3*
pen	*2*

doc2	
term	*count*
this	*2*
example	*2*
has	*1*
pen	*2*

doc3	
term	*count*
a	*3*
pen	*4*
example	*1*
shines	*2*

KEY CONCEPTS - TF-IDF - EXAMPLE

TFIDF('this', doc1, [doc1, doc2, doc3]) = 1/7 * log(3/2) = 0.025
TFIDF('shines', doc3, [doc1, doc2, doc3]) = 2/10 * log(3/1) = 0.095

doc1	
term	*count*
this	*1*
is	*1*
a	*3*
pen	*2*

doc2	
term	*count*
this	*2*
example	*2*
has	*1*
pen	*2*

doc3	
term	*count*
a	*3*
pen	*4*
example	*1*
shines	*2*

GENERAL ASSEMBLY

Document Similarity

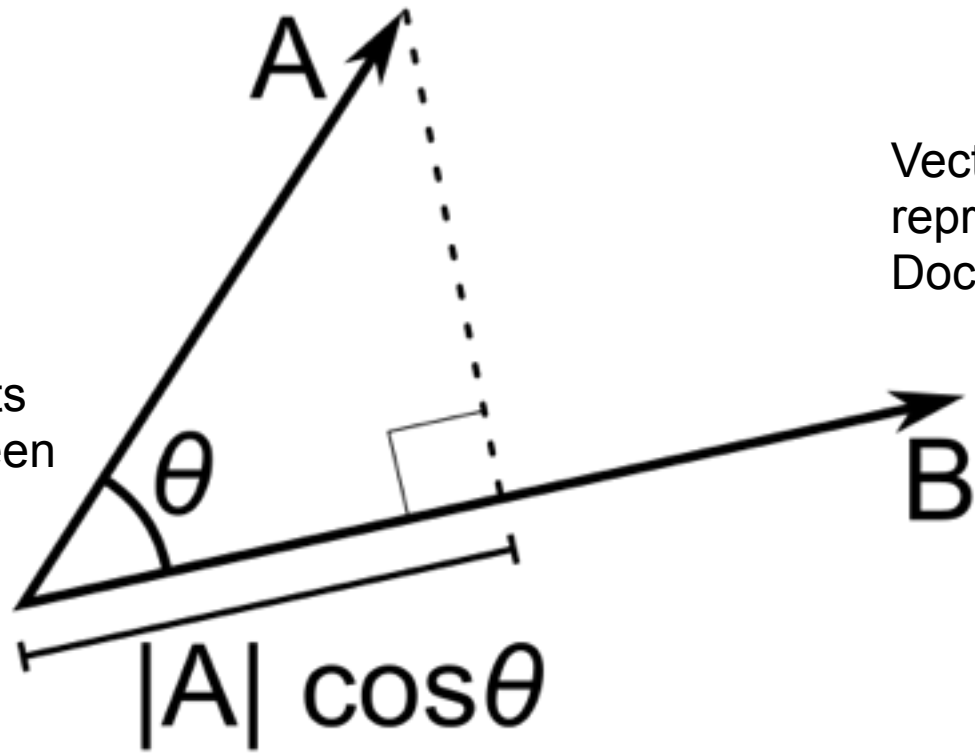
Detailed iPython notebook on this!!

KEY CONCEPTS - COSINE SIMILARITY

- By utilizing tf-idf values can describe a document as a vector
- The vector represents the tf-idf values produced for that document within a corpus of documents
- If 2 documents contain similar words, and those words are somewhat unique to those 2 documents, then the vectors produced will ‘point’ in a similar direction
- We can measure the angle between 2 vectors
- Thus the angle provides an objective measure of the similarity of 2 documents

KEY CONCEPTS - COSINE SIMILARITY

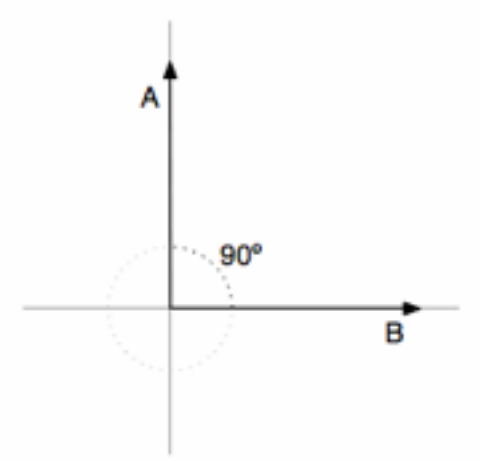
Vector A
represents
Document A



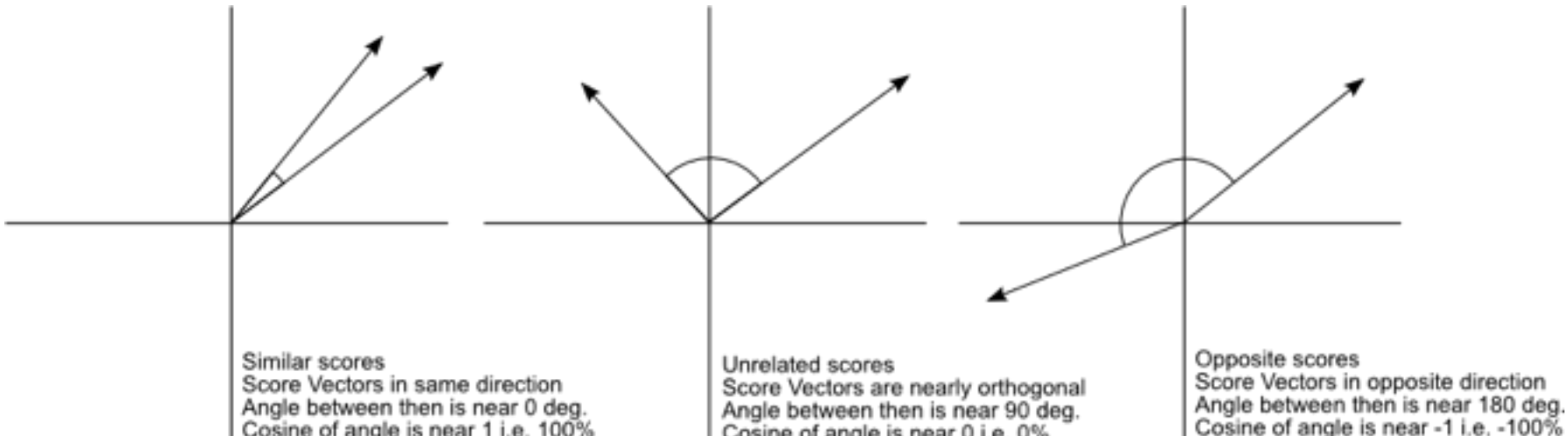
Vector B
represents
Document B

Theta represents
the angle between
the 2 vectors

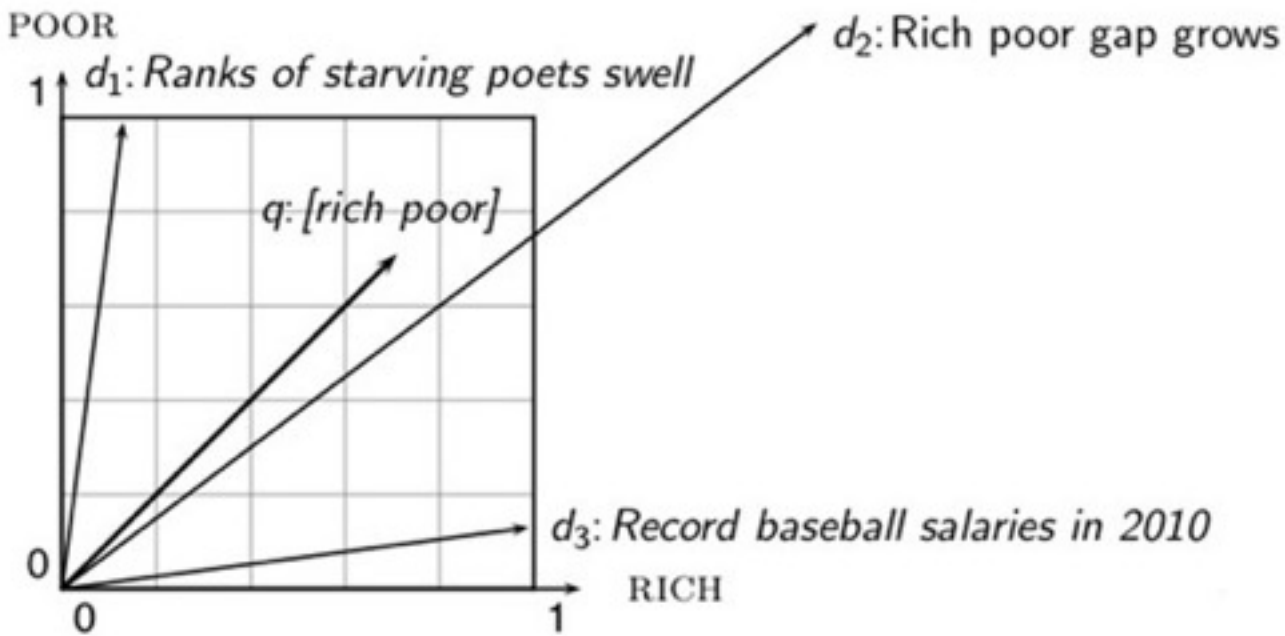
Q: What is the cosine of ninety degrees?



KEY CONCEPTS - COSINE SIMILARITY

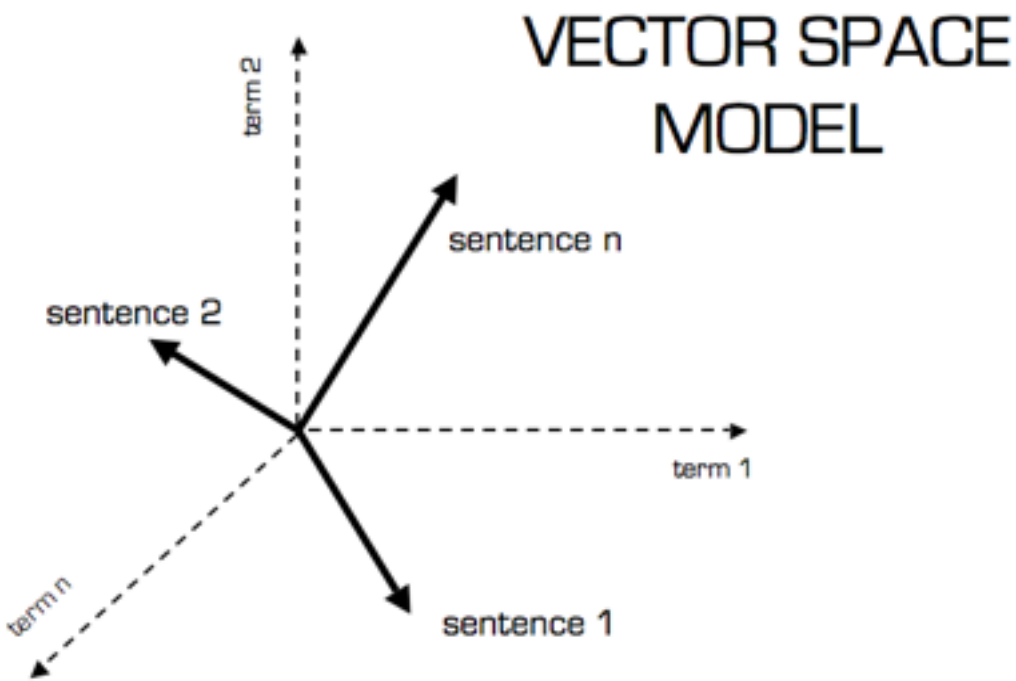


KEY CONCEPTS - COSINE SIMILARITY



KEY CONCEPTS - THE VECTOR SPACE MODEL

Modeled as vectors (with TF-IDF weights)



GENERAL ASSEMBLY

Topic Modeling

KEY CONCEPTS - TOPIC MODELING

- A statistical technique to identify themes, or topics in a set of documents
  - provides a thematic summary of a set of documents
  - topic discovery
  - reports the proportion of each topic in a document
- Different topics generate different words at particular frequencies
- Using the words in a new document one can predict membership to a topic
- Useful for news aggregators
- Useful for segmenting a corpus

Key Concepts - Topic Modeling - Latent Dirichlet Allocation

- A common implementation of topic modeling
- Unsupervised learning algorithm
- Each document is viewed as a mixture of various topics, which are determined by the presence of specific words
- Each document is viewed as a bag of words
- The various topics “generate” the words within the document
- It is a latent model, because the underlying mechanism of grouping is hidden
- It is a generative model because it assumes words flow from topics

KEY CONCEPTS - TOPIC MODELING - LATENT DIRICHLET ALLOCATION

- Corpus of documents -> topics based upon co-occurrence of words
- Each document -> probabilistically categorized into the various topics
- New documents can be similarly categorized

Key Concepts - Topic Modeling - Latent Dirichlet Allocation

- The topic distribution is assumed to have a Dirichlet Distribution
- This is a k-dimensional probability distribution with parameter α, referred to as the concentration parameter
- α is a vector of position reals
- Each probability distribution in each of the k dimensions is weighted according to the value of α.
- If α contains equal values in all k dimensions then the distributions are equally weighted

KEY CONCEPTS - TOPIC MODELING - LATENT DIRICHLET ALLOCATION

- This makes the dirichlet distribution ideal for topic analysis
- Say you have 3 topics
- Each topic will be represented by a dimension in the dirichlet distribution
- Within each topic there is a probability distribution describing the training set of documents according to the frequency of words within each document
- If α is equally weighted, for example, α = [1.0, 1.0, 1.0], then the density of words used to describe each topic is equal

KEY CONCEPTS - TOPIC MODELING - LATENT DIRICHLET ALLOCATION

Example:

- An LDA model might have topics CAT_related and DOG_related
- Within the CAT_related topic there is a high probability of generating words such as milk, meow, kitten, fur
- Within the DOG_related topic there is a high probability of generating words such as bark, bone, stick, puppy
- Stop words, when not removed, have equal probabilities amongst all the topics

KEY CONCEPTS - TOPIC MODELING - LATENT DIRICHLET ALLOCATION

- Each topic is a probability distribution over the words
- Presented with a new document and LDA model allows us to classify (probabilistically) the document in terms of it's topics

LATENT DIRICHLET ALLOCATION - EXAMPLE

(1) `Fruits and vegetables are healthy.`

(2) `I like apples, oranges, and avocados. I don't like the flu or colds.`

Let's remove stop words, giving:

(1) `fruits vegetables healthy`

(2) `apples oranges avocados flu colds`

source: [These Are Your Tweets on LDA, Part I]

LATENT DIRICHLET ALLOCATION - EXAMPLE

Let k = 2 (denoting the number of topics)
V = 8 (denoting the number of words in the corpus)

(1) fruits vegetables healthy
(2) apples oranges avocados flu colds

Topic 1 = Fruits, Vegetables, Apples, Oranges, Avocados
Topic 2 = Healthy, Flu, Colds

And:

doc1 = (2/3) Topic 1, (1/3) Topic 2
doc2 = (3/5) Topic 1, (2/5) Topic 2

Conclude there is a food topic and a health topic
See the words defining the topic
See the tweets and the topic composition of each tweet

source: These Are Your Tweets on LDA, Part I

LATENT DIRICHLET ALLOCATION - EXAMPLE

- Each topic in LDA is a probability distribution over the words.
- In this case, LDA would give ***k = 2*** distributions of size ***V = 8***.
- Each item of the distribution corresponds to a word in the vocabulary.
- For instance, let's call one of these distributions β_1.
    - β_1 lets us answer questions like: given that our topic is Topic1 ('Food'), what is the probability of generating word1 ('Fruits')?