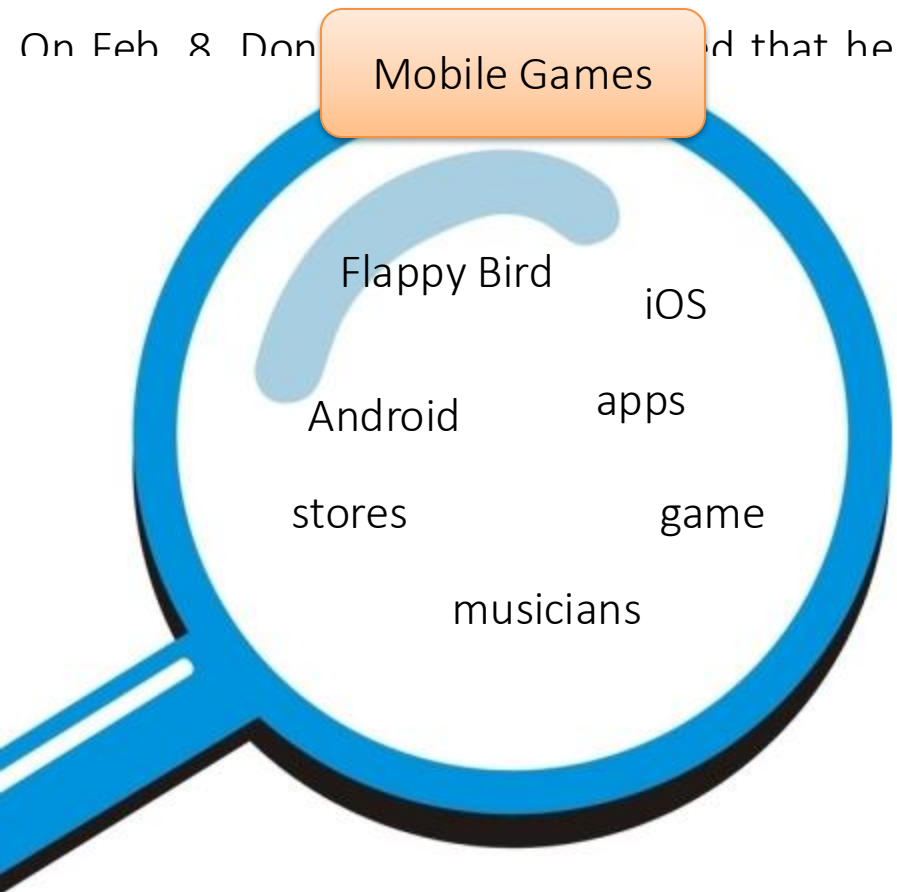# Natural Language Processing

Vector Space Model

Instructor: Yangqiu Song

# Frequency Distributions

- How can we identify the words of a text that are most informative about the topic and genre of the text?
  - You might go about finding the 50 most frequent words of a book

On Feb. 8, Don...............d that he

Mobile Games

Flappy Bird

iOS

Android

apps

stores

game

musicians

7 February 201................eat day in

Sports

Russia

Olympics

Winter

champions

Sochi

beaches

mountains

sports

# Corpus based Approach

- Distributional semantics
  - The **distributional hypothesis** in linguistics is derived from the semantic theory of language usage, i.e. words that are used and occur in the same contexts tend to purport similar meanings.

  - The basic idea of distributional semantics can be summed up in the so-called Distributional hypothesis: *linguistic items with similar distributions have similar meanings.*

We will mention **distributed representation** based neural language models in later classes
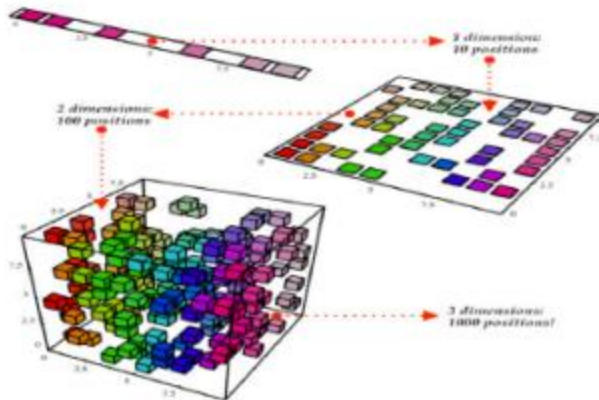
# Corpus based Approach

## 1) Corpus



## 2) Preprocessing



## 3) Dimensionality Reduction



## 4) Post Processing

# Context of Words

- Let's try to keep the kitchen _____ .


- We used WordNet to _____ the synset of *cat*.


    What does ____ mean?

# Let's try to keep the kitchen _____ .

- Observation: context can tell us a lot about word meaning
- Context: local window around a word occurrence (for now)

- Roots in linguistics from two separate papers:
  - Distributional hypothesis [Harris, 1954]:
    - Semantically similar words occur in similar contexts
    - "If A and B have almost identical environments we say that they are synonyms."
  - "You shall know a word by the company it keeps." [Firth, 1957]

- Pros: data-driven, easy to implement
- Cons: ambiguity

# Intuition of distributional word similarity

- Nida example:

  ```
  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.

# Different Kinds of Vector Models

- Sparse vector representations
  - Weighted word co-occurrence matrices

- Dense vector representations:
  - Singular value decomposition (and Latent Semantic Analysis)
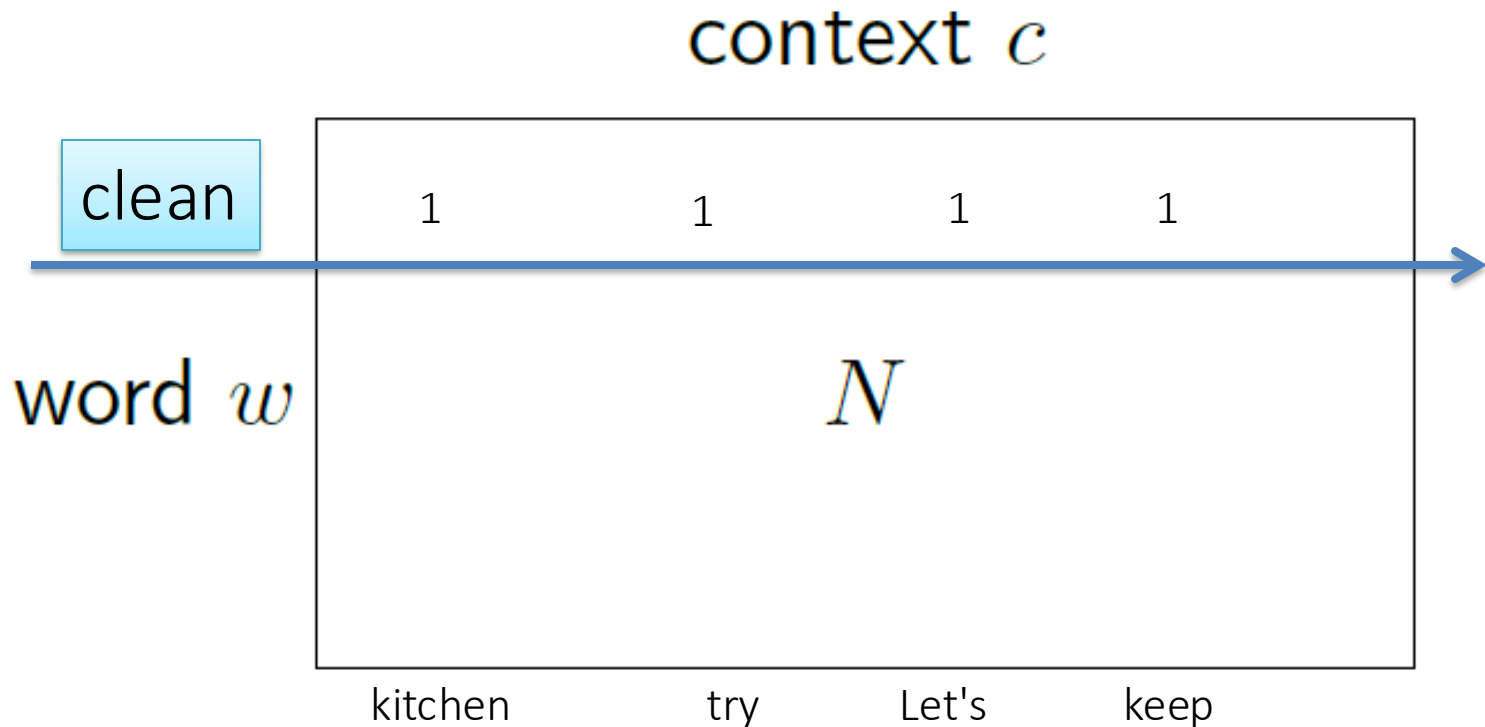  - Neural-network-inspired models (word embeddings)

# Vector Space Model (VSM)

- Represent each word with its context words
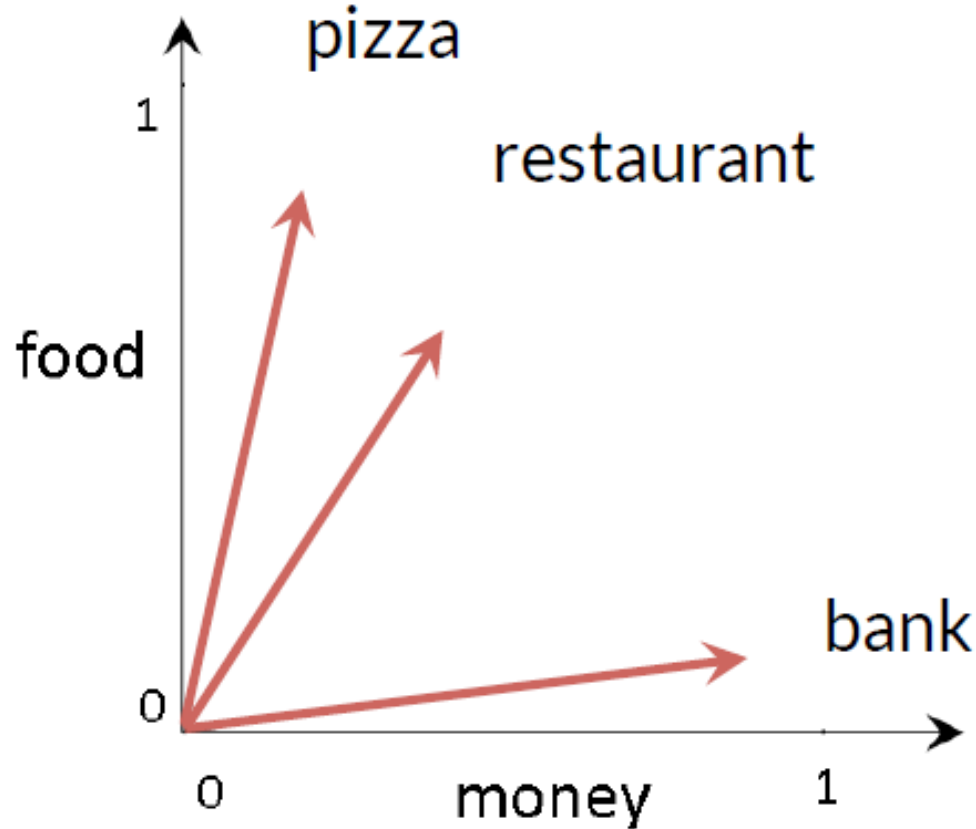
# Context Vector Construction

- Form a word-context matrix of counts (data)

context $c$

| clean | 1 | 1 | 1 | 1 |

word $w$        $N$

kitchen        try        Let's        keep

Let's try to keep the kitchen clean.

# Features for Part-of-speech Induction

- Matrix: **contexts (2)** = words on left, words on right

  Doc1: Cats have tails.
  Doc2: Dogs have tails.

|       | cats_L | dogs_L | tails_R | have_L | have_R |
|-------|--------|--------|---------|--------|--------|
| cats  | 0      | 0      | 0       | 0      | 1      |
| dogs  | 0      | 0      | 0       | 0      | 1      |
| have  | 1      | 1      | 1       | 0      | 0      |
| tails | 0      | 0      | 0       | 1      | 0      |

# Document Representation
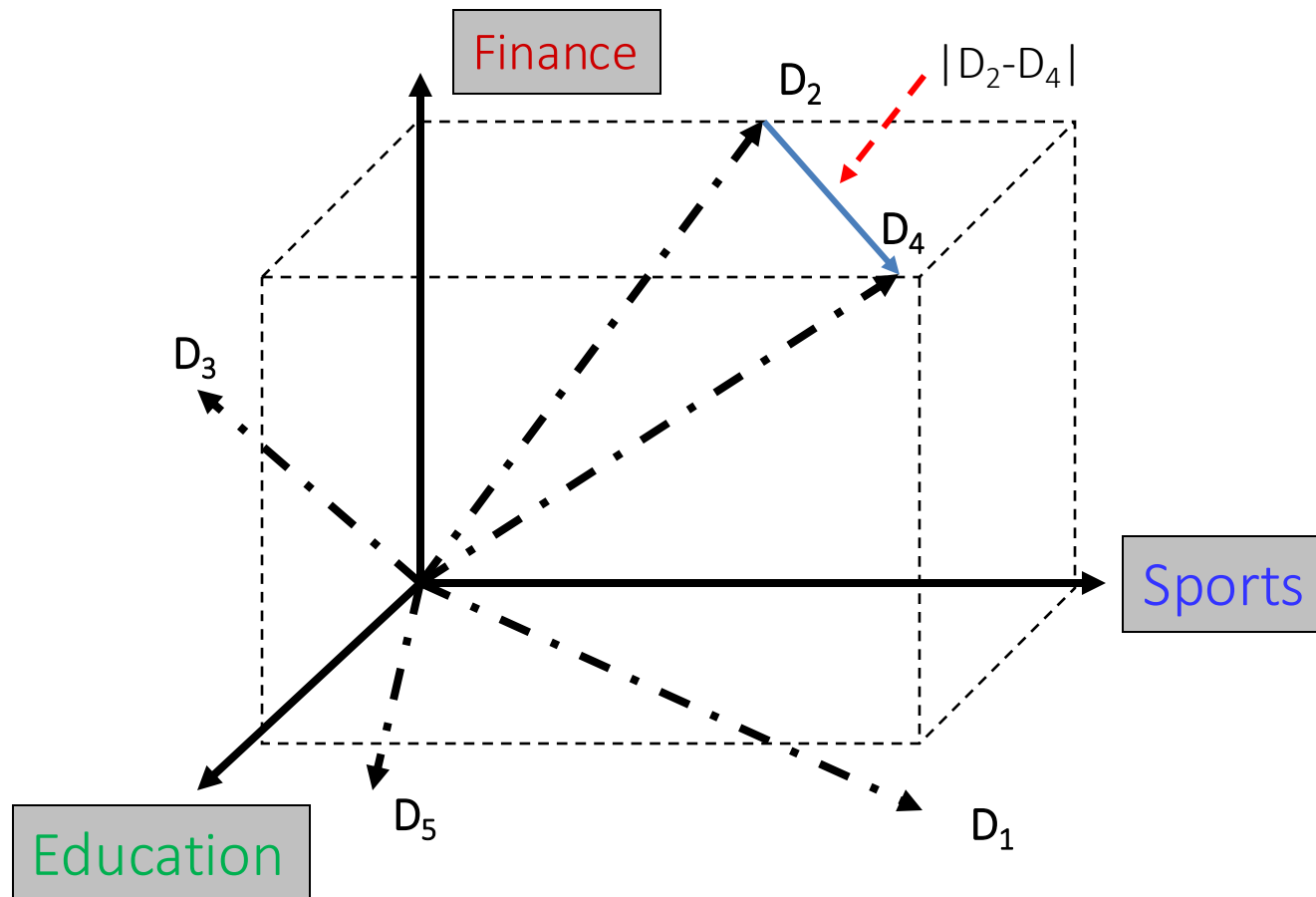
- Matrix: contexts = documents that word appear in

Doc1: Cats have tails.
Doc2: Dogs have tails.

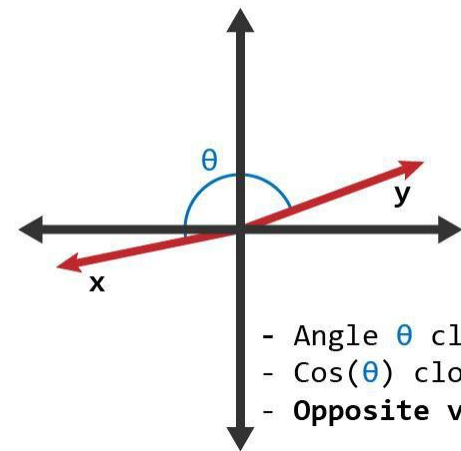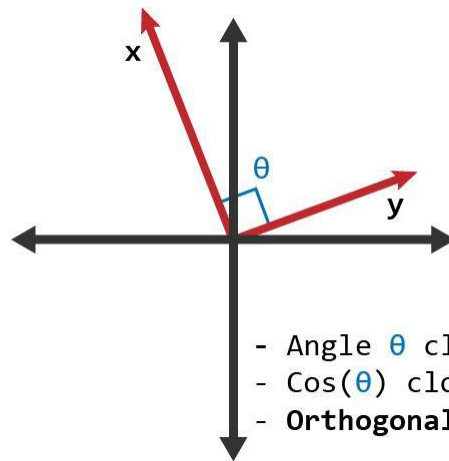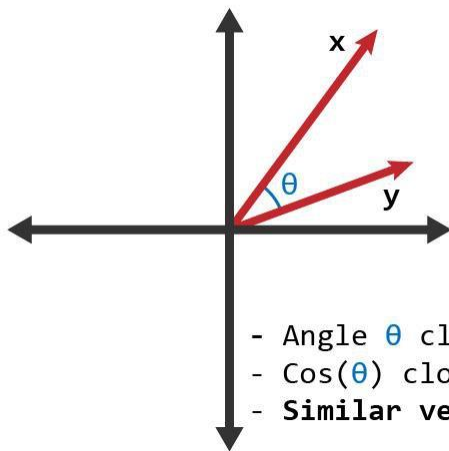|      | Doc1 | Doc2 |
|------|------|------|
| cats | 1    | 0    |
| dogs | 0    | 1    |
| have | 1    | 1    |
| tails | 1   | 1    |

# Document Vector Space Model

- All documents are projected into this concept space

# Cosine Similarity

$$cos\theta = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$$

-1: vectors point in opposite directions
+1: vectors point in same directions
0: vectors are orthogonal



- Angle θ close to 0
- Cos(θ) close to 1
- **Similar vectors**

- Angle θ close to 90
- Cos(θ) close to 0
- **Orthogonal vectors**

- Angle θ close to 180
- Cos(θ) close to -1
- **Opposite vectors**

https://www.learndatasci.com/glossary/cosine-similarity/
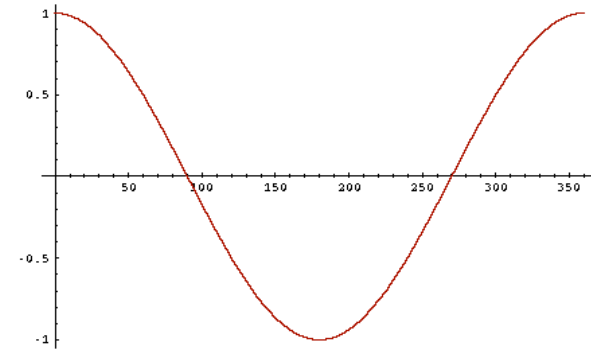
# Is This Just as Simple as Counting?

What if I give you a raw document?

# Let's take a look at a document

- On Feb. 8, Dong Nguyen announced that he would be removing his hit game Flappy Bird from both the iOS and Android app stores, saying that the success of the game is something he never wanted. Some fans of the game took it personally, replying that they would either kill Nguyen or kill themselves if he followed through with his decision.

- Frank Lantz, the director of the New York University Game Center, said that Nguyen's meltdown resembles how some actors or musicians behave. "People like that can go a little bonkers after being exposed to this kind of interest and attention," he told ABC News. "Especially when there's a healthy dose of Internet trolls."

# Summary of Preprocessing

Example: *'Text mining is to identify useful information.'*

- ## Tokenization:
  - D1: *'Text', 'mining', 'is', 'to', 'identify', 'useful', 'information', '.'*

- ## Stemming/normalization:
  - D1: *'text', 'mine', 'is', 'to', 'identify', 'use', 'inform', '.'*

- ## N-gram construction:
  - D1: *'text-mine', 'mine-is', 'is-to', 'to-identify', 'identify-use', 'use-inform', 'inform-.'*

- ## Stopword/controlled vocabulary filtering:
  - D1: *'text-mine', 'to-identify', 'identify-use', 'use-inform'*

# Document Tokenization

- Regular expressions
  - \\w+: so-called -> 'so', 'called'
  - \\s+: It's -> 'It's' instead of 'It', ''s'
- Statistical methods
  - Explore rich features to decide where the boundary of a word is
    - Apache OpenNLP (http://opennlp.apache.org/)
    - Stanford NLP Parser (http://nlp.stanford.edu/software/lex-parser.shtml)
  - Online Demo
    - Stanford (http://corenlp.run/)

# Bag-of-words Representation

- Term as the basis for vector space
  - Doc1: Text mining is to identify useful information.
  - Doc2: Useful information is mined from text.
  - Doc3: Apple is delicious.

|  | text | information | identify | mining | mined | is | useful | to | from | apple | delicious |
|------|------|-------------|----------|--------|-------|----|--------|----|------|-------|-----------|
| Doc1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| Doc2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| Doc3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

- Assumption
  - Words are independent from each other
- Pros
  - Simple
- Cons
  - Basis vectors are clearly not linearly independent!
  - Grammar and order are missing

# Bag-of-Words with N-grams

- N-grams: a contiguous sequence of n tokens from a given text
  - "Text mining is to identify useful information."
  - Bi-grams: "text_mining", "mining_is", "is_to", "to_identify" , "identify_useful", "useful_information", "information_."
- Pros: capture local dependency and order
- Cons: increase the vocabulary size

# Statistics of Words in Corpus

- Zipf's law
  - Frequency of a word is inversely proportional to its rank in the frequency table
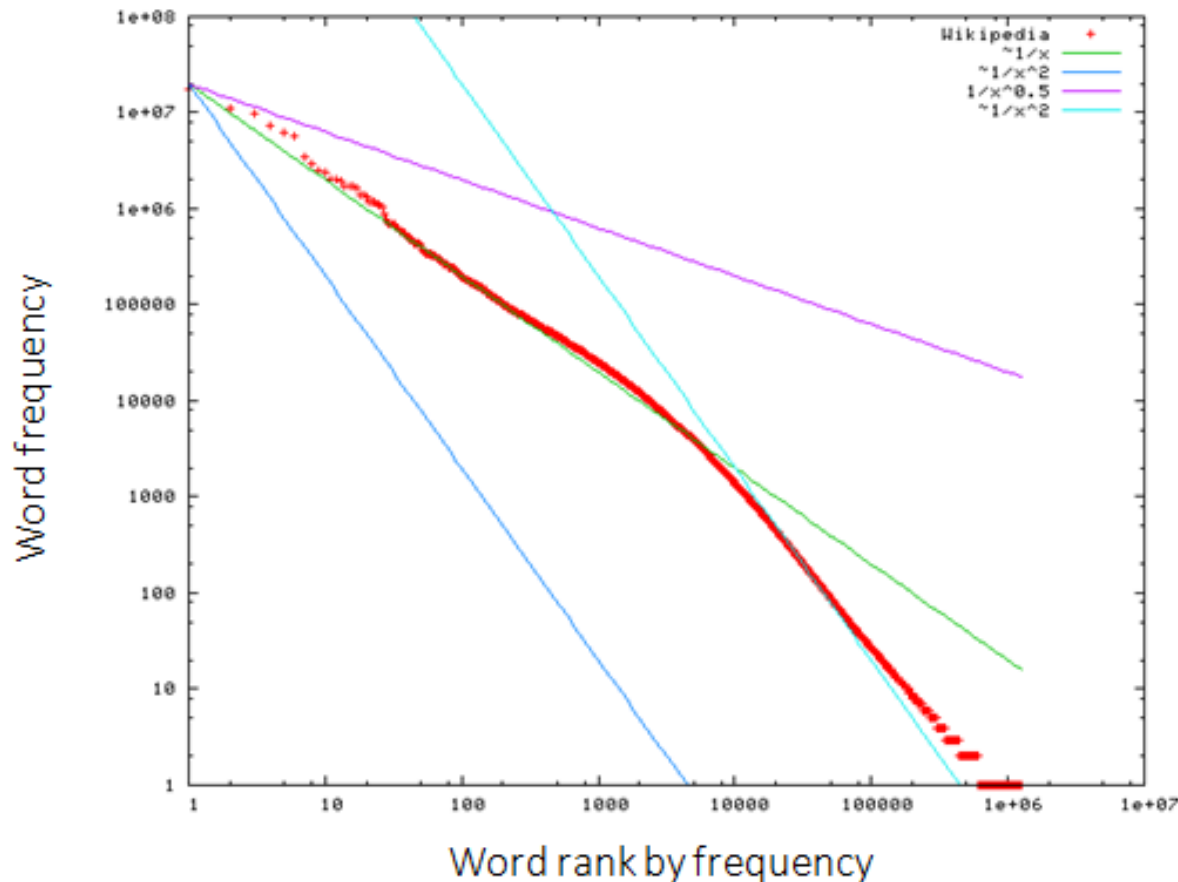


A plot of word frequency in Wikipedia (Nov 27, 2006)

# Zipf's Law Tells Us

- Head words take large portion of occurrences, but they are semantically meaningless
  - E.g., the, a, an, we, do, to
- Tail words take major portion of vocabulary, but they rarely occur in documents
  - E.g., dextrosinistral
- The rest is most representative
  - To be included in the controlled vocabulary

*In the Brown Corpus of American English text, the word "the" is the most frequently occurring word, and by itself accounts for nearly **7%** of all word occurrences; the second-place word "of" accounts for slightly over **3.5%** of words.*

# Better Document Representation

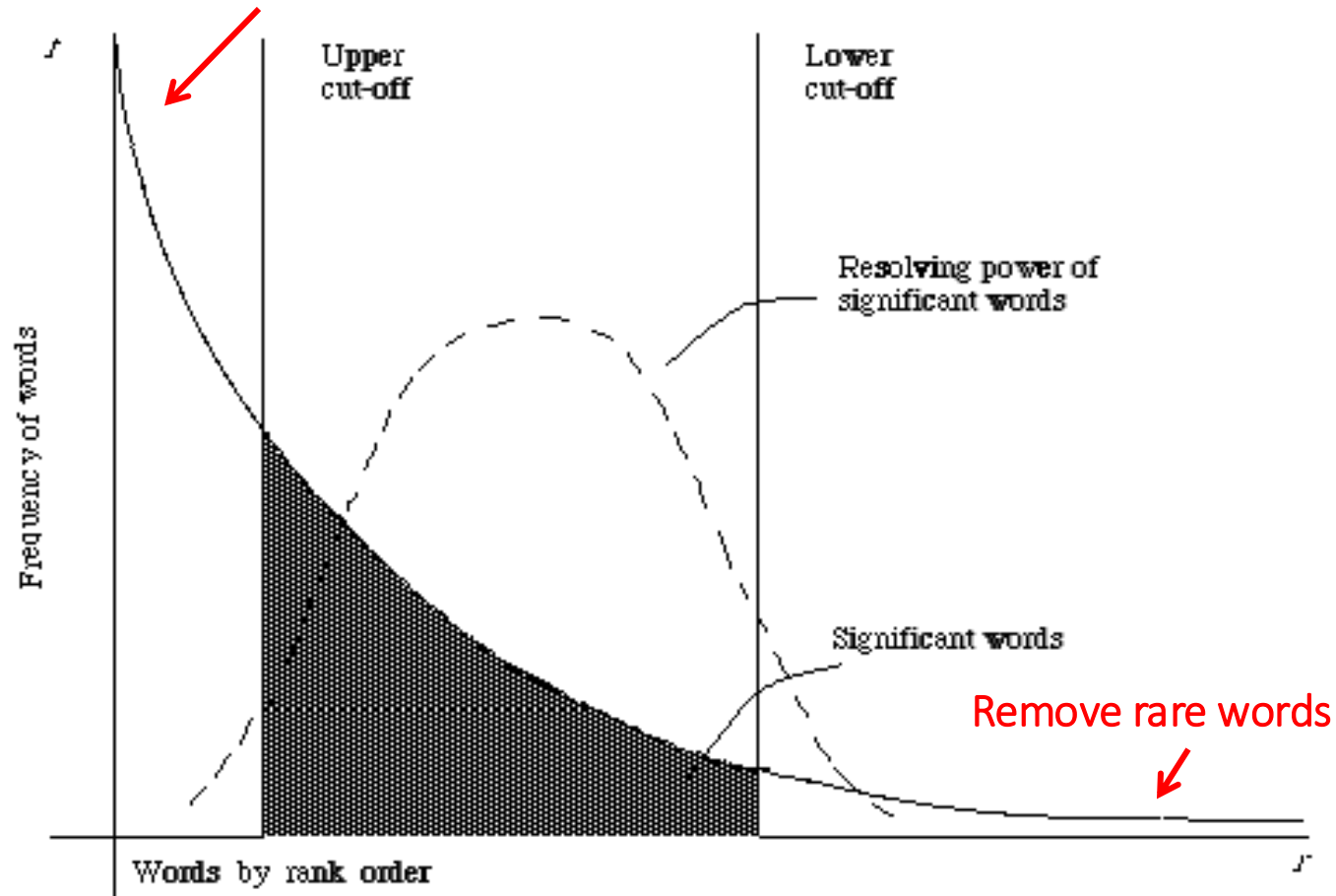Remove non-informative words

Remove rare words



Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz[44] page 120)

# Stopwords

- Useless words for document analysis
  - Not all words are informative
  - Remove such words to reduce vocabulary size
  - No universal definition
  - Risk: break the original meaning and structure of text
    - E.g., this is not a good option -> option

      to be or not to be -> null

| Nouns | Verbs | Adjectives | Prepositions | Others |
|---|---|---|---|---|
| 1. time | 1. be | 1. good | 1. to | 1. the |
| 2. person | 2. have | 2. new | 2. of | 2. and |
| 3. year | 3. do | 3. first | 3. in | 3. a |
| 4. way | 4. say | 4. last | 4. for | 4. that |
| 5. day | 5. get | 5. long | 5. on | 5. I |
| 6. thing | 6. make | 6. great | 6. with | 6. it |
| 7. man | 7. go | 7. little | 7. at | 7. not |
| 8. world | 8. know | 8. own | 8. by | 8. he |
| 9. life | 9. take | 9. other | 9. from | 9. as |
| 10. hand | 10. see | 10. old | 10. up | 10. you |
| 11. part | 11. come | 11. right | 11. about | 11. this |
| 12. child | 12. think | 12. big | 12. into | 12. but |
| 13. eye | 13. look | 13. high | 13. over | 13. his |
| 14. woman | 14. want | 14. different | 14. after | 14. they |
| 15. place | 15. give | 15. small | 15. beneath | 15. her |
| 16. work | 16. use | 16. large | 16. under | 16. she |
| 17. week | 17. find | 17. next | 17. above | 17. or |
| 18. case | 18. tell | 18. early | | 18. an |
| 19. point | 19. ask | 19. young | | 19. will |
| 20. government | 20. work | 20. important | | 20. my |
| 21. company | 21. seem | 21. few | | 21. one |
| 22. number | 22. feel | 22. public | | 22. all |
| 23. group | 23. try | 23. bad | | 23. would |
| 24. problem | 24. leave | 24. same | | 24. there |
| 25. fact | 25. call | 25. able | | 25. their |

The OEC: Facts about the language

# Stemming

- Reduce inflected or derived words to their root form
  - Plurals, adverbs, inflected word forms
    - E.g., ladies -> lady, referring -> refer, forgotten -> forget
  - Solutions (for English)
    - **Porter stemmer**: patterns of vowel-consonant sequence
    - **Krovetz stemmer**: morphological rules
  - Risk: lose precise meaning of the word
    - E.g., lay -> lie (a false statement? or be in a horizontal position?)

# Summary of Preprocessing

Example: '*Text mining is to identify useful information.*'

- **Tokenization**:
  - D1: *'Text', 'mining', 'is', 'to', 'identify', 'useful', 'information', '.'*

Optional

- **Stemming/normalization**:
  - D1: *'text', 'mine', 'is', 'to', 'identify', 'use', 'inform', '.'*

- **N-gram construction**:
  - D1: *'text-mine', 'mine-is', 'is-to', 'to-identify', 'identify-use', 'use-inform', 'inform-.'*

- **Stopword/controlled vocabulary filtering**:
  - D1: *'text-mine', 'to-identify', 'identify-use', 'use-inform'*

# Term Weighting

- Term as the basis for vector space
  - Doc1: Text mining is to identify useful information.
  - Doc2: Useful information is mined from text.
  - Doc3: Apple is delicious.

|  | text | information | identify | mining | mined | is | useful | to | from | apple | delicious |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Doc1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| Doc2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| Doc3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

- "Repeated occurrences" are less informative than the "first occurrence"

- Information about semantic does not increase proportionally with number of term occurrence
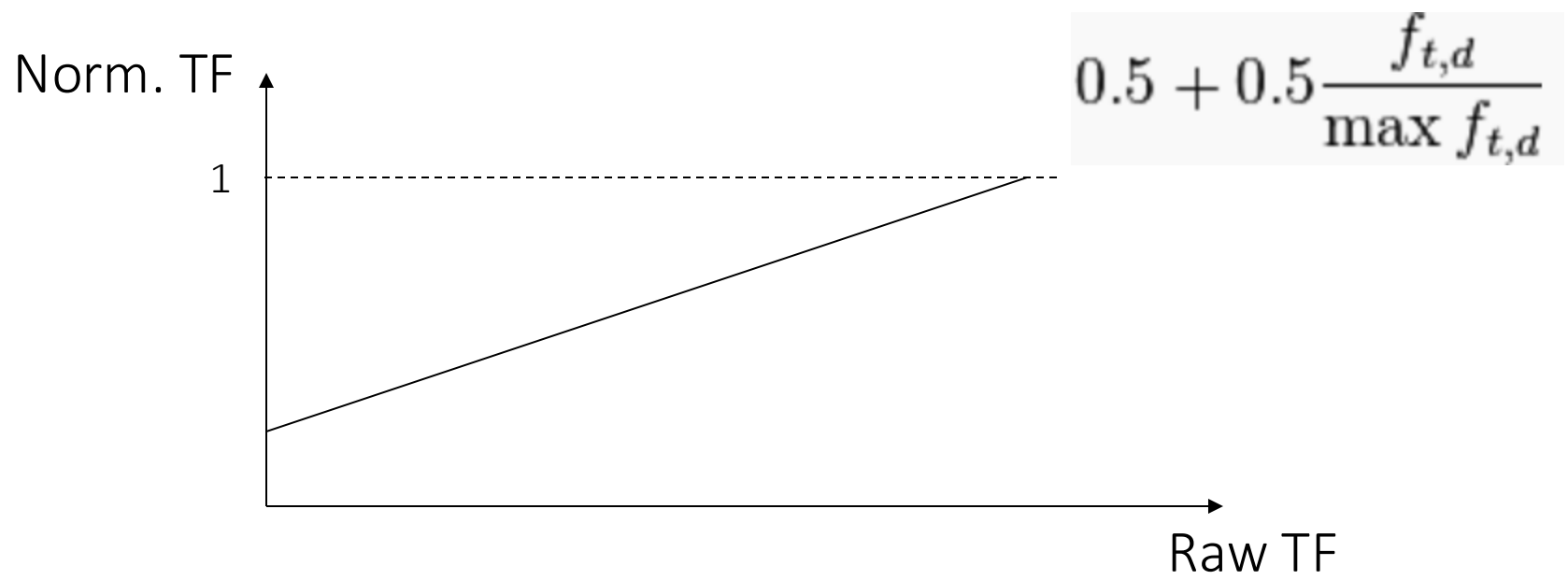
# Term Frequency Weights

- Examples of weighting terms

## Variants of TF weight

| weighting scheme | TF weight |
| --- | --- |
| binary | $0, 1$ |
| raw frequency | $f_{t,d}$ |
| log normalization | $\log(1 + f_{t,d})$ |
| double normalization 0.5 | $0.5 + 0.5 \dfrac{f_{t,d}}{\max f_{t,d}}$ |
| double normalization K | $K + (1 - K) \dfrac{f_{t,d}}{\max f_{t,d}}$ |

Number of times term $t$ appearing in document $d$

Norm. TF

$$0.5 + 0.5 \frac{f_{t,d}}{\max f_{t,d}}$$

1

Raw TF

Norm. TF

$$\log(1 + f_{t,d})$$

Raw TF
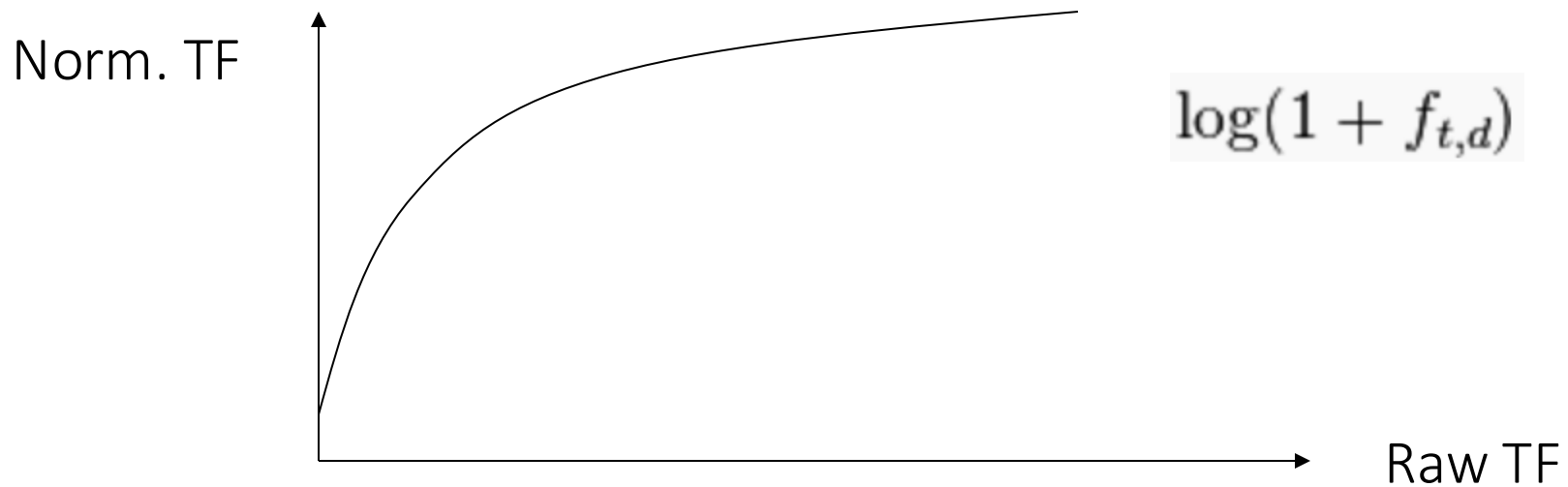
# Document Frequency Weighting

- Idea: a term is more discriminative if it occurs only in fewer documents
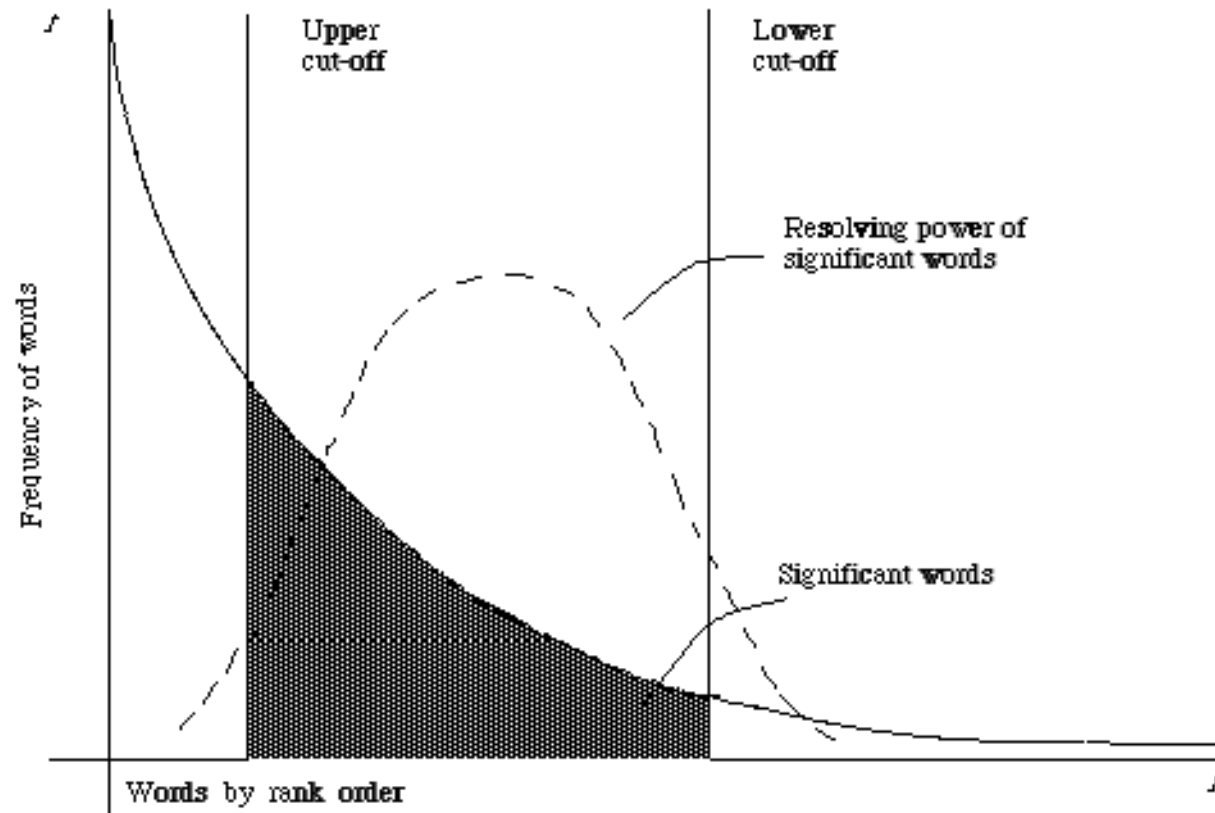


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

# Inverse Document Frequency

- Examples of IDF

### Variants of IDF weight

| weighting scheme | IDF weight |
|---|---|
| unary | $1$ |
| inverse frequency | $\log \dfrac{N}{n_t}$ |
| inverse frequency smooth | $\log(1 + \dfrac{N}{n_t})$ |
| inverse frequency max | $\log \left( 1 + \dfrac{\max_t n_t}{n_t} \right)$ |
| probabilistic inverse frequency | $\log \dfrac{N - n_t}{n_t}$ |

Non-linear scaling

Total number of docs in collection

Number of docs containing term *t*

# TFIDF

- ## Term frequency–Inverse document frequency
  - Higher tf: more frequently a word appearing in a document
  - Higher idf: less frequently a word appearing in a corpus

**Recommended TF-IDF weighting schemes**

| weighting scheme | document term weight | query term weight |
|---|---|---|
| 1 | $f_{t,d} \times \log \dfrac{N}{n_t}$ | $\left(0.5 + 0.5 \dfrac{f_{t,q}}{\max_t f_{t,q}}\right) \times \log \dfrac{N}{n_t}$ |
| 2 | $1 + \log f_{t,d}$ | $\log\left(1 + \dfrac{N}{n_t}\right)$ |
| 3 | $(1 + \log f_{t,d}) \times \log \dfrac{N}{n_t}$ | $(1 + \log f_{t,q}) \times \log \dfrac{N}{n_t}$ |

# Summary of Vector Space Model

- Pros:
  - Empirically effective!
  - Intuitive
  - Easy to implement
  - Well-studied/mostly evaluated
  - <span style="color:red">Warning: many variants of TF-IDF!</span>
- Cons
  - Assume term independence
  - Lack of "predictive adequacy"
    - Arbitrary term weighting
    - Arbitrary similarity measure
  - Lots of parameter tuning!
- Any improvements?