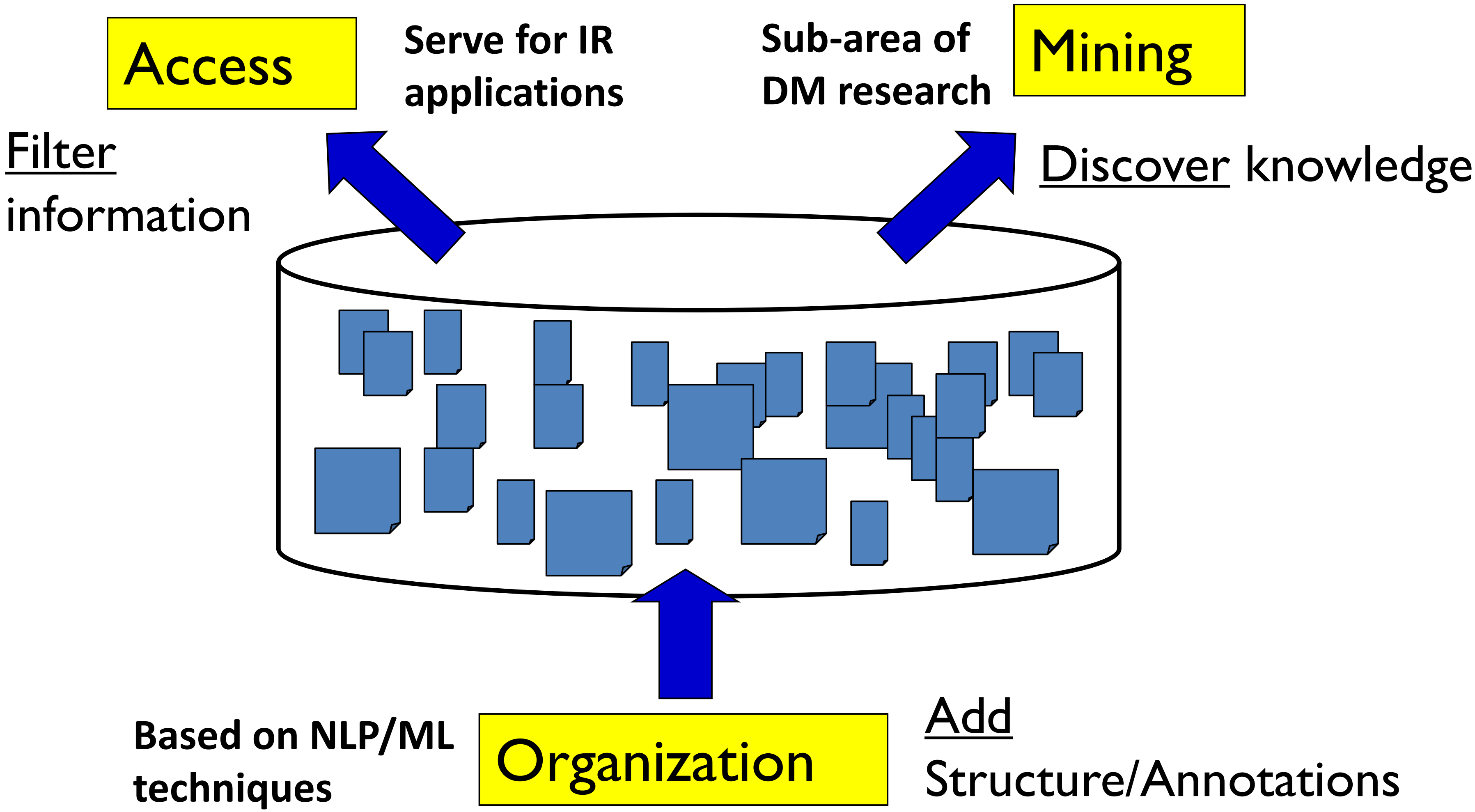
Vector Space Model

Hongning Wang
CS@UVa

Recap: what is text mining

- *"Text mining, also referred to as **text data mining**, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text."* - wikipedia
- *"Another way to view text data mining is as a process of **exploratory** data analysis that leads to **heretofore unknown** information, or to answers for questions for which the answer is not currently known."* - Hearst, 1999

Recap: text mining in general



Today’s lecture

1. How to represent a document?
   – Make it computable
2. How to infer the relationship among documents or identify the structure within a document?
   – Knowledge discovery

How to represent a document

- Represent by a string?
  –

University of Virginia

From Wikipedia, the free encyclopedia

The **University of Virginia** (**UVA** or **U.Va.**), often referred to as simply **Virginia**, is a public research university in Charlottesville, Virginia. UVA is known for its historic foundations, student-run honor code, and secret societies.

Its initial Board of Visitors included U.S. Presidents Thomas Jefferson, James Madison, and James Monroe. President Monroe was the sitting President of the United States at the time of the founding; Jefferson and Madison were the first two rectors. UVA was established in 1819, with its Academical Village and original courses of study conceived and designed entirely by Jefferson. UNESCO designated it a World Heritage Site in 1987, an honor shared with nearby Monticello.[4]

The first university of the American South elected to the Association of American Universities in 1904, UVA is classified as *Very High Research Activity* in the Carnegie Classification. The university is affiliated with 7 Nobel Laureates, and has produced 7 NASA astronauts, 7 Marshall Scholars, 4 Churchill Scholars, 29 Truman Scholars, and 50 Rhodes Scholars, the most of any state-affiliated institution in the U.S.[5][6][7] Supported in part by the Commonwealth, it receives far more funding from private sources than public, and its students come from all 50 states and 147 countries.[2][8][9] It also operates a small liberal arts branch campus in the far southwestern corner of the state.

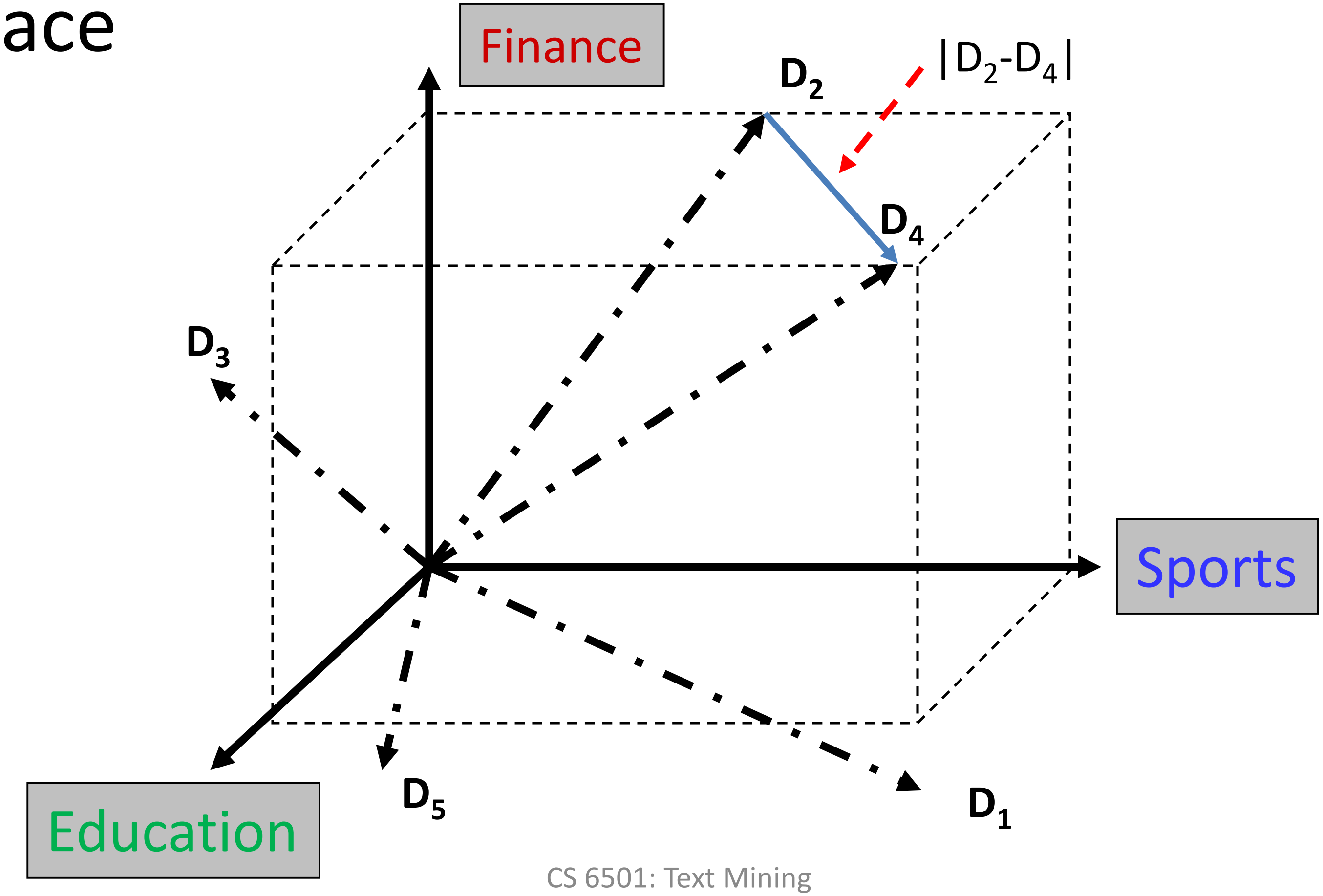
- Re
  – rsive

Vector space model

- Represent documents by <u>concept</u> vectors
  - Each concept defines one dimension
  - *k* concepts define a high-dimensional space
  - Element of vector corresponds to concept weight
    - E.g., $d=(x_1,\ldots,x_k)$, x_i is "importance" of concept i in d
- Distance between the vectors in this concept space
  - Relationship among documents

An illustration of VS model

- All documents are projected into this concept space



What the VS model doesn't say

- How to define/select the "basic concept"
  - Concepts are assumed to be <u>orthogonal</u>
- How to assign weights
  - Weights indicate how well the concept characterizes the document
- How to define the distance metric

What is a good "Basic Concept"?

- Orthogonal
  - Linearly independent basis vectors
    - "Non-overlapping" in meaning
  - No ambiguity
- Weights can be assigned automatically and accurately
- Existing solutions
  - Terms or N-grams, a.k.a., Bag-of-Words
  - Topics ← We will come back to this later

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

Tokenization

- Break a stream of text into meaningful units
  - Tokens: words, phrases, symbols
    - **Input:** It’s not straight-forward to perform so-called “tokenization.”
    - **Output(1):** 'It’s', 'not', 'straight-forward', 'to', 'perform', 'so-called', '“tokenization.”'
    - **Output(2):** 'It', '’', 's', 'not', 'straight', '-', 'forward, 'to', 'perform', 'so', '-', 'called', ‘“', 'tokenization', '.', '”‘
  - Definition depends on language, corpus, or even context

Tokenization

- Solutions
  - Regular expressions
    - [\w]+: so-called -> ‘so’, ‘called’
    - [\S]+: It’s -> ‘It’s’ instead of ‘It’, ‘’s’
  - Statistical methods ← We will come back to this later
    - 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://nlp.stanford.edu:8080/parser/index.jsp)
      - UIUC (http://cogcomp.cs.illinois.edu/curator/demo/index.html)

Bag-of-Words representation

	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
- ***The most frequently used document representation***
  - ***Image, speech, gene sequence***

Bag-of-Words with N-grams

- N-grams: a contiguous sequence of n tokens from a given piece of text
  - E.g., '*Text mining is to identify useful information.*'
  - Bigrams: '*text_mining*', '*mining_is*', '*is_to*', '*to_identify*', '*identify_useful*', '*useful_information*', '*information_.*'
- Pros: capture local dependency and order
- Cons: a purely statistical view, increase the vocabulary size $O(V^N)$

Automatic document representation

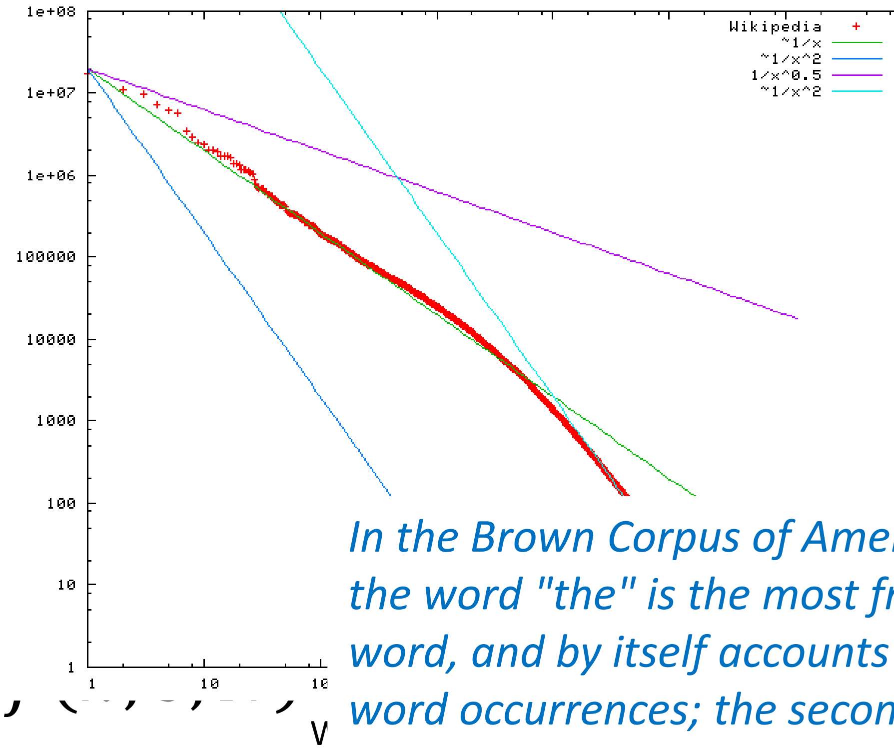
- Represent a document with all the occurring words
  - Pros
    - Preserve all information in the text (hopefully)
    - Fully automatic
  - Cons
    - Vocabulary gap: cars v.s., car, talk v.s., talking
    - Large storage: N-grams needs $O(V^N)$
  - Solution
    - Construct controlled vocabulary

A statistical property of language

- Zipf's law

Discrete version of power law

  – Freque portional to its rank
  – Formal
    - $f(k$
    where bulary size; s is langu
  – Simply: f



A plot of word freq

*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.*

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

Automatic document representation

Remove non-informative words

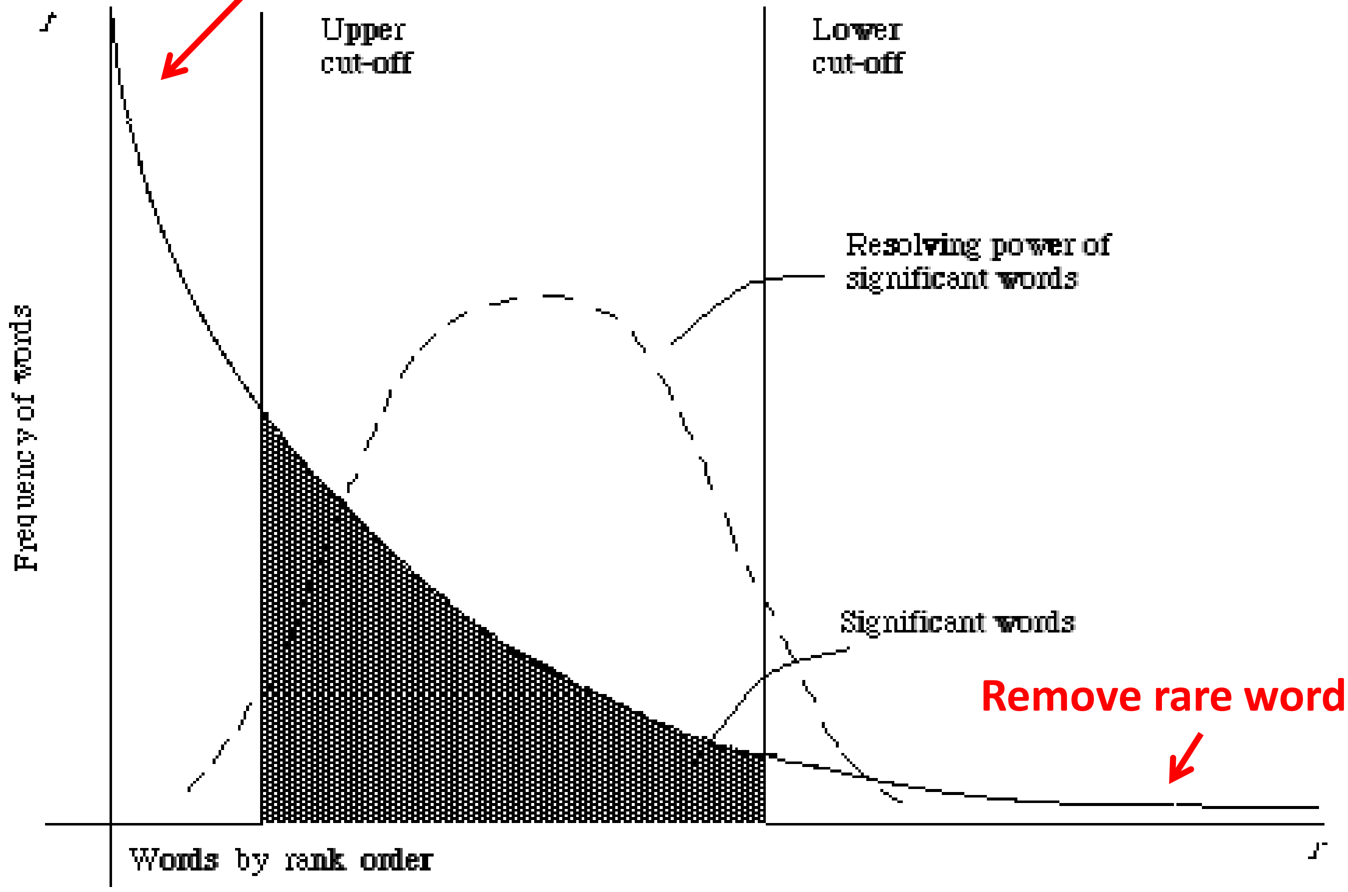


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

Normalization

- Convert different forms of a word to a normalized form in the vocabulary
  - U.S.A. -> USA, St. Louis -> Saint Louis
- Solution
  - Rule-based
    - Delete periods and hyphens
    - All in lower cases
  - Dictionary-based ← We will come back to this later
    - Construct equivalent class
      - Car -> “automobile, vehicle”
      - Mobile phone -> “cellphone”

Stemming

- Reduce inflected or derived words to their root form
  - Plurals, adverbs, inflected word forms
    - E.g., ladies -> lady, referring -> refer, forgotten -> forget
  - Bridge the vocabulary gap
  - 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?)

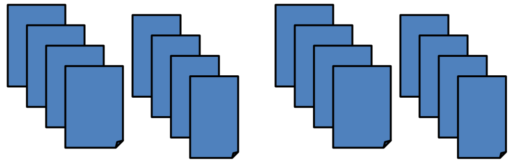
Stopwords

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

Constructing a VSM representation

Mapper



Naturally fit into MapReduce paradigm!

D1: ‘*Text mining is to identify useful information.*’

1. **Tokenization:**

   D1: *‘Text’, ‘mining’, ‘is’, ‘to’, ‘identify’, ‘useful’, ‘information’, ‘.’*

2. **Stemming/normalization:**

   D1: *‘text’, ‘mine’, ‘is’, ‘to’, ‘identify’, ‘use’, ‘inform’, ‘.’*

3. **N-gram construction:**

   D1: *‘text-mine’, ‘mine-is’, ‘is-to’, ‘to-identify’, ‘identify-use’, ‘use-inform’, ‘inform-.’*

4. **Stopword/controlled vocabulary filtering:**

   D1: *‘text-mine’, ‘to-identify’, ‘identify-use’, ‘use-inform’*

Reducer

Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0

Documents in a vector space!

How to assign weights?

- Important!
- Why?
  - Corpus-wise: some terms carry more information about the document content
  - Document-wise: not all terms are equally important
- How?
  - Two basic heuristics
    - TF (Term Frequency) = Within-doc-frequency
    - IDF (Inverse Document Frequency)

Term frequency

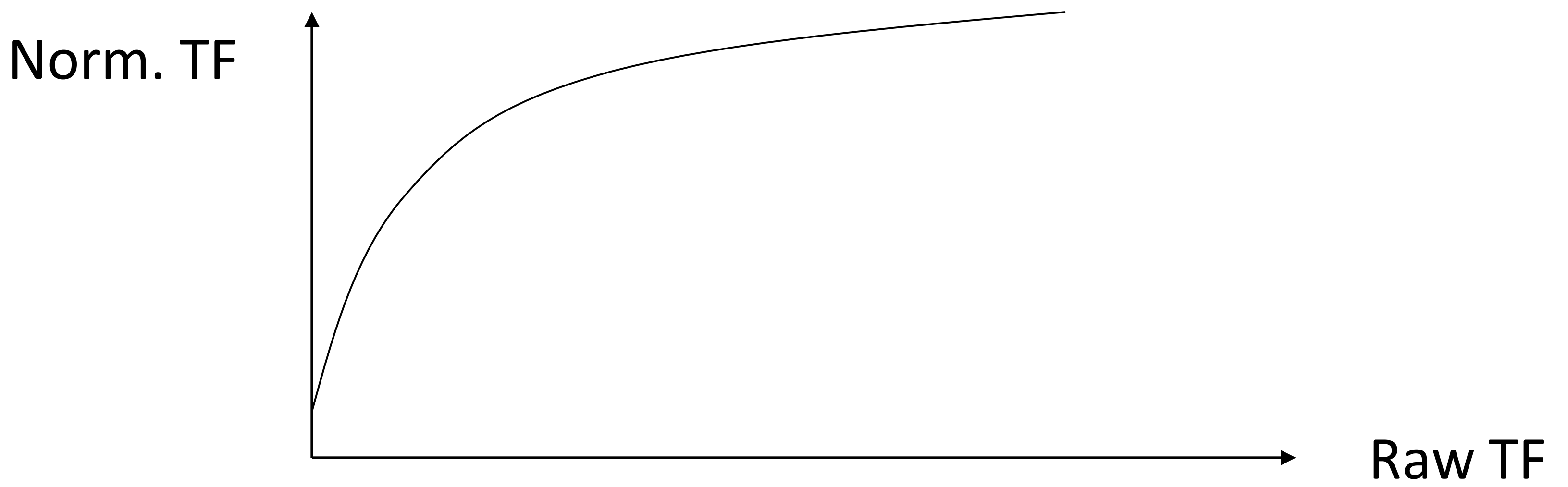
- Idea: a term is more important if it occurs more frequently in a document
- TF Formulas
  - Let $c(t,d)$ be the frequency count of term t in doc d
  - Raw TF: $tf(t,d) = c(t,d)$

TF normalization

- Two views of document length
  - A doc is long because it is verbose
  - A doc is long because it has more content
- Raw TF is inaccurate
  - Document length variation
  - "Repeated occurrences" are less informative than the "first occurrence"
  - Information about semantic does not increase proportionally with number of term occurrence
- Generally penalize long document, but avoid over-penalizing
  - Pivoted length normalization

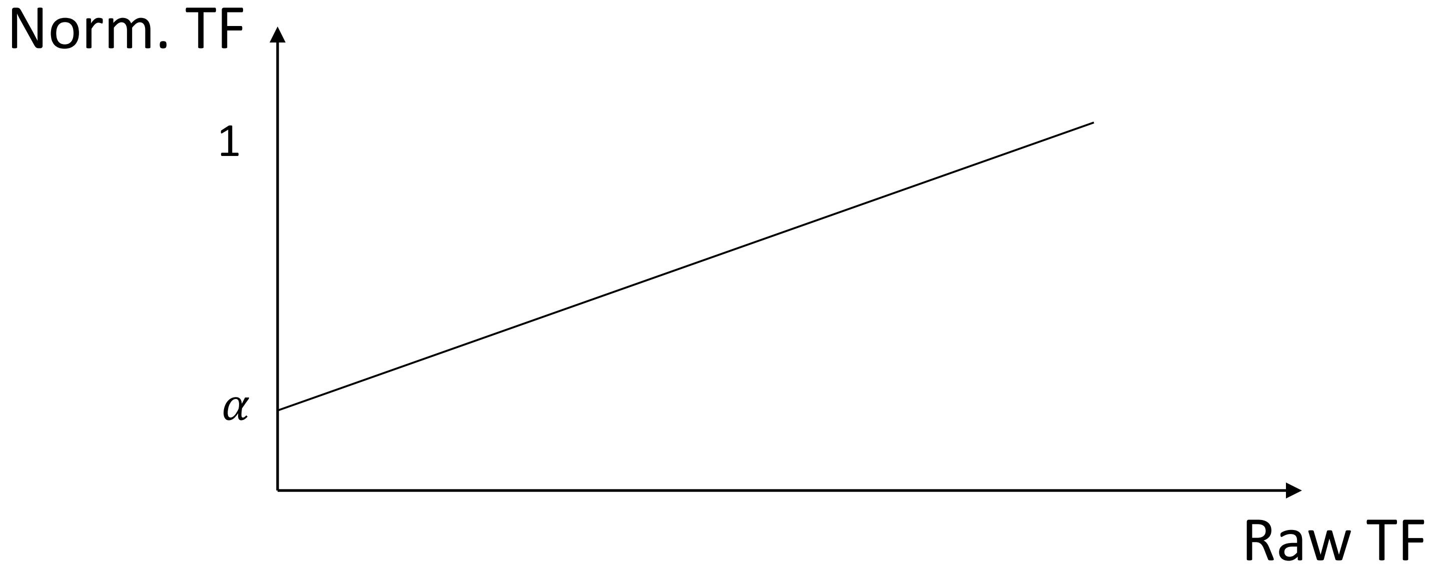
TF normalization

- Sub-linear TF scaling
  - $tf(t,d) = \begin{cases} 1 + \log c(t,d), if\ c(t,d) > 0 \\ 0, otherwise \end{cases}$



TF normalization

- Maximum TF scaling
  - $tf(t,d) = \alpha + (1-\alpha)\frac{c(t,d)}{\max_t c(t,d)}$
  - Normalize by the most frequent word in this doc



Document frequency

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

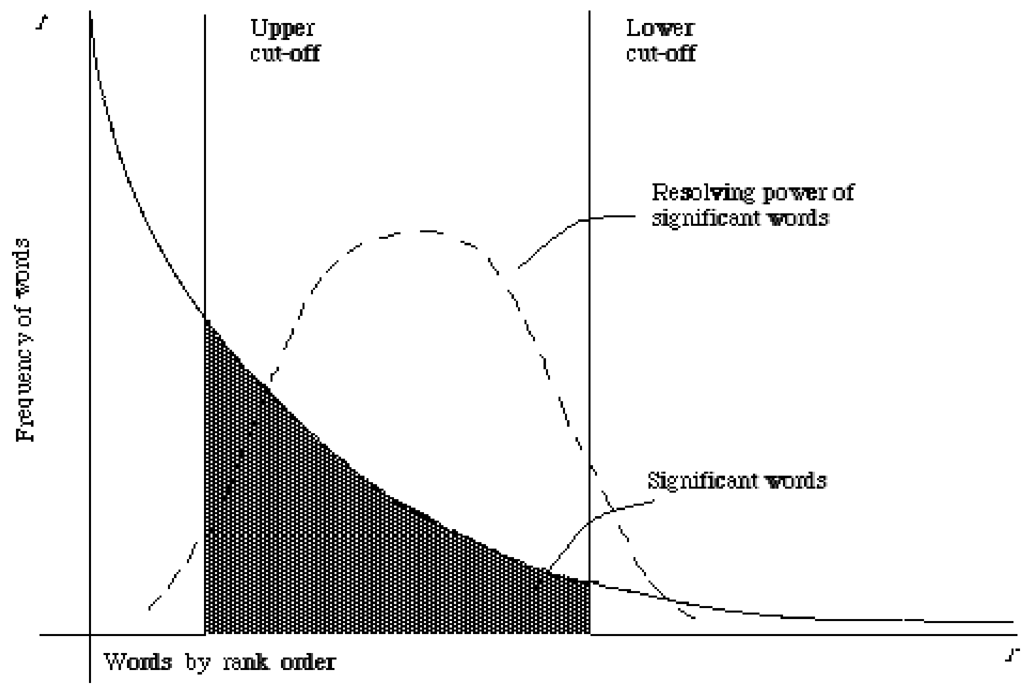


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

Inverse document frequency

- Solution
  - Assign higher weights to the rare terms
  - Formula
    - $IDF(t) = 1 + \log(\frac{N}{df(t)})$

Non-linear scaling

Total number of docs in collection

Number of docs containing term t

  - A corpus-specific property
    - Independent of a single document

Why document frequency

- How about total term frequency?
  - $ttf(t) = \sum_d c(t, d)$

Table 1. Example total term frequency v.s.
document frequency in Reuters-RCV1 collection.

Word	ttf	df
try	10422	8760
insurance	10440	3997

  - Cannot recognize words frequently occurring in a subset of documents

TF-IDF weighting

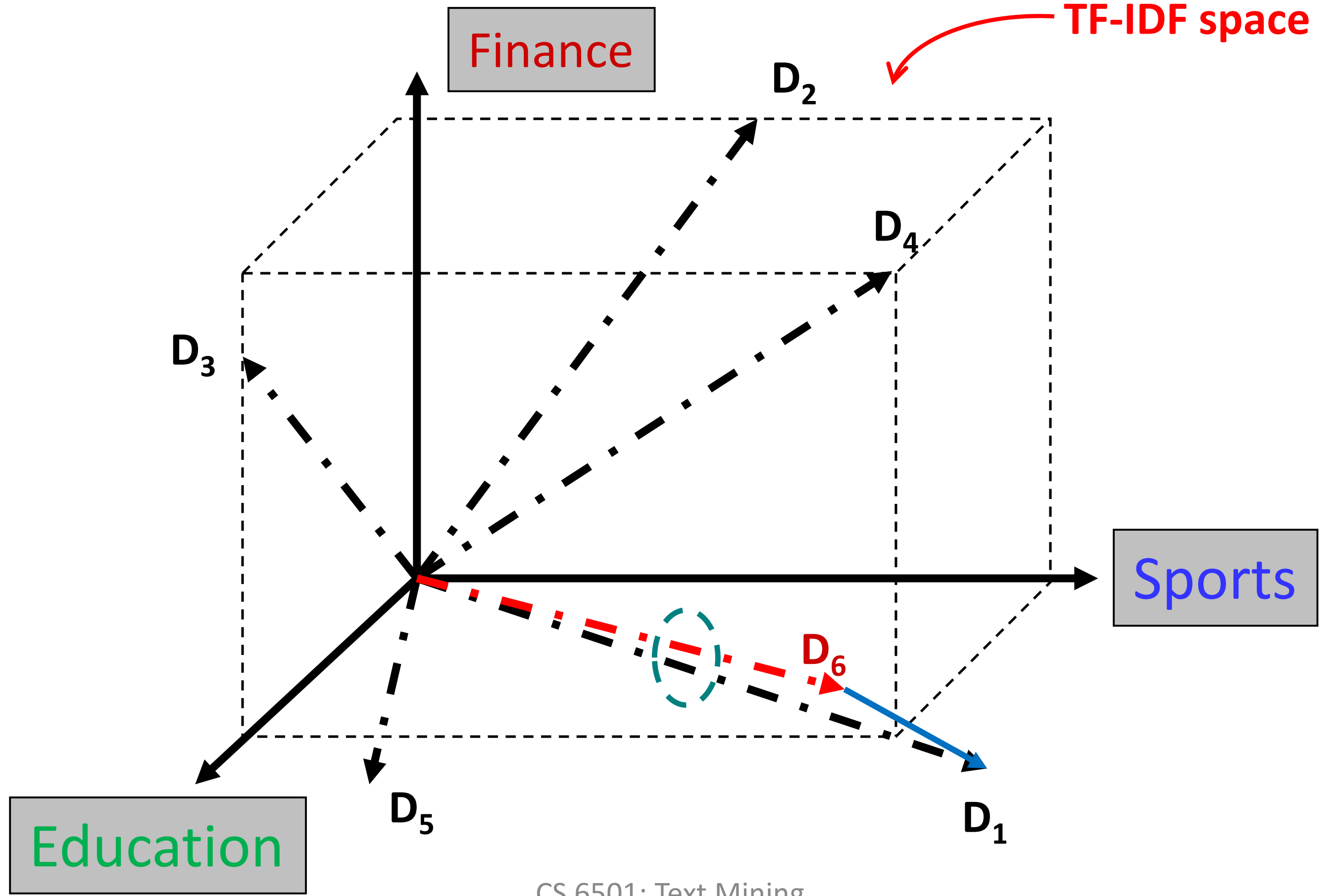
- Combining TF and IDF
  - Common in doc → high tf → high weight
  - Rare in collection→ high idf→ high weight
  - $w(t,d) = TF(t,d) \times IDF(t)$
- Most well-known document representation schema in IR! (G Salton et al. 1983)

"Salton was perhaps the leading computer scientist working in the field of information retrieval during his time." - wikipedia

Gerard Salton Award – highest achievement award in IR

How to define a good similarity metric?

- Euclidean distance?

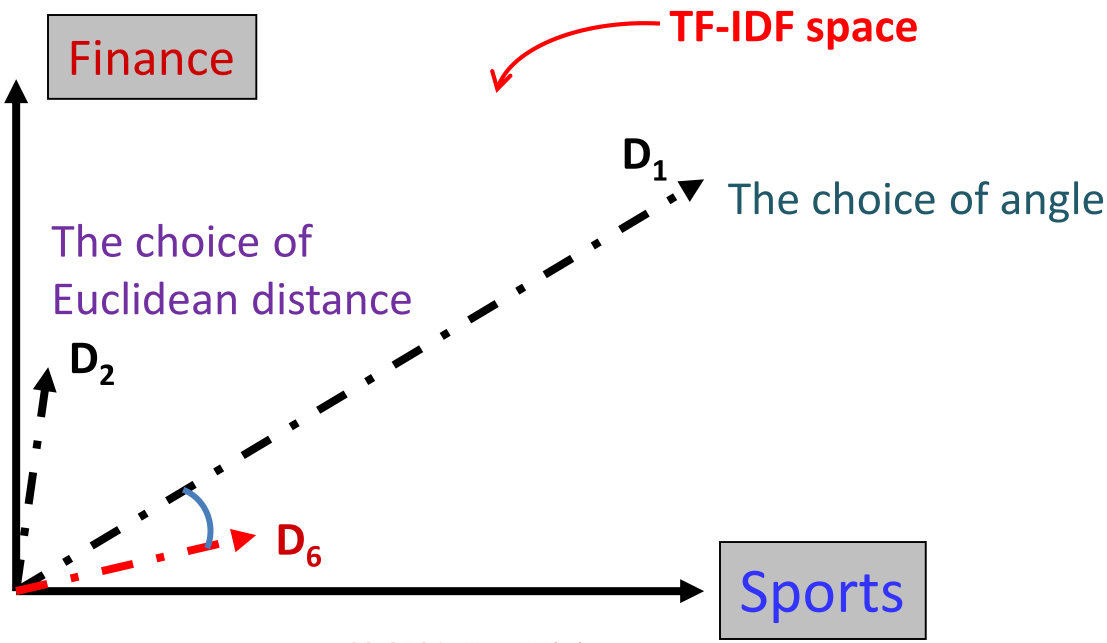


How to define a good similarity metric?

- Euclidean distance
  - $dist(d_i, d_j) = \sqrt{\sum_{t \in V} [tf(t, d_i) idf(t) - tf(t, d_j) idf(t)]^2}$
  - Longer documents will be penalized by the extra words
  - We care more about how these two vectors are overlapped

From distance to angle

- Angle: how vectors are overlapped
  - Cosine similarity – projection of one vector onto another

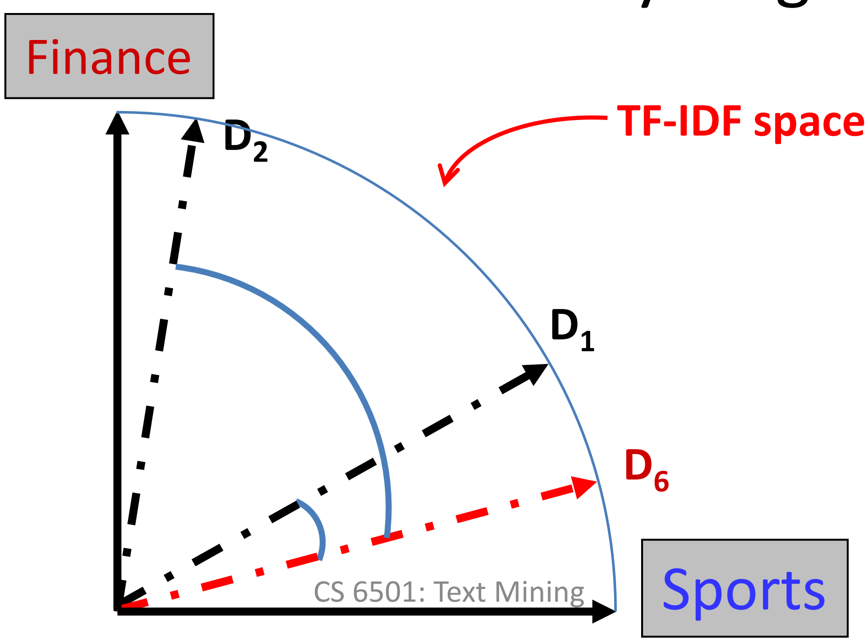


Cosine similarity

- Angle between two vectors
  - $cosine(d_i, d_j) = \frac{V_{d_i} \times \boxed{V_{d_j}}}{|V_{d_i}|_2 \times \boxed{|V_{d_j}|_2}}$

    TF-IDF vector

    Unit vector
  - Documents are normalized by length



Advantages of VS model

- Empirically effective!
- Intuitive
- Easy to implement
- Well-studied/mostly evaluated
- The Smart system
  – Developed at Cornell: 1960-1999
  – Still widely used
- Warning: many variants of TF-IDF!

Disadvantages of VS model

- Assume term independence
- Lack of “predictive adequacy”
  - Arbitrary term weighting
  - Arbitrary similarity measure
- Lots of parameter tuning!

What you should know

- Basic ideas of vector space model
- Procedures of constructing VS representation for a document
- Two important heuristics in bag-of-words representation
  - TF
  - IDF
- Similarity metric for VS model

Today's reading

- Introduction to information retrieval
  - Chapter 2.2: Determining the vocabulary of terms
  - Chapter 6.2: Term frequency and weighting
  - Chapter 6.3: The vector space model for scoring
  - Chapter 6.4: Variant tf-idf functions