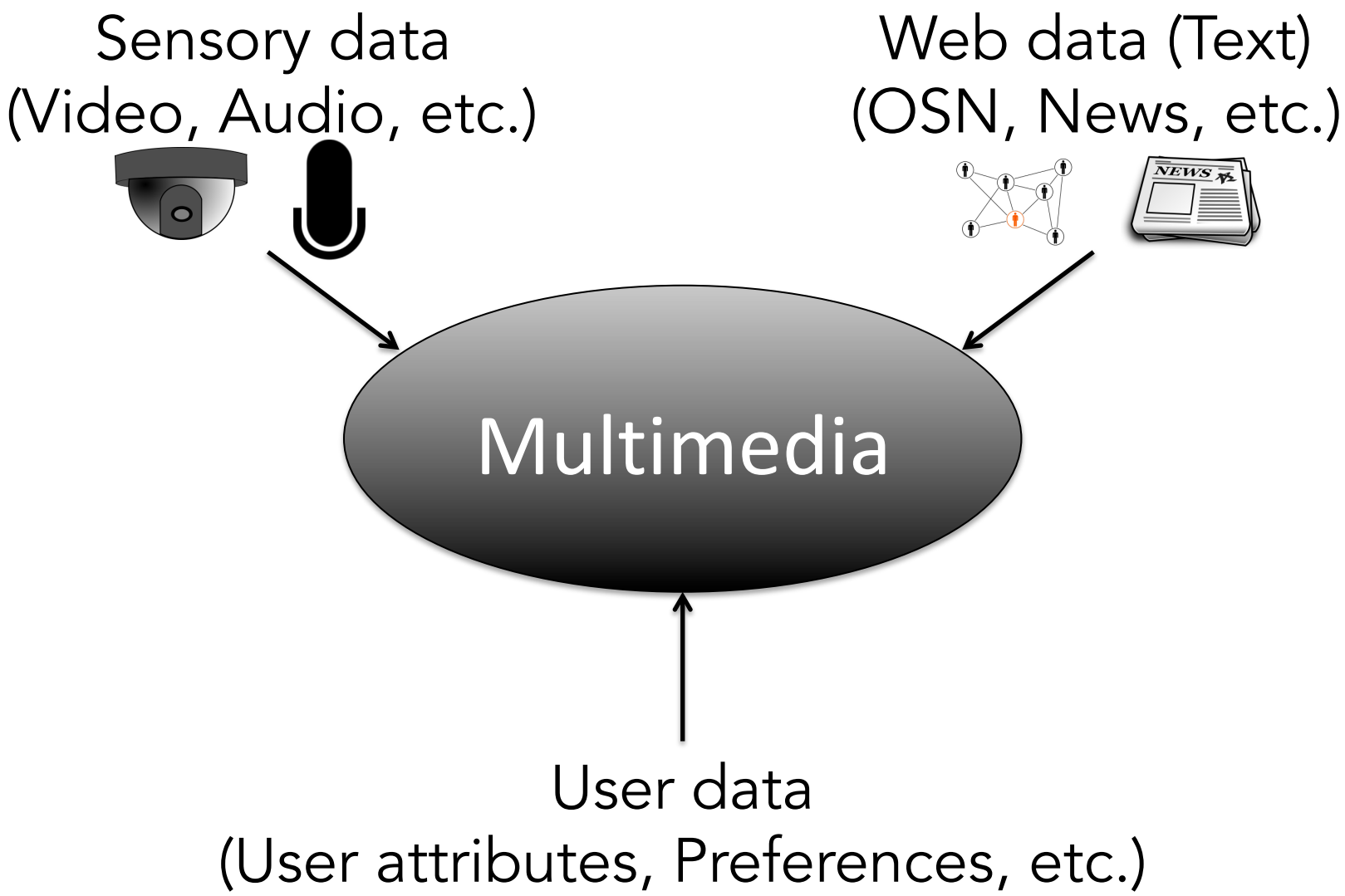
Week 5

Text Analysis

Reference and Slide Source: ChengXiang Zhai and Sean Massung. 2016. Text Data Management and Analysis: a Practical Introduction to Information Retrieval and Text Mining. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.

What is Multimedia?

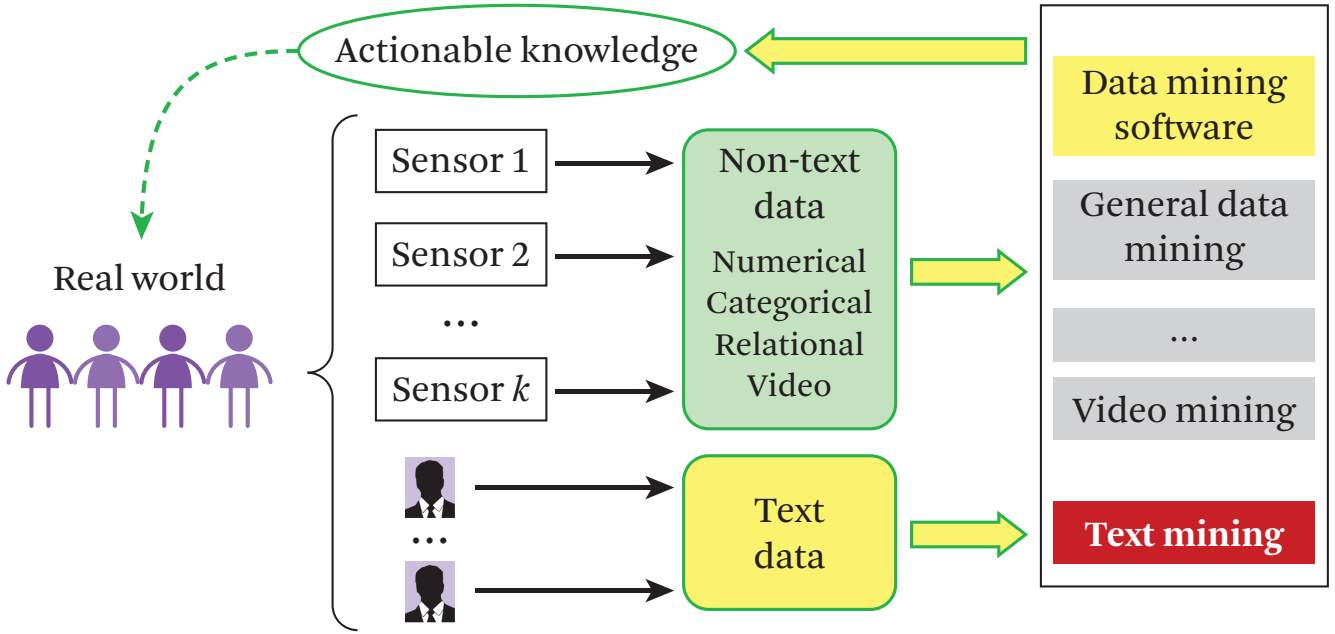
Multiple Carriers of information


Sensory data
(Video, Audio, etc.)
Web data (Text)
(OSN, News, etc.)
NEWS
Multimedia
User data
(User attributes, Preferences, etc.)

Main Topics

- Audio
- Text
- Video
- Information Fusion
- Case Studies

Text Mining in Multimedia

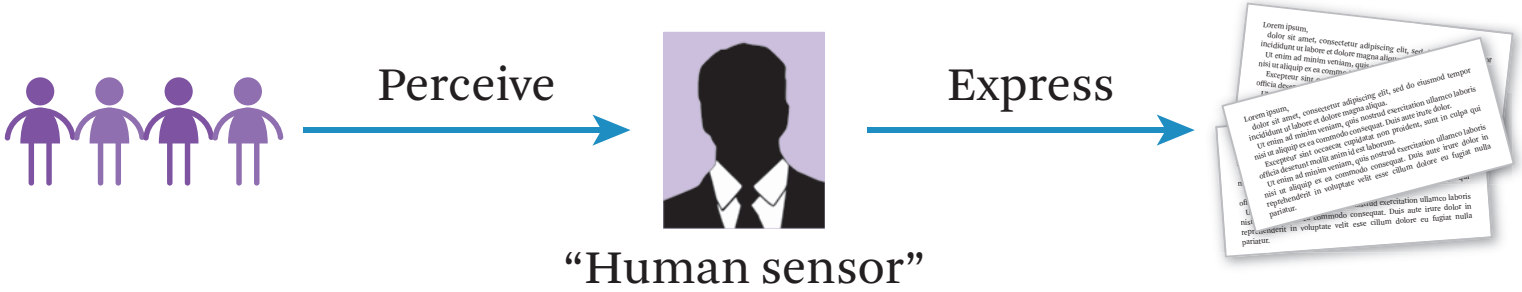


Humans as Sensors

Weather ---------> Thermometer ---------> 3°C, 15°F, ...

Locations ---------> Geo sensor ---------> 41°N and 120°W ...

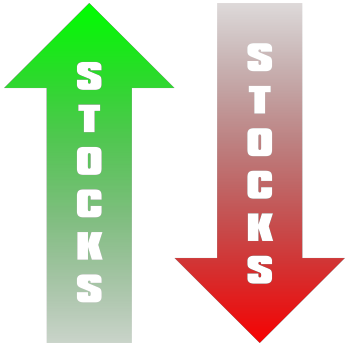
Networks ---------> Network sensor ---------> 01000100011100



"Human sensor"

Knowledge of the observed world

- Events and activities
- Market trend
- Prediction of future events
- Product popularity
- Ratings/reviews



Knowledge about the Observer

- Preferences
- Sentiments
- Mood

Text Analysis is also called Text Mining!

Text Mining

- Information Retrieval (IR)
- Natural Language Processing (NLP)
- Concept Extraction
- Web Mining
- Document Clustering
- Document Classification

How to "Capture" Text?

1. Offline documents
2. Online on the web
3. From speech signal

Capturing text from the Web

- Web crawling/Web scrapping
- APIs
- JSON/CSV

Challenges in Text Analysis

- Documents are described with many attributes
- The attributes are either characters, words, or phrases

How to Represent Text?

Digital Representation

- ASCII Characters
  - American Standard Code for Information Interchange
  - 7-bits
- EBCDIC Characters
  - Extended Binary Coded Decimal Interchange Code
  - 8-bits

How to Represent for Text Analysis?

Let’s take a sentence

“A dog is chasing a boy on the playground.”

Idea 1: String of characters!

- Most general and simple approach
- But semantic analysis is difficult
- Words have more meaning than characters

Idea 2: Sequence of words!

A dog is chasing a boy on the playground

Advantages

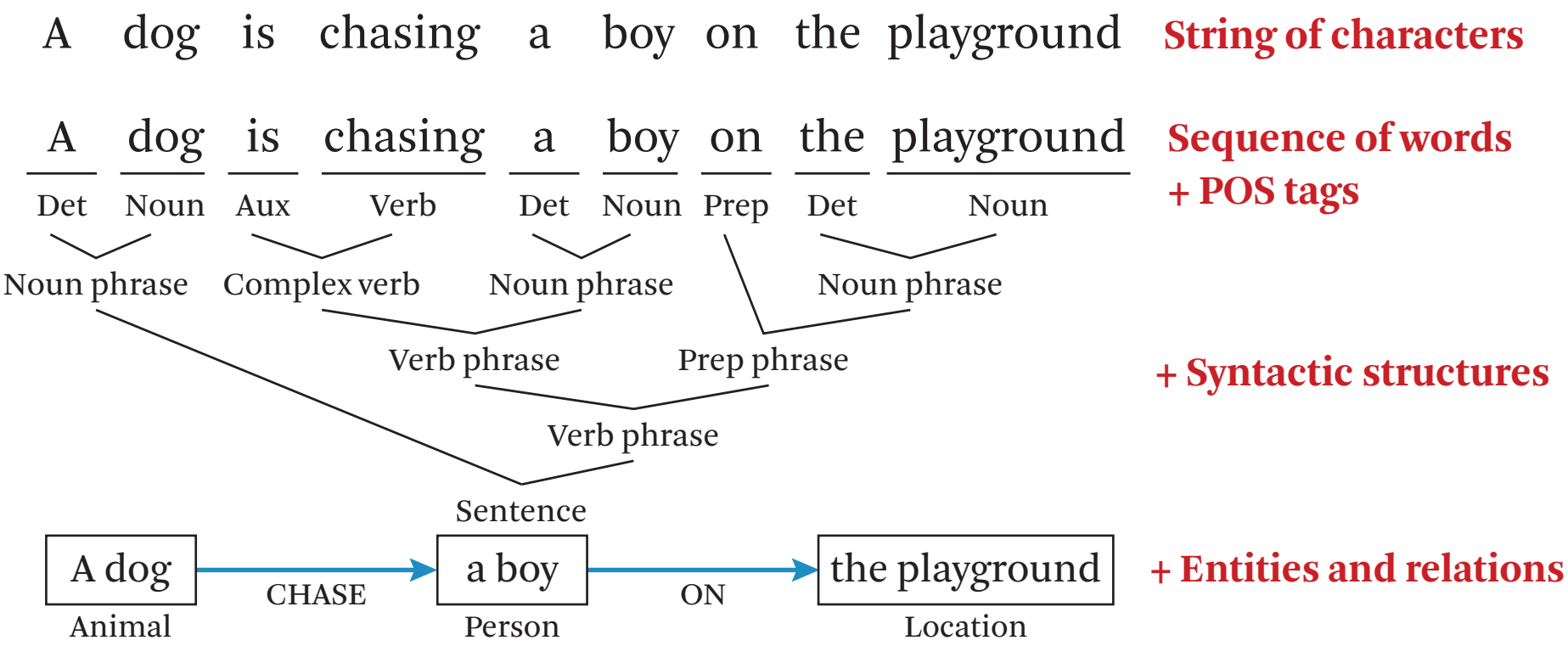
- Easily discover most frequent words
- Most frequent words lead to topics
- Many other analysis possibilities

Challenges

- In some languages, e.g. Chinese, it is difficult to identify word boundaries
- All words are treated equally, nouns, adjectives, verbs all are same
- This may limit the semantic analysis

Idea 3: Part of Speech (POS) Tagging!

+Noun, Verb, Determiners, etc.



Deeper analysis requires more efforts!

POS Tagging Challenges

- Even less robust than sequence of words
- Difficult to identify all entities with the right types
- Relations are even harder to find

How would you differentiate a love letter from a research paper?

- Love
- Like
- Feel
- I, you, this, that
- …

- Challenge
- Method
- Application
- We, you, this, that
- …

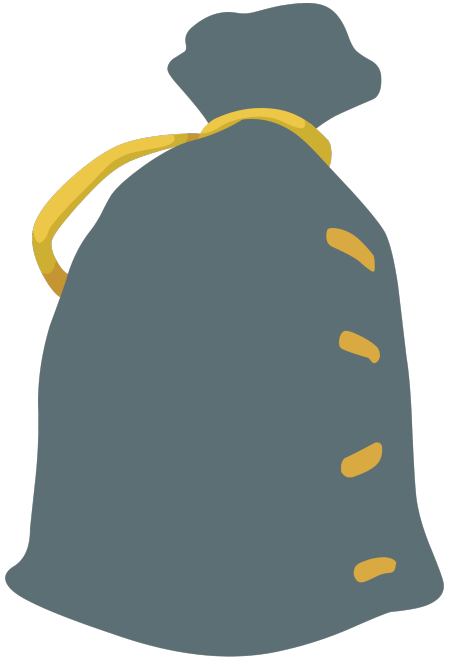
Idea 4: Count the word frequencies for each document!

Example

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Bag-of-Words (BoW)

Consider the document as a bag/set of words!



What assumptions are we making?

1. Words are mutually independent
2. Word order in text is irrelevant

BoW is a Vector Space Model!

Dimensions

- Each term represents one dimension of the vector space
- The term can be a single word or a sequence of words (phrase)
- The number of terms determines the dimension of the vector space

Vector Elements

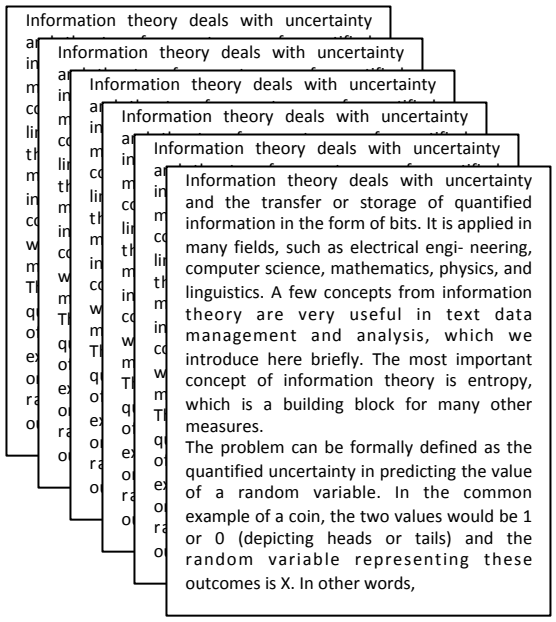
- The vector elements are weights associated with each term
- These weights reflect the relevance of the term
- If the corpus contains n terms, a document is represented as

$d = \{w_1, w_2, \ldots w_n\}$

Term Document Matrix (TDM)

- An *m* x *n* matrix with following features:
  - Rows (*i=1,m*) represent terms from the corpus
  - Columns (*j=1,n*) represent documents from the corpus
  - Cell *ij* stores the weight of the term *i* in the context of the document *j*

VSM: Term Document Matrix



	D1	D2	D3	D4	D5
complexity	0.2	0	0.6	0.2	0.1
algorithm	0.3	0	0	4	0.4
entropy	0.1	0	0	0	0.2
traffic	0.2	3	0	0	0
network	0	0	0.1	0.4	0

Text Preprocessing

Objective: to choose most relevant words for the corpus – also called a dictionary!

1. Transform various forms of the terms in a common normalized form

- Transform all words to lower case
- Use a dictionary to replace synonyms with a common, general term

Examples

- Apple, apple, APPLE -> apple
- Intelligent Systems, Intelligent systems, Intelligent-systems -> intelligent systems
- “automobile, car” -> vehicle

2. Remove High Frequency Terms

- Very high frequency terms are semantically almost useless
- Examples: the, a, an, we, do, to

3. Remove Low Frequency Terms

- Very low frequency terms are semantically rich, but not representative of the class
- Examples: dextrosinistral

The rest of the words are those that represent the corpus the best!

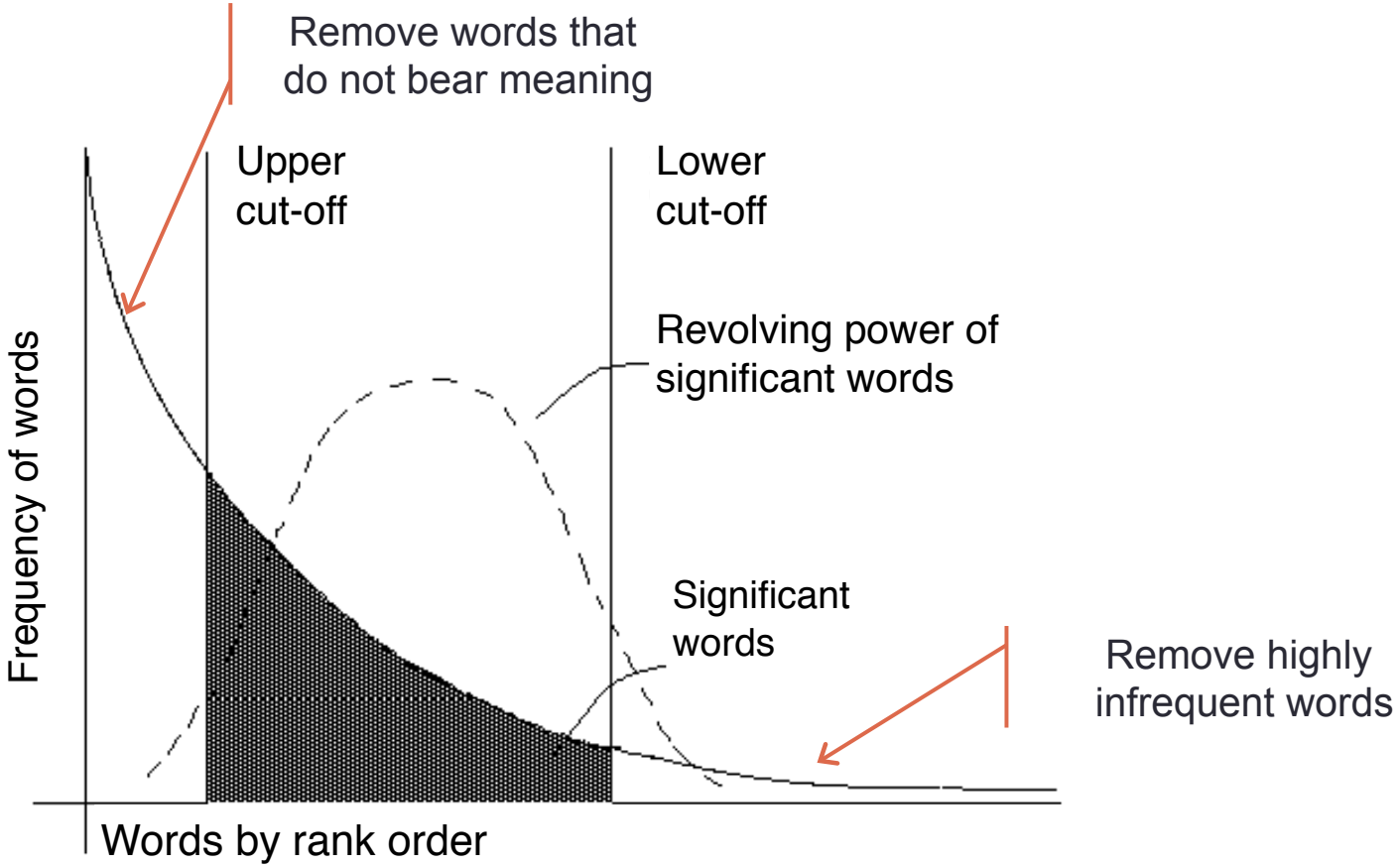


Image source:
http://www.dcs.gla.ac.uk/Keith/Chapter.2/Ch.2.html

4. Stop Words

- Stop-words are those words that (on their own) do not bear any information / meaning
- It is estimated that they represent 20-30% of words in any corpus
- There is no unique stop-words list
- Frequently used lists are available at:
  - http://www.ranks.nl/stopwords

Removing words causes loss of meaning and structure!

- “this is not a good option” -> “option”
- “to be or not to be” -> null

5. Lemmatization & Stemming to reduce the variability of words

Stemming

- It is a crude heuristic process that chops off the ends of words to get to a basic form
- Does not consider linguistic features of the words
- Normalizes verb tenses

Examples

- walking, walks, walked, walker => walk
- argue, argued, argues, arguing => argu
- apply, applications, reapplied => apply
- denormalization => norm

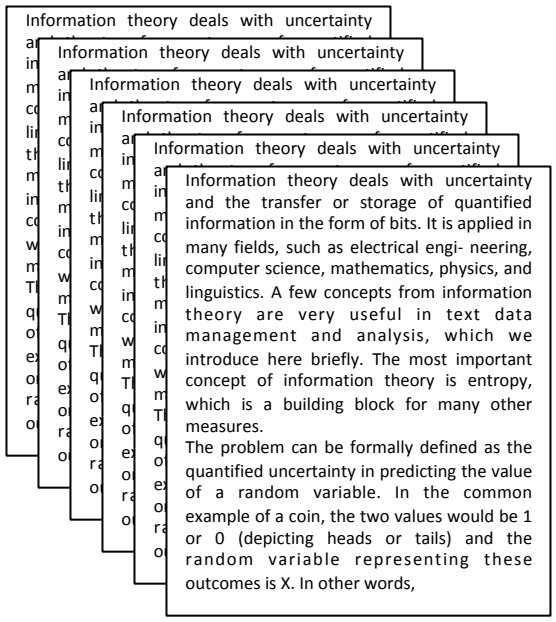
Lemmatization

- Use vocabulary and morphological analysis of words to get the base or dictionary form of a word
- Base form is also known as the lemma
- E.g., argue, argued, argues, arguing -> argue

Summary of Preprocessing Steps

1. Transform various forms of the terms in a common normalized form
2. Remove high frequency terms
3. Remove low frequency terms
4. Remove stop-words
5. Lemmatization & Stemming

How do we compute term weights?



	D1	D2	D3	D4	D5
complexity					
algorithm					
entropy					
traffic					
network					

Idea 1: take the value of 0 or 1, to reflect the presence (1) or absence (0) of the term in a particular document!

Example

Doc1: Text mining is to identify useful information.

Doc2: Useful information is mined from text.

Doc3: Apple is delicious.

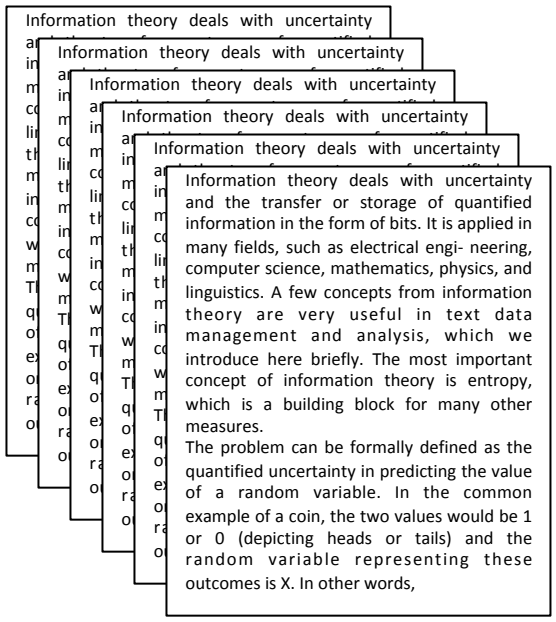
	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

Observation: few terms occur more often in a specific class of documents!

Idea 2: measure the term frequency (TF) in a specific document!

Assumption: If a term occurs more often in a doc, it measures something important for that doc!

Term Frequency (TF)



	D1	D2	D3	D4	D5
complexity	2	0	6	2	1
algorithm	3	0	0	4	4
entropy	1	0	0	0	2
traffic	2	3	0	0	0
network	0	0	1	4	0

Example

- Task: Separate technical and non-technical document
- ‘job’ and ‘engineer’ both occur 100 times in a corpus of 100 documents
- Which term is more important?

Most likely, the term ‘job’ appears in most documents while ‘engineer’ occurs mainly in technical documents!

Observation: generally words not so common in the corpus are more important!

Idea 3: assign higher weights to terms occurring in fewer documents!

Inverse Document Frequency (IDF)

$$\mathrm{IDF}_t = 1 + \log\left(\frac{N}{N_t}\right)$$

N = Number of documents

N_t = Number of documents with term t

Note - IDF is computed at the corpus level for each term

A rare term has high idf and a frequent term has low idf!

Observation: terms that are not so common in the corpus, but still have same reasonable level of frequency are important!

TF-IDF

- Most frequently used weight matrix
- Generally computer as follows:

$$\text{TF-IDF}_t = \text{TF} \times \text{IDF}_t$$

Text Retrieval Example

Query = "news about presidential campaign"

d_1 … news about …

d_2 … news about organic food campaign …

d_3 … news of presidential campaign …

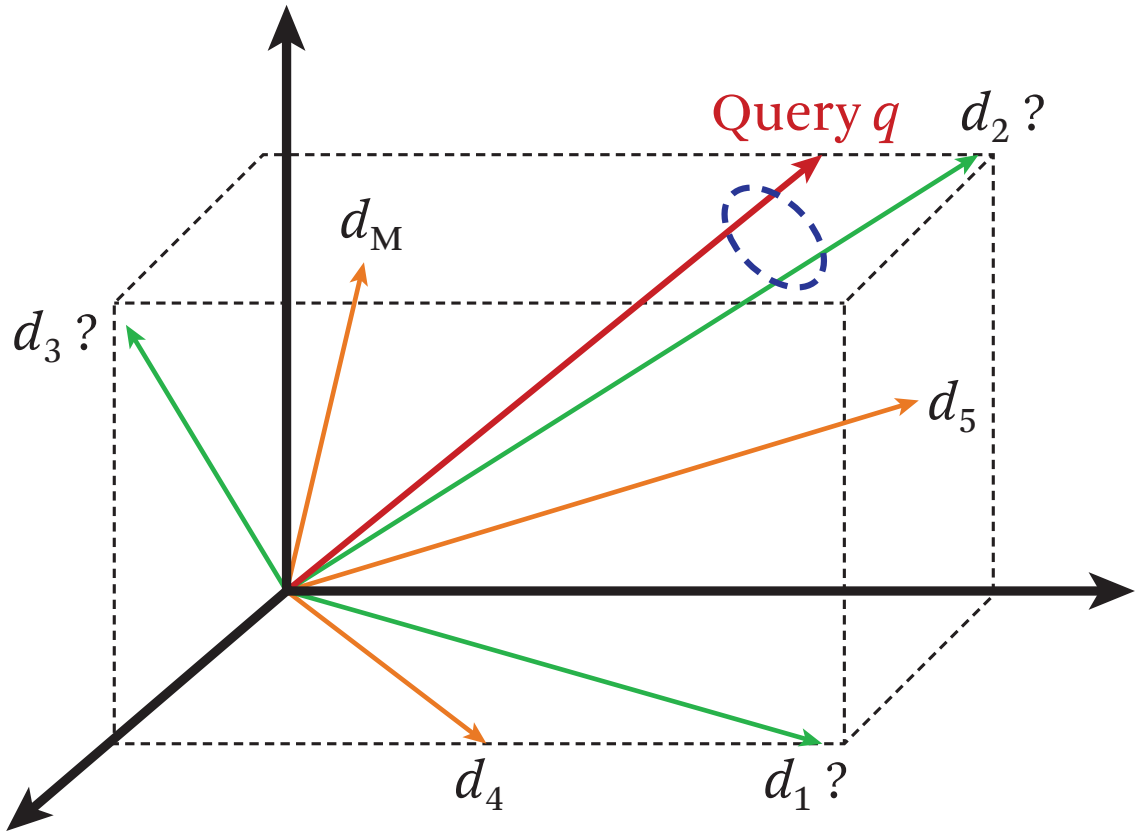
d_4 … news of presidential campaign …
… presidential candidate …

d_5 … news of organic food campaign …

Text Retrieval Example

1. Define the dimensions
2. Define how the vectors are obtained
3. Define similarity between vectors

The goal is to find the closest document in the corpus!



Dot product similarity with bit vector representation

$$q = (x_1, \ldots, x_N)$$

$$d = (y_1, \ldots, y_N)$$

$x_i, y_i \in \{0, 1\}$ where

1: word W_i is **present**

0: word W_i is **absent**

$$Sim(q, d) = q.d = x_1 y_1 + \cdots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

Dot product similarity with bit vector representation

Query = "news about presidential campaign"

		Similarity score
d_1	… news about …	2
d_2	… news about organic food campaign …	3
d_3	… news of presidential campaign …	3
d_4	… news of presidential campaign … … presidential candidate …	3
d_5	… news of organic food campaign …	2

Problems

- d_2, d_3, d_4 all are scored same
- d_2 should be ranked lower than d_3 and d_4
- d_4 should score more than d_3 because d_4 mentioned presidential more times

Dot product similarity with Term Frequency representation

$q = (x_1, \ldots, x_N)$ $\quad x_i = \text{count of word } W_i \text{ in query}$

$d = (y_1, \ldots, y_N)$ $\quad y_i = \text{count of word } W_i \text{ in doc}$

$$Sim(q, d) = q.d = x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i$$

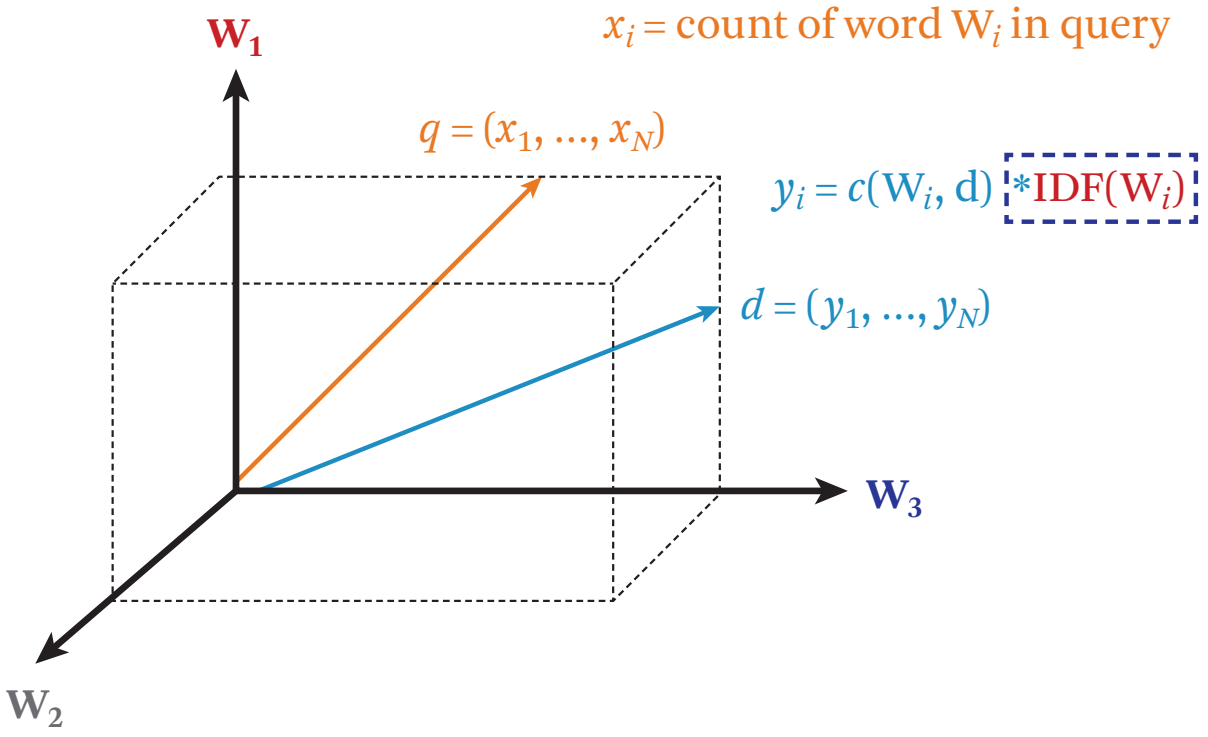
Dot product similarity with Term Frequency representation

Query = "news about presidential campaign"		Similarity score
d_1	… news about …	2
d_2	… news about organic food campaign …	3
d_3	… news of presidential campaign …	3
d_4	… news of presidential campaign … … presidential candidate …	4
d_5	… news of organic food campaign …	2

Problems

- d_2 and d_3 have the same scores
- d_2 should be ranked lower than d_3 and d_4

Dot product similarity with TF-IDF representation



Dot product similarity with TF-IDF representation

Query = "news about presidential campaign"		Similarity score
d_1	… news about …	2.5
d_2	… news about organic food campaign …	5.6
d_3	… news of presidential campaign …	7.1
d_4	… news of presidential campaign … … presidential candidate …	9.6
d_5	… news of organic food campaign …	4.6