Text Mining

Fátima Rodrigues
mfc@isep.ipp.pt
Departamento de Engenharia Informática (DEI/ISEP)

What is Text Mining?

- Text Mining
  - extract high quality (previously unknown) information from large amounts of unstructured text

- Text mining is also known as Text Data Mining (TDM) and Knowledge Discovery in Textual Database (KDT)
- A process of identifying novel information from a collection of texts (also known as a corpus)

Text Mining: Examples

- Text mining is an exercise to gain knowledge from stores of language text
- Text:
  - Web pages
  - Medical records
  - Customer surveys
  - Email filtering (spam)
  - DNA sequences
  - Incident reports
  - Drug interaction reports
  - News stories (e.g. predict stock movement)

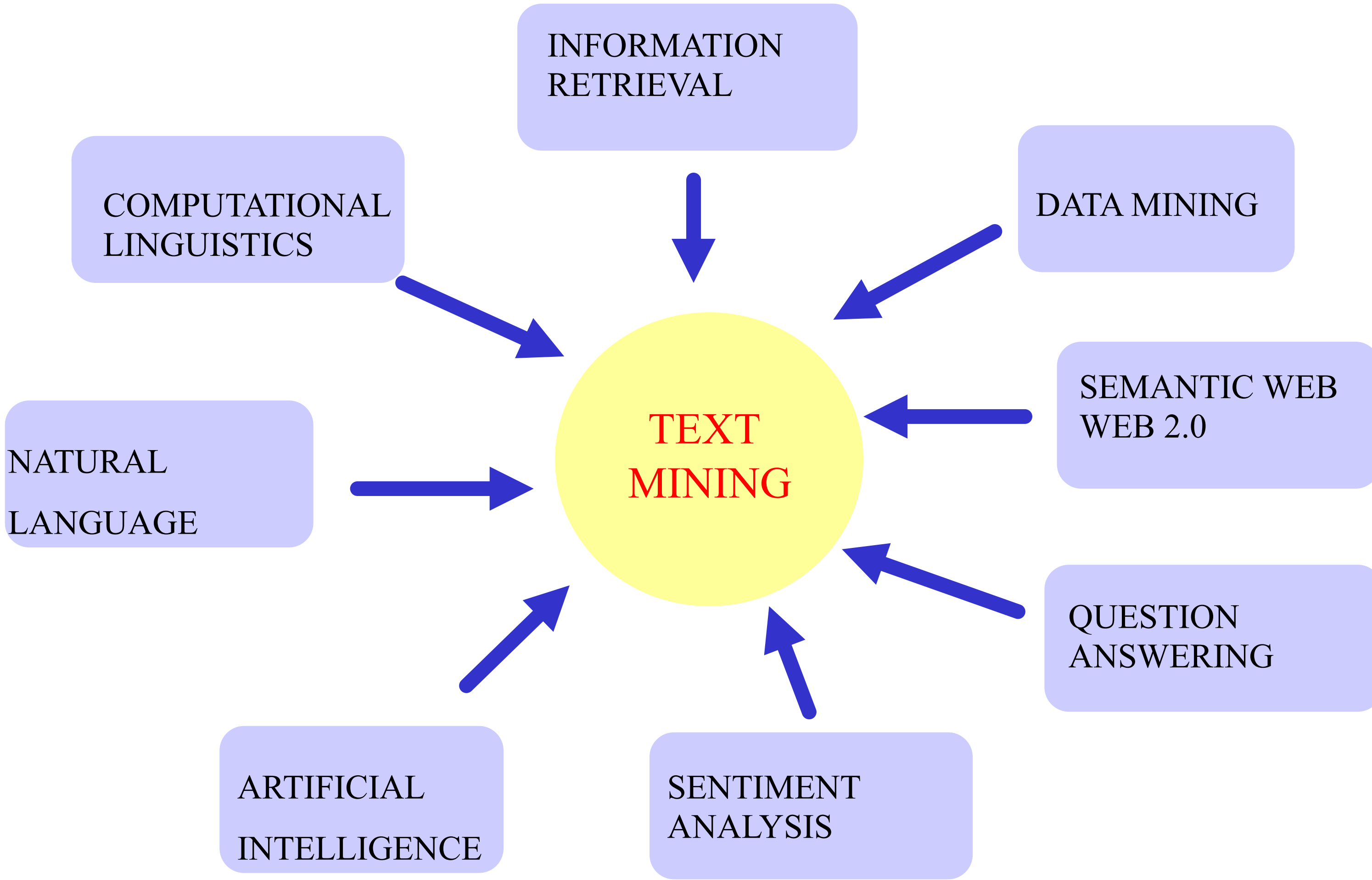
Data Mining vs. Text Mining

- Data Mining
  - process directly
  - Identify causal relationship
  - Structured numeric transaction data residing in files
- Text Mining
  - Linguistic processing or natural language processing (NLP)
  - Discover heretofore unknown information
  - Applications deal with much more diverse and eclectic collections of systems and formats

Why dealing with text is difficult?

- Abstract concepts are **difficult to represent**
- **“Countless” combinations** of subtle, abstract relationships among concepts
- **Many ways** to represent similar concepts, e.g. space ship, flying saucer, UFO
- Concepts are **difficult to visualize**
- **High dimensionality**
- **Tens or hundreds of thousands of features**

Several areas are related to Text Mining



Levels of Text Representation

- Character (character n-grams and sequences)
- Words (stop-words, stemming, lemmatization)
- Phrases (word n-grams, proximity features)

Word Level

- The most common representation of text used for many techniques
  - ...there are many tokenization software packages which split text into the words

Word Level Representation

- **Basic Concepts**
  - A document is described by a set of representative keywords called index terms
  - Different index terms have varying relevance when used to describe document contents
  - This effect is captured through the assignment of numerical weights to each index term of a document. (e.g.: frequency, tf-idf)

- DBMS Analogy
  - Index Terms → Attributes
  - Weights → Attribute Values

Words Properties

- Relations among word surface forms and their senses:
  - **Homonomy**: same form, but different meaning (e.g. bank: river bank, financial institution)
  - **Polysemy**: same form, related meaning (e.g. bank: blood bank, financial institution)
  - **Synonymy**: different form, same meaning (e.g. singer, vocalist)
  - **Hyponymy**: one word denotes a subclass of an another (e.g. breakfast, meal)

- Word frequencies in texts have **power distribution**:

  ...small number of very frequent words

  ...big number of low frequency words

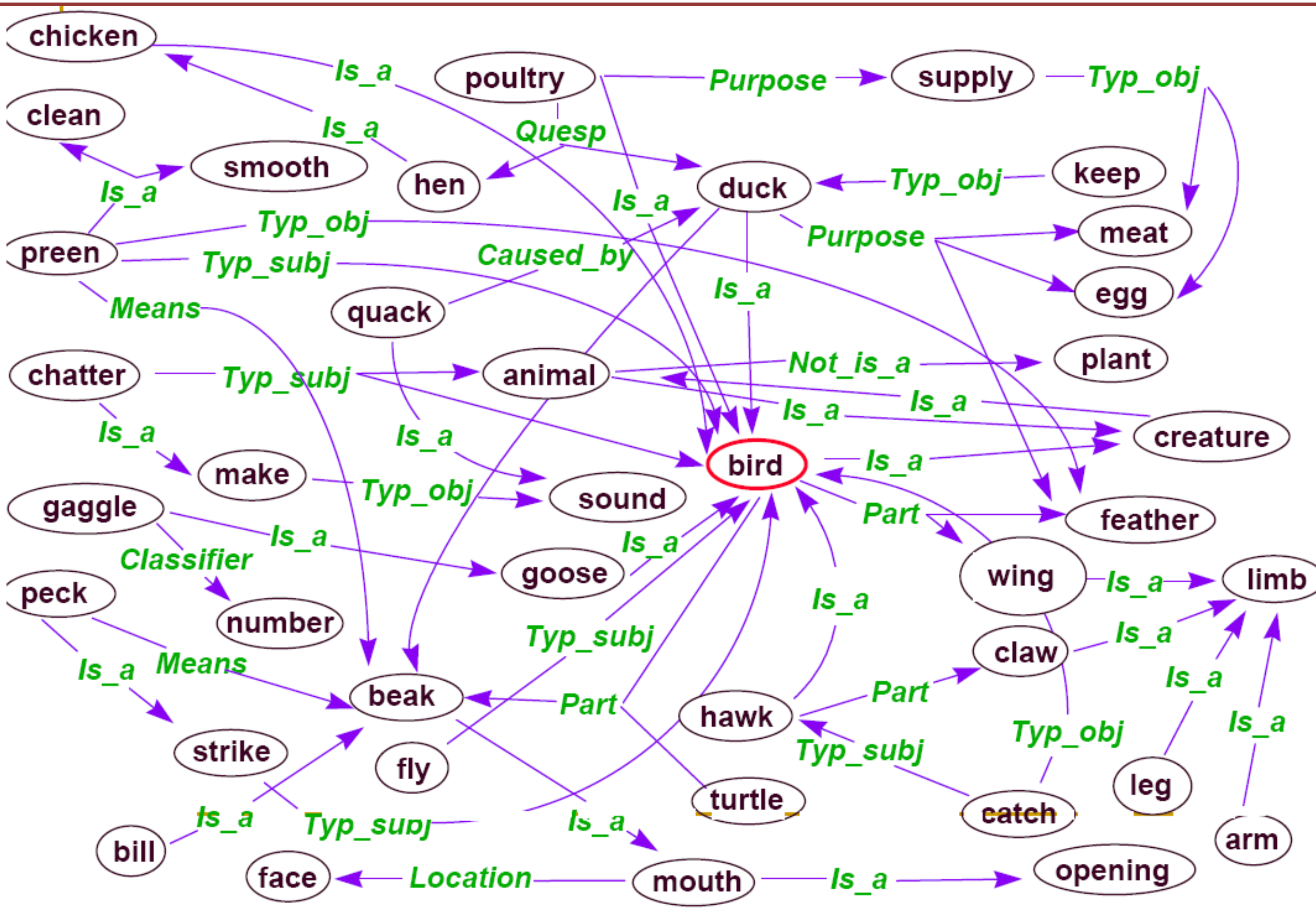
Taxonomies/thesaurus level

- Thesaurus has a main function to connect different surface word forms with the same meaning into one sense (synonyms)
  - ...additionally we often use hypernym relation to relate general-to-specific word senses
  - ...by using synonyms and hypernym relation we compact the feature vectors
- The most commonly used general thesaurus is **WordNet** which exists in many other languages (e.g. EuroWordNet) http://www.illc.uva.nl/EuroWordNet/

WordNet – database of lexical relations

- WordNet is the most well developed and widely used lexical database for English
  - ... it consist from 4 databases (nouns, verbs, adjectives, and adverbs)
- Each database consists from sense entries – each sense consists from a set of synonyms, e.g.:
  - musician, instrumentalist, player
  - person, individual, someone
  - life form, organism, being

WordNet – excerpt from the graph



WordNet relations

- Each WordNet entry is connected with other entries in the graph through relations
- Relations in the database of nouns:

Relation	Definition	Example
Hypernym	From lower to higher concepts	breakfast -> meal
Hyponym	From concepts to subordinates	meal -> lunch
Has-Member	From groups to their members	faculty -> professor
Member-Of	From members to their groups	copilot -> crew
Has-Part	From wholes to parts	table -> leg
Part-Of	From parts to wholes	course -> meal
Antonym	Opposites	leader -> follower

Document Bag of Words Phases Creation

- Identification of terms (simple or compound)
- Removal of irrelevant words, such as stop-words
- Morphological normalization (stemming)
- Selection of terms

Document Bag of Words Phases Creation

- Identification of terms (simple or compound)
- Removal of irrelevant words, such as stop-words
- Morphological normalization (stemming)
- Selection of terms

Stop-words

- Stop-words are words that from linguistic view do not carry information
  - ...they have mainly functional role
  - ...usually we remove them to help the methods to perform better

- Stop words are language dependent – examples:
  - **English**: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ...
  - **Dutch**: de, en, van, ik, te, dat, die, in, een, hij, het, niet, zijn, is, was, op, aan, met, als, voor, had, er, maar, om, hem, dan, zou, of, wat, mijn, men, dit, zo, ...
  - **Slovenian**: A, AH, AHA, ALI, AMPAK, BAJE, BODISI, BOJDA, BRŽKONE, BRŽČAS, BREZ, CELO, DA, DO, ...

Stop words

- For an application, an additional domain specific stop words list may be constructed

- Why do we need to remove stop words?
  - Reduce indexing (or data) file size
    - stopwords accounts 20-30% of total word counts.
  - Improve efficiency
    - stop words are not useful for searching or text mining
    - stop words always have a large number of hits

Stemming

- Different forms of the same word are usually problematic for text data analysis, because they have different spelling and similar meaning (e.g. learns, learned, learning,...)

- Stemming is a process of transforming a word into its stem (normalized form)
  - ...stemming provides an inexpensive mechanism to merge

Stemming

For English is mostly used Porter stemmer at
http://www.tartarus.org/~martin/PorterStemmer/

Example cascade rules used in English Porter stemmer

- ATIONAL -> ATE relational -> relate
- TIONAL -> TION conditional -> condition
- ENCI -> ENCE valenci -> valence
- ANCI -> ANCE hesitanci -> hesitance
- IZER -> IZE digitizer -> digitize
- ABLI -> ABLE conformabli -> conformable
- ALLI -> AL radicalli -> radical
- ENTLI -> ENT differentli -> different
- ELI -> E vileli -> vile
- OUSLI -> OUS analogousli -> analogous

Stemming

- Techniques used to find out the root/stem of a word:
  - E.g.,

– user	engineering
– users	engineered
– used	engineer
– using	

- stem: use engineer

Usefulness

- improving effectiveness of retrieval and text mining
  - matching similar words
- reducing indexing size
  - combing words with same roots may reduce indexing size as much as 40-50%

Bag-of-Tokens representation

Documents

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or …

Feature Extraction

Token Sets

nation – 5
civil - 1
war – 2
men – 2
died – 4
people – 5
Liberty – 1
God – 1
…

Loses all order-specific information!
Severely limits context!

Word weighting

In the bag-of-words representation each word is represented as a separate variable having numeric weight (importance)

The most popular weighting schema is normalized word frequency TF-IDF

Term frequency

- The term frequency $tf_{t,d}$ of term t in document d is defined as the number of times that t occurs in d

- Raw term frequency has some drawbacks:
  - A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term
  - But not 10 times more relevant
  - Relevance does not increase proportionally with term frequency

Term frequency (tf) weight

- There are many variants for tf weight, where log-frequency weighting is a common one, dampening the effect of raw tf (raw count)

$$\log tf_{t,d} \begin{cases} 1 + \log 10\ \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.
- The score is 0 if none of the query terms is present in the document

Document frequency

- Rare terms are more informative than frequent terms
- Consider a term in the query that is rare in the collection (e.g., *arachnocentric*)
- A document containing this term is very likely to be relevant to the query *arachnocentric*

  → Rare terms like *arachnocentric must have* a high weight

Document frequency

- Consider a query term that is frequent in the collection (e.g., *high, increase, line*)
- A document containing such a term is more likely to be relevant than a document that doesn't, but it's not a sure indicator of relevance
- For frequent terms, we want positive weights for words like *high, increase, and line*, but lower weights than for rare terms
- We will use document frequency (df) to capture this in the score
- df ($\leq N$) is the number of documents that contain the term

Inverse document frequency (idf) weight

- df_t is the document frequency of *t*: the number of documents that contain *t*
  - df_t is an inverse measure of the informativeness of *t*
  - Inverse document frequency is a direct measure of the informativeness of *t*
- We define the idf (inverse document frequency) of *t* by

$$\mathrm{idf}_t = \log_{10} N/\mathrm{df}_t$$

  - use log to dampen the effect of N/df_t
- Most common variant of idf weight

idf example, suppose N = 1 million

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

$$\mathrm{idf}_t = \log_{10}\left(N/\mathrm{df}_t\right)$$

There is one idf value for each term *t* in the collection

Effect of idf on ranking

- Does idf have an effect on ranking for one-term queries, like
  - iPhone
- idf has no effect on ranking one term queries
  - idf affects the ranking of documents for queries with at least two terms
  - For the query capricious person, idf weighting makes occurrences of capricious count for much more in the final document ranking than occurrences of person

Collection vs. Document frequency

- The collection frequency of *t* is the number of occurrences of *t* in the collection, counting multiple occurrences
- Example: which word is a better search term (and should get a higher weight)?

Word	Collection frequency	Document frequency
insurance	10440	3997
try	10422	8760

- The example suggests that df is better for weighting than cf

tf-idf weighting

- The tt-idf weight of a term is the product of its tf weight and its idf weight.

$$tf\text{-}idf_{t,d} = weight(t,d) \times idf\ \ weight(t)$$

- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

Document example and its vector representation

TRUMP MAKES BID FOR CONTROL OF RESORTS Casino owner and real estate Donald Trump has offered to acquire all Class B common shares of Resorts International Inc, a spokesman for Trump said. The estate of late Resorts chairman James M. Crosby owns 340,783 of the 752,297 Class B shares. Resorts also has about 6,432,000 Class A common shares outstanding. Each Class B share has 100 times the voting power of a Class A share, giving the Class B stock about 93 pct of Resorts' voting power.

[RESORTS:0.624] [CLASS:0.487] [TRUMP:0.367] [VOTING:0.171] [ESTATE:0.166] [POWER:0.134] [CROSBY:0.134] [CASINO:0.119] [DEVELOPER:0.118] [SHARES:0.117] [OWNER:0.102] [DONALD:0.097] [COMMON:0.093] [GIVING:0.081] [OWNS:0.080] [MAKES:0.078] [TIMES:0.075] [SHARE:0.072] [JAMES:0.070] [REAL:0.068] [CONTROL:0.065] [ACQUIRE:0.064] [OFFERED:0.063] [BID:0.063] [LATE:0.062] [OUTSTANDING:0.056] [SPOKESMAN:0.049] [CHAIRMAN:0.049] [INTERNATIONAL:0.041] [STOCK:0.035] [YORK:0.035] [PCT:0.022] [MARCH:0.011]

Bag-of-Words representation (high dimensional sparse vector)

Term / document matrix

- Most common form of representation in text mining is the ***term - document* matrix**
  - Term: typically a single word, but could be a word phrase like “data mining”
  - Document: a generic term meaning a collection of text to be retrieved
  - Can be large - terms are often 50k or larger, documents can be in the billions (www).
  - Can be binary, or use counts

Term document matrix

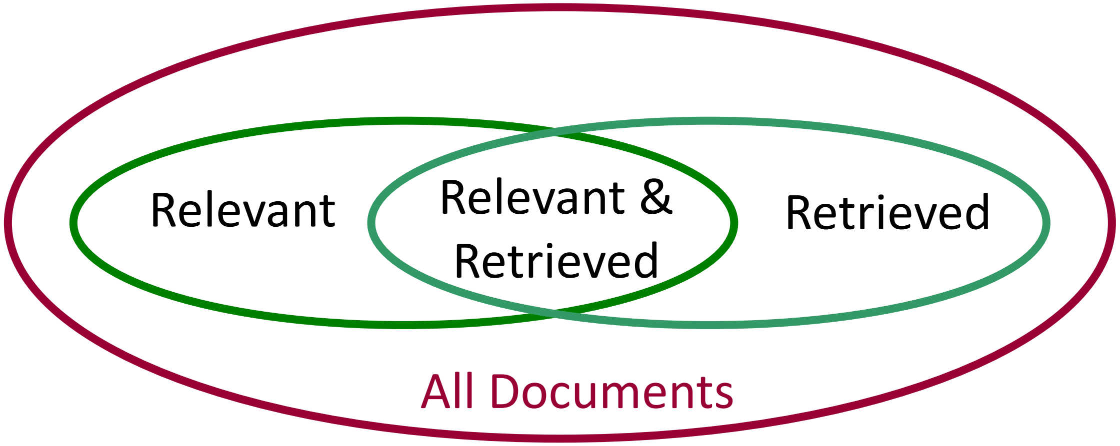
Example: 10 documents: 6 terms

	Database	SQL	Index	Regression	Likelihood	linear
D1	24	21	9	0	0	3
D2	32	10	5	0	3	0
D3	12	16	5	0	0	0
D4	6	7	2	0	0	0
D5	43	31	20	0	3	0
D6	2	0	0	18	7	6
D7	0	0	1	32	12	0
D8	3	0	0	22	4	4
D9	1	0	0	34	27	25
D10	6	0	0	17	4	23

$$D_1 = (d_{i1}, d_{i2}, \ldots, d_{it})$$

- Each document now is just a vector of terms, sometimes boolean

Basic Measures for Text Retrieval



$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses)

$$recall = \frac{|\{\mathrm{Re}levant\} \cap \{\mathrm{Re}trieved\}|}{|\{\mathrm{Re}levant\}|}$$

Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$fallout = \frac{|\{non - \mathrm{Re}levant\} \cap \{\mathrm{Re}trieved\}|}{|\{non - \mathrm{Re}levant\}|}$$

Fallout: measures the proportion of non-relevant documents retrieved from all available non-relevant documents

Information Retrieval Models

- Boolean Model
- Vector Model
- Probabilistic Model

Boolean Model

- Consider that index terms are either present or absent in a document
- As a result, the index term weights are assumed to be all binaries
- A query is composed of index terms linked by three connectives: not, and, and or
  - e.g.: car *and* repair, plane *or* airplane
- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Vector Model

- Deals with a |V|-dimensional vector space
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional
  - hundreds of millions of dimensions when you apply this to a web search engine
- This is a very sparse vector
  - most entries are zero

Queries as vectors

- Key idea 1: Do the same for queries: represent them as vectors in the space
- Key idea 2: Rank documents according to their proximity to the query in this space
  - proximity = similarity of vectors
  - proximity ≈ inverse of distance
- The goal to do this is to get away from the either-in-or-out Boolean model
- Instead: rank more relevant documents higher than less relevant documents

Cosine (query,document)

Dot product

Unit vectors

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{t=1}^{|V|} q_t d_t}{\sqrt{\sum_{t=1}^{|V|} q_t^2}\sqrt{\sum_{t=1}^{|V|} d_t^2}}$$

q_t is the tf-idf weight of term *t* in the query
d_t is the tf-idf weight of term *t* in the document

$\cos(\vec{q},\vec{d})$ is the cosine similarity of $\vec{q}$ and $\vec{d}$ … or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$

The cosine similarity can be seen as a method of normalizing document length during comparison

Vector Model

- Represent the query as a weighted TF-IDF vector
- Represent each document as a weighted TF-IDF vector
- Compute the cosine similarity score for the query vector and each document vector
- Rank documents with respect to the query by score
- Return the top k (e.g., $k = 10$) to the user

Document Summarization

- A summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)
- Summaries may be classified as:
  - Extractive: are created by reusing portions (words, sentences, etc.) of the input text

    Ex: search engines typically generate extractive summaries from webpages

  - Abstractive: information from the source text is re-phrased

Document Classification

Given a set of documents and their classes, e.g.

- Spam, no-spam
- Topic categories in news: current affairs, business, sports, entertainment, ...
- Any other classification

1. Learn which document features characterize the classes = learn a classifier
2. Predict, from document features, the classes
   - For old documents with known classes
   - For new documents with unknown classes

Document Clustering

- Unsupervised process through which documents are classified into groups called *clusters*

- The key operation in the clustering operation is the similarity measure used to compare documents:
  - the most used measure - cosine similarity

- Most used clustering algorithms in document segmentation:
  - K-Means Algorithm
  - Agglomerative Hierarchical Clustering Algorithms ...