Data Mining II / Adv. Topics in Data Science

Information Retrieval

Álvaro Figueira
Rita Ribeiro

1

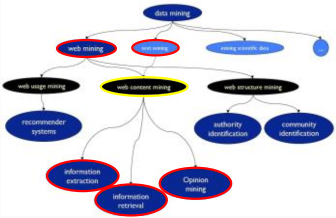
Summary

2

Basic Concepts

3

Data Mining - a structured view



4

Information Retrieval and Big Data

*"Big data is a term that describes large **volumes** of high **velocity**, complex and **variable** data that require advanced techniques and technologies to enable the capture, storage, distribution, management, and analysis of the information." (TechAmerica Foundation's Federal Big Data Commission, 2012)*

- Big data has three dimensions described by *the Three V's* [Gandomi and Haider, 2015]: **Volume, Velocity and Variety**.
- **Variety** → structural heterogeneity in data
  - structured data (tables in relational databases) represent only 5% of the existing data;
  - non-structured data: text, images, audio and videos represent all the remaining data, which do not have the structure required by a computer for the analysis.

5

Information Retrieval: Definitions

- *"Information retrieval **deals with** the representation, storage, and access to documents or representatives of documents (document surrogates)"* - Gerard Salton
- *"Information retrieval (IR) **is the activity of** obtaining information resources relevant to an information need from a collection of information resources"* - Wikipedia
- *"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)"* - Manning & Raghava

6

Information Retrieval

- IR helps users finding information that matches their needs.
- This process involves:
  - acquisition of information
  - organization of information
  - storage of information
  - retrieval of information
  - distribution of information
- **Goal: predict**, given an information source, and apriori-knowledge from the user, **the objects which are the most relevant for the user**.

7

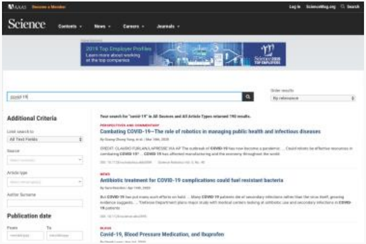
Information Retrieval - architecture model

General Architecture of a Information Retrieval Model

1. user poses a query;
2. the query is sent to the information retrieval system (IRS);
3. which uses the document index;
4. to get documents with query terms;
5. compute relevance of documents
6. rank results

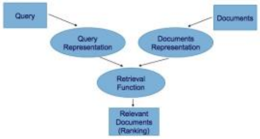
8

Information Retrieval (example)



9

Information Retrieval (schematic process)



- How to represent documents?
- How to represent queries?
- How to compare documents with queries, such that the most relevant documents to the queries are selected?

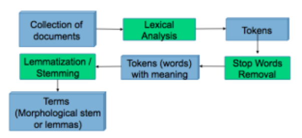
10

Text Preprocessing

11

Text Preprocessing

To get from a documents collection a set of terms, the following text preprocessing steps are usually followed.



12

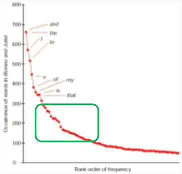
Text Preprocessing: Lexical Analysis

- Process by which a sequence of characters is separated in tokens.
- Tokens are groups of characters with a collective meaning (words).
- Word identification can be performed by means of:
  - blank spaces
  - punctuation marks: usually ignored
  - capitalization: convert to lower case
  - digits: usually ignored, but can be kept depending on the context (e.g. dates, H_2O, CC4061)
  - hyphens: ignored (e.g. on-line → online) or kept (e.g., in the Knuth-Morris-Prat algorithm)

13

Text Preprocessing: discriminant words

- Some words occur very frequently.
- A medium number of words present a medium frequency.
- Lots of words occur rarely.
- Those very **frequent words are not good discriminant** of the content of the documents.
- Words the **occur very rarely** are specific to particular topic, or can be **jargon, proper nouns**, etc.
- Those with **medium frequency are the most discriminant**.



14

Text Preprocessing: Stop words removal

- **Stop words**: are those with no meaning.
- Parts of speech that usually don't have meaning: pronouns, prepositions, conjunctions, determiners, ...
- Very frequent words (typically, represent more than 80% of the total number of words) and are not useful for retrieval.

- If removed, space is saved, and retrieval efficiency gained.
- The remaining is usually stored in a fast data structure (e.g., a Hash table)

15

Text Preprocessing: Stop words Removal (cont.)

- They depend on the language
  - In portuguese:
    - *a, agora, ainda, alguém, algum, alguma, algumas, alguns, antes, ao, aos, após, aquela, aquelas, aquele, aqueles, aquilo, as, até, através, cada, com, como, da, daquele, daqueles, das, de, dela, delas, dele, deles, depois,....*
  - In english:
    - *a, about, above, across, after, again, against, all, an, and, any, anybody, anything, are, as, at, be, because, been, before, being, below, between, both, but, by, ...*
- *"To be or not to be"*
  - A short list of stop words should be used for general collections or for unexperienced users.
  - Longer stop words lists for very specific domains.

16

Text Preprocessing: Stemming

- Process by which the *lexical stem* is extracted from a word.
- **Stem**: part of the word resulting when removing affixes (suffixes, infixes, prefixes)
- The same stem will represent a family of words semantic and morphologically related
  - *{connected, connecting connection, connections, disconnected} → connect*
  - *{computer, computational, computation} → comput*
  - *{compressed, compression} → compress*
- The morphologic analysis to extract the stem depends on the language and is usually complex

17

Text Preprocessing: Stemming (cont.)

- Affix removal algorithms:
  - based on heuristic methods;
  - successively, applying rules to the words;
  - different words could generate the same stem;
  - different methods: Lovins, Slaton, Dawson, Porter.
- "Porter Stemmer"
  - mainly **suffix stripping**;
  - example: plural removal using the rule with longest
  -

suffix rule	example
sses → ss	caresses → caress
ies → i	ponies → poni
ss → ss	caress → caress
s →	cats → cat

18

Text Preprocessing: Lemmatization

- Transformation of the word (inflectional and derivationally related forms) to the corresponding **lemma** ("the common base")
- Example of rules:
  - Convert any verbal form to infinitive:
    {am, are, is} → *be*
    {going, gone, went} → *go*
  - Plural to singular: *cars* → *car*
  - Female gender to male gender: *menina* → *menino*
- The strip will always **result in an actual word**.
- More precise than stemming but needs more resources (usually dictionaries) and more disciplines as:
  - Natural Language Processing and
  - Computational Linguistics.

19

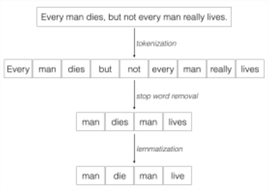
Text Preprocessing: Lemmatization and Stemming

- Lemmatization vs Stemming:
  - developing a stemmer is far simpler than building a lemmatizer;
  - for lemmatizer deep linguistics knowledge is required; but, the noise will be reduced, and the information retrieval process will be more accurate.
- Both methods are used to reduce the size of the vocabulary.
- Not all the words are indexed but only a kind of representatives.
- Advantages:
  - reduction in the vocabulary → efficiency and space saving;
  - the number of matching documents to retrieve increases.
- Disadvantages:
  - information about the complete word is lost.

20

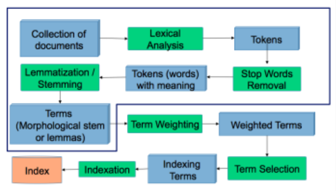
Text Preprocessing

Tokenization → Stop words removal → Lemmatization



21

Convered so far...

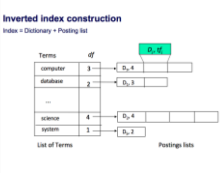


Remember: this can be done on a set of documents

22

Indexation of terms

Objective: To build an **index** from a set of documents



- efficient data structure (e.g. Hash Table, B-Tree, Trie)
- fast access to the collection to get the relevant documents for a query
- terms are stored, pointing to the documents where they occur, and their corresponding weight.

23

Retrieval Models

24

Information Retrieval Models

- Each document is seen as the set of its terms.
- A **term** may represent a single word or multiword units (e.g., “White House”).
- **Weights can be associated to terms**.
- Different models find **weights** in different ways
  - Boolean Model
  - Vector Space Model
    - Binary Scheme
    - TF Scheme
    - TF-IDF Scheme
  - Statistical Language Model (uses Bayesian reasoning)

We will study these

21

25

Information Retrieval Models: Boolean Model

- The simplest retrieval model.
- Each document is represented by the presence/absence of terms.
- A query is a boolean expression with ***and***, ***or*** and ***not*** operators to join terms.
- It looks for exact match, i.e. every document that matches the query.
- **Result:** set of documents, without ordering, satisfying the query.

document	t_1	t_2	t_3
d_1	1	0	1
d_2	1	0	0
d_3	0	1	1
d_4	1	1	1

query:
$t_1 \wedge (t_2 \vee \neg t_3)$

result: $\{ d_2, d_4 \}$

26

Information Retrieval Models: Boolean Model (cont.)

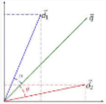
- Advantages:
  - very efficient;
  - predictable, easy to explain;
  - it works well with experienced users;
  - it is still used in easy searches (e.g., email).
- Disadvantages:
  - returned documents are not represented by relevance;
  - "close" documents are not presented as such;
  - it is difficult to formulate complex boolean queries for normal users;
  - they can return too many documents, or very few.

27

Information Retrieval Models: Vector Space Model

- **Terms** are **axes** of the space.
- Documents and queries are **vectors** in this space.
- Vector coordinates are term **weights**.

- Very high-dimensionality but very sparse vectors (most entries are zero)
- Ranks the documents by similarity degree (e.g., cosine)
- Allows to retrieve documents that partially match the query.



28

Information Retrieval Models: Vector Space Model (Binary)

Binary Scheme

- The weights indicate the presence or the absence of a term in a document.

$$W_{bi}(t_j, d_i) = \begin{cases} 1 & if\ t_j \in d_i \\ 0 & otherwise \end{cases}$$

29

Information Retrieval Models: Term-Frequency

Term-Frequency (TF) Scheme

- **Motivation:** A term appearing often in a document may be more important for identification than a term appearing rarely.
- This aspect is captured by *tf* , the "term frequency" in a document.

$$w_{TF}(t_j, d_i) = tf(t_j, d_i)$$

where $tf(t_j, d_i)$ is the **frequency** of term j in document i

- It can be normalized:

$$w_{TF}(t_j, d_i) = \frac{tf(t_j, d_i)}{\sum_{t_k \in d_i} tf(t_k, d_i)}$$

30

Information Retrieval Models: Term-Frequency (cont.)

Term-Frequency (TF) Scheme

Example:

document	t_1	t_2	t_3	$w_{tf}(t_1, d_i)$	$w_{tf}(t_2, d_i)$	$w_{tf}(t_3, d_i)$
d_1	1	0	1	1/2	0	1/2
d_2	1	0	0	1	0	0
d_3	1	1	0	1/2	1/2	0
d_4	2	1	1	1/2	1/4	1/4
	5	2	2			

31

Information Retrieval Models: Inverse Document Frequency

Inverse Document Frequency (IDF)

- **Motivation:** If a term appears many times, in many documents, it will probably be irrelevant. However, if a term occurs in just a few documents it will have a greater discriminating power.
- This aspect is captured by *idf*, the "inverse document frequency"

$$idf(t) = log\left(\frac{N}{|\{d \in D: t \in d\}|}\right)$$

Where, ***N*** is the number of documents in the corpus (D).
Therefore, we are dividing it by the number of documents in which term ***t*** occurred.

32

Information Retrieval Models: TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) Scheme

- The weights take into account both intra-document and inter-document term frequency.

$$w_{TF\text{-}IDF}(t_j, d_i) = tf(t_j, d_i) \times idf(t_j) =$$

$$= \frac{tf(t_j, d_i)}{\sum_{t_k \in d_i} tf(t_k, d_i)} \times log\left(\frac{N}{|\{d \in D: t_j \in d\}|}\right)$$

Example:

document	t_1	t_2	t_3	TF_{t1}	TF_{t2}	TF_{t3}	$W_{tf-idf}(t_1, d_i)$	$W_{tf-idf}(t_2, d_i)$	$W_{tf-idf}(t_3, d_i)$
d_1	1	0	1	1/2	0/2	1/2	0	0	1/2*log(2)
d_2	1	0	0	1	0/1	0/1	0	0	0
d_3	1	1	0	1/2	1/2	0/3	0	1/2*log(2)	0
d_4	1	1	1	1/3	1/3	1/3	0	1/3*log(2)	1/3*log(2)
$df(t_j)$	4	2	2						
$N/df(t_j)$	1	2	2						
$log(N/df(t_j))$	0	log(2)	log(2)						

33

Information Retrieval Models: Document Similarity

Similarity for retrieval

- A *query* is represented in the same manner as other documents.
- Relevant documents are the ones closer to the query, using some similarity metric, e.g., cosine similarity:

$$\cos(d_i, q) = \frac{\sum_{t_k \in T} w(t_k, d_i) \times w(t_k, q)}{\sqrt{\sum_{t_k \in T} w(t_k, d_i)^2} \times \sqrt{\sum_{t_k \in T} w(t_k, q)^2}}$$

Example:

document	t_1	t_2	t_3
d_1	1	0	1
d_2	1	0	0
d_3	1	1	0
d_4	1	1	1
q	0	1	1

$cos(d_1, q) = 0.5000000$

$cos(d_2, q) = 0.0000000$

$cos(d_3, q) = 0.5000000$

$cos(d_4, q) = 0.8164966$

34

Retrieval Evaluation

35

Retrieval Evaluation

The **quality** of an Information Retrieval System depends on:

- space for indexing the documents
- time, i.e., efficiency in retrieving and indexing
- user satisfaction in terms of usability or operation
- retrieval effectiveness

36

Evaluation Measures

- How to evaluate the retrieval effectiveness?

$$precision = \frac{\text{\# relevant docs retrieved}}{\text{\# docs retrieved}}$$

$$recall = \frac{\text{\# relevant docs retrieved}}{\text{\# relevant docs}}$$

- **Objective:** get as much relevant documents as possible while at the same time, getting as few irrelevant documents as possible

37

Evaluation Measures (summary)

Summary of the most used measures

- prec@k: precision at top k retrieved documents
- rec@k: recall at top k retrieved documents
- MAP (Mean Average Precision): average precision each time a relevant document is retrieved
- F1: harmonic mean between precision and recall: $F1 = 2 \times \frac{P \times R}{P + R}$
- precision-recall curves
- NDCG (Normalized Discounted Cumulative Gain): gain, or graded relevance/usefulness of a document, is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks

38

The difference between micro, macro, and weighted averages

- In multi-class classification problems, models often compute a metric for each class.
- Example: in a 3-class problem, three precision scores are returned.
- However, we just need a single global metric – the averaging methods

1.Macro average
A simple arithmetic mean. Example: if precision scores are 0.7, 0.8, 0.9, macro average would be their mean = 0.8.

2.Weighted average
This method takes ınto account the class imbalance as metrics for each class are multiplied by the proportion of that class. Example: if there are **100 samples** (**30**, **45**, **25** for each class respectively) and the precision scores are .7, .8, .9, the weighted average would be:

0.3 * 0.7 + 0.45 * 0.8 + 0.25 * 0.9 = 0.795

3.Micro average
Micro average is the same as accuracy — it is calculated by dividing the number of all correctly classified samples (True Positives) by the total number of correctly and incorrectly classified (True Positives + False Positives) samples of each class.

Obs: Use micro average when there is an imbalanced problem. This approach does not take into account class distribution/contributions. Which means it is sensitive to performance on rare classes.

39

Evaluation Measures (cont.)

Example: the system ranked the documents like this…

Rank	Relevant
1	+
2	+
3	-
4	+
5	-
6	-
7	+
8	-
9	-

(#Relevants = 4)

$prec@2 = 1 \quad prec@4 = 3/4 \quad prec@8 = 1/2$

$rec@2 = 1/2 \quad rec@4 = 3/4 \quad rec@8 = 1$

$MAP = (1/1 + 2/2 + 3/4 + 4/7)/4 = 0.8303571$

$F1@2 = 2 * (1 * 1/2)/(1 + 1/2) = 0.6666667$

40