Information retrieval I

Introduction, efficient indexing, querying

Clovis Galiez

Mastère Big Data

December 3, 2019

Objectives of the course

- Acquire a culture in information retrieval
- Master the basics concepts allowing to understand:
  - what is at stake in novel IR methods
  - what are the technical limits

This will allow you to have the basics tools to analyze current limitations or lacks, and imagine novel solutions.

Proceedings of the lecture

- No lecture handbook, only slides and materials of the practicals session.
- So... take notes and ask questions!
- Evaluation: exam and optional project (bonus)

Outline of the lectures

- Indexing, basic querying
- Vector-space model, latent semantics, ranking
- Modern IR techniques: artificial intelligence, embeddings.
  Hands-on session: programming a search engine (Python)
- Hands-on Part-II.

What is information retrieval (IR)?

What is information retrieval (IR)?

Definition

Answering a query by extracting **relevant information** from **a collection of documents**.

What is information retrieval (IR)?

Definition

Answering a query by extracting **relevant information** from **a collection of documents**.

Typical example

Google.

Some open-source tools for in-house IR

IR tools:

- Lucene
- elastic

NLP tools:

- NLTK (Python)
- spaCy

(far to be exhaustive!)

What is "document" and "information"?

Information retrieval

Answering a query by extracting **relevant information** from **a collection of documents**.

What is "document" and "information"?

Information retrieval

Answering a query by extracting **relevant information** from **a collection of documents**.

What is **relevant information** in the context of information retrieval?

What is "document" and "information"?

Information retrieval

Answering a query by extracting **relevant information** from **a collection of documents**.

What is **relevant information** in the context of information retrieval?

Information

Subset of documents relevant to a query.

Here, **documents** are web pages, images, pdf, etc.

What is "document" and "information"?

Information retrieval

Answering a query by extracting **relevant information** from **a collection of documents**.

What is **relevant information** in the context of information retrieval?

Information

Subset of documents relevant to a query.

Here, **documents** are web pages, images, pdf, etc.
What is a good answer quality?

Correctness, relevance and truth

When was the last US presidential elections?

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or...

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or**...**

- 1st Sept. 2018

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or...

- 1st Sept. 2018
- Nov. 2016

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or**...**

- 1st Sept. 2018
- Nov. 2016

Relevance

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or**...**

- 1st Sept. 2018
- Nov. 2016

Relevance

- Same time as the previous ones, but 5 years later

Correctness, relevance and truth

When was the last US presidential elections?

correct or **incorrect**

- Blue
- 42:17

true, **false** or**...**

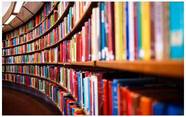
- 1st Sept. 2018
- Nov. 2016

Relevance

- Same time as the previous ones, but 5 years later
- during the 21st Century
- 1478563200s since Unix Epoch[a]

[a] Number of seconds elapsed since 1st of January 1970

Querying and ranking systems



Collection of documents

↓

Query ⟶ $\boxed{\text{Querying system}}$

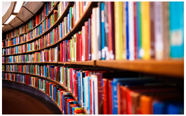
↓

Correct answers

↓

Ranking of answers
(relevance)

Querying systems deal with correctness



Collection of documents

↓

Query ⟶ Querying system

↓

Correct answers

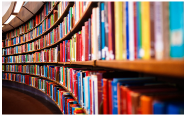
↓

Ranking of answers (relevance)

Filter documents that *correctly* answers a given query

- Boolean queries
  - Checks if a word is present or not in a document
- Vector-based models
- Probabilistic models
  - Naïve Bayes model

Ranking systems deal with relevance



Collection of documents

↓

Query ⟶ $\boxed{\text{Querying system}}$

↓

Correct answers

↓

Ranking of answers
(relevance)

Ranking methods:

- structure-agnostic algorithms
- unsupervised ranking
  - PageRank
- supervised ranking
  - machine learning

What are the pitfalls?

Exercise

Take a few minutes to list what could be the different pitfalls for querying and ranking systems.

What are the pitfalls?

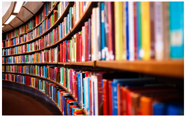
Exercise

Take a few minutes to list what could be the different pitfalls for querying and ranking systems.

- Complexity of natural language
- Ambiguity of natural language
- Size of the data
- ...

IR specific to the World Wide Web

IR and the web



Collection of documents

↓

Query ⟶ Querying system

↓

Correct answers

↓

Ranking of answers
(relevance)

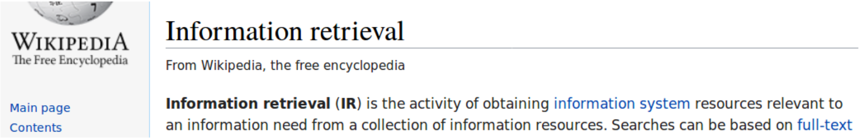
2 specificities:

- Building the collection of documents
  - **Crawling** the web
  - **Indexing** documents
- Ranking the documents (next lecture)

Gathering data on the web (crawling)

The web structure: languages

HTML (*HyperText Markup Language*) is the main language for describing a web page. From `https://en.wikipedia.org/wiki/Information_retrieval`:



HTML source code behind:

```
... obtaining <a href="/wiki/Information_system" title="Information system">
information system</a>  resources relevant to an ...
```

The web structure: languages

HTML (*HyperText Markup Language*) is the main language for describing a web page. From `https://en.wikipedia.org/wiki/Information_retrieval`:

HTML source code behind:

```
... obtaining <a href="/wiki/Information_system" title="Information system">
information system</a>  resources relevant to an ...
```

How would you collect information from the web?

The web structure: crawling

Hopping from link to link, one can collect/process data on the web:

Web crawling
JumpStation (1993)

Must keep track of already visited pages (e.g. trie, hash table).

Parsing links from a web page

Use parsing tools, like context-free grammars for XML for instance.
Note that for extracting links, regular expersssions are enough!

Exercise

Find a regex that extracts the URL of an HTML link.

```html
<a href="http://www.wikipedia.org">The linked text</a>
```

Extend your regex to extract both the URL and the linked text.

Quality of the data: HTML errors, difficult parsing.

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	
Data size crawled by Google	
Number of pages with $> k/2$ incoming links	
Number of pages of length $> L/2$	

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	
Number of pages with $> k/2$ incoming links	
Number of pages of length $> L/2$	

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	$> 10^8$Gb
Number of pages with $> k/2$ incoming links	
Number of pages of length $> L/2$	

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	$> 10^8$Gb
Number of pages with $> k/2$ incoming links	$\sim 2^{\gamma}.N_k$ (Zipf's law)
Number of pages of length $> L/2$	

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	$> 10^8$Gb
Number of pages with $> k/2$ incoming links	$\sim 2^\gamma.N_k$ (Zipf's law)
Number of pages of length $> L/2$	$\sim 2^\gamma.N_L$ (Zipf's law)

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	$> 10^8$Gb
Number of pages with $> k/2$ incoming links	$\sim 2^\gamma . N_k$ (Zipf's law)
Number of pages of length $> L/2$	$\sim 2^\gamma . N_L$ (Zipf's law)

6 degrees of separation law.

[1] [Pandya et al. IJIRST 2017]

What is the size of the crawled data?

From technical solution to practice... Quizz:

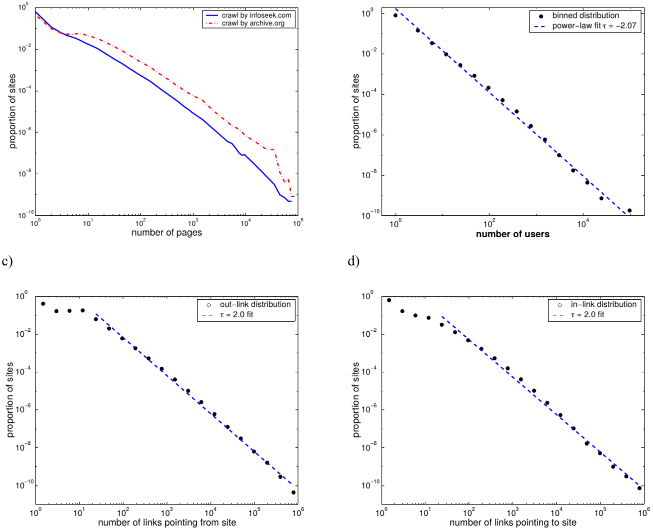
Number of pages indexed by Google	$\sim 10^{11}$
Data size crawled by Google	$> 10^8$Gb
Number of pages with $> k/2$ incoming links	$\sim 2^\gamma.N_k$ (Zipf's law)
Number of pages of length $> L/2$	$\sim 2^\gamma.N_L$ (Zipf's law)

6 degrees of separation law.

Between 0.2% and 4% of the web is accessible by crawling[1].
What is "uncrawlable" is coined the **deep web**.

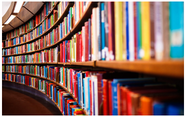
[1] [Pandya et al. IJIRST 2017]

Experimental evidence of the Zipf's law



[Adamic et al. *Glottometrics*, 2002]

From gathering to representation



Collection of documents

↓

Query ⟶ Querying system

↓

Correct answers

↓

Ranking of answers (relevance)

Representations of a web document

How to query for correct documents?

Exercise

Take a few minutes to think how you would retrieve the web documents corresponding to the query:

`result last elections president united states`

How to query for correct documents?

Exercise

Take a few minutes to think how you would retrieve the web documents corresponding to the query:

```
result last elections president united states
```

You may have encounter the following issues:

- how to correctly match words in the document (tokenization)
- how to match equivalent word (e.g. plural)
- how to implement it

Tokenization

Process of chopping the text of a document in atomic elements:

Brian is in the kitchen $\rightarrow$ <u>Brian</u> <u>is</u> <u>in</u> <u>the</u> <u>kitchen</u>

Tokenization

Process of chopping the text of a document in atomic elements:

Brian is in the kitchen	$\rightarrow$	<u>Brian</u> <u>is</u> <u>in</u> <u>the</u> <u>kitchen</u>
United States president	$\rightarrow$	

Tokenization

Process of chopping the text of a document in atomic elements:

Brian is in the kitchen	$\rightarrow$	<u>Brian</u> <u>is</u> <u>in</u> <u>the</u> <u>kitchen</u>
United States president	$\rightarrow$	<u>United States</u> <u>president</u>

Usually, tokenizers remove the punctuation.

Tokenization

Process of chopping the text of a document in atomic elements:

Brian is in the kitchen	$\rightarrow$	Brian is in the kitchen
United States president	$\rightarrow$	United States president

Usually, tokenizers remove the punctuation.

May be difficult: `United States` $\neq$ `United + States`!

- Data mining approaches help to extract the right tokens: if two words are significantly seen one after the other, may be consider as a token.
- Some languages are agglutinative (e.g. Turkish).

Python library for NLP: nltk

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
```

https://www.nltk.org/

Stemming

Language-specific rules defining equivalent words up to a usual transformation (e.g. -ing, -ed, -s, etc.).
For instance, we would transform:

plural forms:	elections	$\rightarrow$	election
substantive/adjectival forms:	presidential	$\rightarrow$	president
stop-words removal:	in	$\rightarrow$	`NULL`

Stemming

Language-specific rules defining equivalent words up to a usual transformation (e.g. -ing, -ed, -s, etc.).
For instance, we would transform:

plural forms:	elections	$\rightarrow$	election
substantive/adjectival forms:	presidential	$\rightarrow$	president
stop-words removal:	in	$\rightarrow$	`NULL`

Again, same difficulties could appear:

- ambiguity: police, policy $\rightarrow$ polic
- Non-conflating: mother, maternal

Implementation of stemming in Python

```
>>> from nltk.stem.porter import *
>>> stemmer = SnowballStemmer("english")
>>> print(stemmer.stem("running"))
run
```

https://www.nltk.org/

After tokenizing and stemming: querying

Query: `result last elections president united states`

Stemmed tokens:

`result, last, election, president, united states`

After tokenizing and stemming: querying

Query: `result last elections president united states`

Stemmed tokens:

`result, last, election, president, united states`

How to query your documents?

Naive representation: vector of tokens

Matrix with the occurrence of tokens in documents.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...

Naive representation: vector of tokens

Matrix with the occurrence of tokens in documents.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...
Query	1	1	0	0	1	...

Naive representation: vector of tokens

Matrix with the occurrence of tokens in documents.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...
Query	1	1	0	0	1	...

Naive representation: vector of tokens

Matrix with the occurrence of tokens in documents.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...
Query	1	1	0	0	1	...

Exercise

Can you foresee any practical problem?

Naive representation: vector of tokens

Matrix with the occurrence of tokens in documents.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...
Query	1	1	0	0	1	...

Exercise

Can you foresee any practical problem? What is the size of the matrix? Can it fit in **memory**?

Sparse representation

Since documents contain only a small fraction of existing tokens, most of the vector of token entries are null.
We can use a sparse encoding of the same information:

tok 1	tok 2	tok 3	tok 4	tok 5
election	president	crazy	united	United States

doc 1→tok 1, tok 2, tok 5
doc 2→tok 2, tok 3, tok 5
doc 3→tok 1, tok 2, tok 3 ,tok 5

Exercise

What is the size of the sparse encoding data structure?

Sparse representation

Since documents contain only a small fraction of existing tokens, most of the vector of token entries are null.
We can use a sparse encoding of the same information:

tok 1	tok 2	tok 3	tok 4	tok 5
election	president	crazy	united	United States

doc 1→tok 1, tok 2, tok 5
doc 2→tok 2, tok 3, tok 5
doc 3→tok 1, tok 2, tok 3 ,tok 5

Exercise

What is the size of the sparse encoding data structure?

Hmmm

What is the complexity of a naive algorithm extracting the matching documents from a sparse encoding doc-tok?

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm?

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$
- Memory complexity of the sparse indexing algorithm?

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$
- Memory complexity of the sparse indexing algorithm? $\mathcal{O}(N.k)$

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$
- Memory complexity of the sparse indexing algorithm? $\mathcal{O}(N.k)$
- Time complexity of your search algorithm in a sparse index?

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$
- Memory complexity of the sparse indexing algorithm? $\mathcal{O}(N.k)$
- Time complexity of your search algorithm in a sparse index? $\mathcal{O}(N.k)$

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Some complexities

Let N be the number of indexed documents, T the total number of tokens, and k the average number of token per document[2].

- Memory complexity of the naive indexing algorithm? $\mathcal{O}(N.T)$
- Memory complexity of the sparse indexing algorithm? $\mathcal{O}(N.k)$
- Time complexity of your search algorithm in a sparse index? $\mathcal{O}(N.k)$

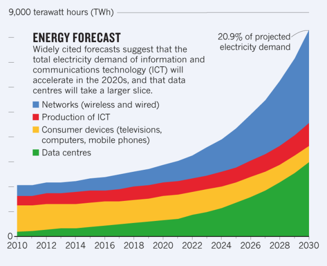
Can we do better?

[2] k can be related to the length of the document through Heap's law: $k = K.L^{\beta}$. In practice, $K = 50, \beta = 0.5$

Complexity matters

Action	Complexity
Sorting	$\mathcal{O}(N \log N)$
Searching a sorted list	$\mathcal{O}(\log N)$
Accessing an element of a matrix	$\mathcal{O}(1)$

Complexity matters!



The chart above is an 'expected case' projection from Anders Andrae, a specialist in sustainable ICT. In his 'best case' scenario, ICT grows to only 8% of total electricity demand by 2030, rather than to 21%.

Electric energy consumption in France per year per person: 0.067GWh
Electric energy of Google per year: 2.500GWh[3].
Equivalent consumption of 40.000 people.

Exercise

Imagine a way of improving how to query a big set of documents.

[3]according to Google. Other sources say $4\times$ more

Inverse sparse index

Idea

Inverting the sparse representation and sorting by document allows to reduce the complexity.

tok 1	tok 2	tok 3	tok 4	tok 5
election	president	crazy	united	United States

doc 1→tok 1,tok 2,tok 5
doc 2→tok 2,tok 3,tok 5
doc 3→tok 1,tok 2,tok 3,tok 5

Inverse sparse index

Idea

Inverting the sparse representation and sorting by document allows to reduce the complexity.

tok 1	tok 2	tok 3	tok 4	tok 5
election	president	crazy	united	United States

doc 1→tok 1,tok 2,tok 5
doc 2→tok 2,tok 3,tok 5
doc 3→tok 1,tok 2,tok 3,tok 5

tok 1→doc 1,doc 3,...
tok 2→doc 1,doc 2,doc 3,...
tok 3→doc 2,doc 3,...
tok 4→doc 102,...
tok 5→doc 1,doc 2,doc 3,...

Inverse sparse index

Idea

Inverting the sparse representation and sorting by document allows to reduce the complexity.

tok 1	tok 2	tok 3	tok 4	tok 5
election	president	crazy	united	United States

doc 1→tok 1,tok 2,tok 5
doc 2→tok 2,tok 3,tok 5
doc 3→tok 1,tok 2,tok 3,tok 5

tok 1→doc 1,doc 3,...
tok 2→doc 1,doc 2,doc 3,...
tok 3→doc 2,doc 3,...
tok 4→doc 102,...
tok 5→doc 1,doc 2,doc 3,...

Exercise

Compute the time complexity for querying an inverse sparse index.

Building an inverted index in practice

The full sparse index does not fit in memory!

Need to use external memory... what is the main limitation ?

[4] https://nlp.stanford.edu/IR-book/html/htmledition/blocked-sort-based-indexing-1.html

Building an inverted index in practice

⚠ The full sparse index does not fit in memory!

Need to use external memory... what is the main limitation ?

Cannot use disk as memory because of **random accesses** when inserting a new document in the tok-doc hash table.

[4] https://nlp.stanford.edu/IR-book/html/htmledition/blocked-sort-based-indexing-1.html

Building an inverted index in practice

The full sparse index does not fit in memory!

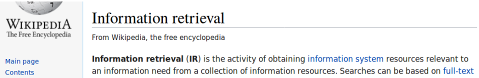
Need to use external memory... what is the main limitation ?

Cannot use disk as memory because of **random accesses** when inserting a new document in the tok-doc hash table.

Block Sort-Based Indexing is a simple algorithm allows to invert big dictionaries that do not fit in main memory, at with linear disk access, and that can even be parallelized[4].

[4] `https://nlp.stanford.edu/IR-book/html/htmledition/blocked-sort-based-indexing-1.html`

Summary

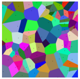

WIKIPEDIA
The Free Encyclopedia

Main page
Contents

Information retrieval

From Wikipedia, the free encyclopedia

Information retrieval (**IR**) is the activity of obtaining information system resources relevant to an information need from a collection of information resources. Searches can be based on full-text



Patent analysis

Read the patent on `https://clovisg.github.io/teaching/mastereBigData/ir/ctd1/USPatentExtract.pdf`, and answer the following questions;

- What problem is at stake?
- What is the strategy to tackle the problem?
- At the end of the patent, guess the application the author is talking about.

Boolean queries are not flexible

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

Boolean queries are not flexible

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

With a good stemming and tokenization, we will match `result` and `election`... we miss the match between `United States` and `White House` :-/

Any solution?

Boolean queries are not flexible

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

With a good stemming and tokenization, we will match `result` and `election`... we miss the match between `United States` and `White House` :-/

Any solution?

- Use semantics (ontologies)
- Use query expansion (add related terms to the query)
- Use a more flexible querying system

Boolean queries do not rank

Example

Query: `result elections United States`

matching results: 718,698,789

How to pick up most relevant results first?

- Next week: With richer querying and representation of the information
- Next week: By exploiting the graph structure of the web (Google)

Some open-source tools for in-house IR

IR tools:

- Lucene
- elastic

NLP tools:

- NLTK (Python)
- spaCy

The vector space model and the latent semantics

Representing documents as vectors in $\mathbb{R}^T$

From binary presence/absence...

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	1	1	0	0	1	...
doc 2	0	1	1	0	1	...
doc 3	1	1	1	0	1	...
...	...	...	...	...	...	...

Representing documents as vectors in $\mathbb{R}^T$

...to real vector space.

	tok 1	tok 2	tok 3	tok 4	tok 5	...
	election	president	crazy	united	United States	...
doc 1	0.01	0.02	0	0	0.006	...
doc 2	0	0.013	0.001	0	0.001	...
doc 3	0.0031	0.008	0.0043	0	0.0021	...
...	...	...	...	...	...	...

What numbers can be useful here ?

Not every term is informative

How do you quantify information according to Shannon theory?

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(E) = f(P(E))$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(E) = f(P(E))$
- it should be contravariant with the probability:

$$P(e_1) < P(e_2) \Rightarrow I(E_1) > I(E_2)$$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(E) = f(P(E))$
- it should be contravariant with the probability:
$$P(e_1) < P(e_2) \Rightarrow I(E_1) > I(E_2)$$
- when E_1 and E_2 are independent, we would like that:
$$I(E_1 \& E_2) = I(E_1) + I(E_2)$$

Not every term is informative

How do you quantify information according to Shannon theory?

Example: which book are you talking about?

Piece of information	Probability	Information content
"the" is frequent	~ 1	Low
"Zarathustra" is frequent	~ 0	High

- information of an event depends on its probability: $I(E) = f(P(E))$
- it should be contravariant with the probability:

$$P(e_1) < P(e_2) \Rightarrow I(E_1) > I(E_2)$$

- when E_1 and E_2 are independent, we would like that:

$$I(E_1 \& E_2) = I(E_1) + I(E_2)$$

If we moreover ask for f to be continuous and non-zero, there is only one possible class of functions: $-log_b$

Information

The information of an event e is defined as $I(E) = -log(P(E))$

Definition

We can now compute the information of a token as:

$$I(t) = -\log(\frac{\#\text{doc including token } t}{\#\text{docs}})$$

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

$||\vec{D}||_1$ carries the total information carried by a document:

Vector representation of a document

A document can be represented by a vector of the fraction information associated to each of its token:

$$D_t = \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t)$$

What does $||\vec{D}||_1$ represent?

$||\vec{D}||_1$ carries the total information carried by a document:

- low if the document contains only common tokens
- average if the document contains few exceptional tokens
- high if the document contains only exceptional items

The tf-idf matrix

Definition

The matrix M which rows – corresponding to each document – are:

$$D_t = \frac{\#\text{ t in D}}{\#\text{ tokens in D}} \times I(t)$$

is called the **tf-idf** (term frequency-inverse document frequency) representation.

The tf-idf matrix

Definition

The matrix M which rows – corresponding to each document – are:

$$D_t = \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t)$$

is called the **tf-idf** (term frequency-inverse document frequency) representation.

Question

What is the unit of elements of the tf-idf matrix?

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query?

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query? Naïve approach: dot product.
Indeed, it makes sense: For each document, compute:

$$\vec{D} \cdot \vec{Q} = \sum_t D_t.Q_t$$

The higher the dot product, the more informative tokens $\vec{Q}$ and $\vec{D}$ share... and the more relevant should be the D with respect to the query Q.

Querying a set of vector

Represent the query the same way:

$$Q_t = \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t)$$

How to retrieve documents related to the query? Naïve approach: dot product.
Indeed, it makes sense: For each document, compute:

$$\vec{D} \cdot \vec{Q} = \sum_t D_t.Q_t$$

The higher the dot product, the more informative tokens $\vec{Q}$ and $\vec{D}$ share... and the more relevant should be the D with respect to the query Q.

For querying purposes, one can select documents such that $\vec{D} \cdot \vec{Q} > \tau$, but it can directly be used for ranking documents.

Correcting for cheaters

Problem

Imagine a way of cheating with this approach.

Correcting for cheaters

Problem

Imagine a way of cheating with this approach.

Content farms

$$\begin{aligned}\vec{D}\cdot\vec{Q} &= \sum_t D_t.Q_t \\ &= \sum_t \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t).\frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t) \\ &\propto \frac{1}{\text{\# tokens in D}} \sum_t \text{\#t in D} \times \text{\#t in Q} \times I(t)^2\end{aligned}$$

Correcting for cheaters

Problem

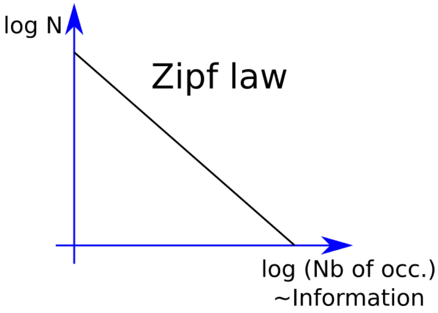
Imagine a way of cheating with this approach.

Content farms

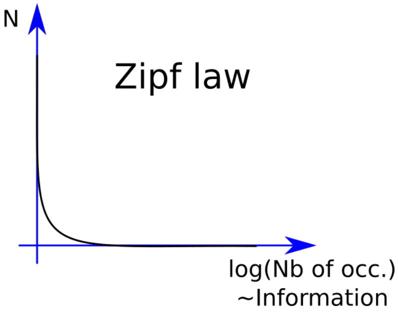
$$
\begin{aligned}
\vec{D} \cdot \vec{Q} &= \sum_t D_t . Q_t \\
&= \sum_t \frac{\text{\# t in D}}{\text{\# tokens in D}} \times I(t) . \frac{\text{\# t in Q}}{\text{\# tokens in Q}} \times I(t) \\
&\propto \frac{1}{\text{\# tokens in D}} \sum_t \text{\#t in D} \times \text{\#t in Q} \times I(t)^2
\end{aligned}
$$

Documents containing many informative words will be selected and ranked first.

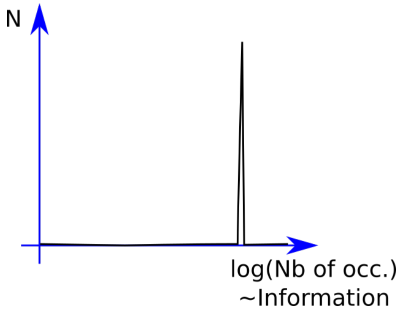
Content farms: pull informative words together



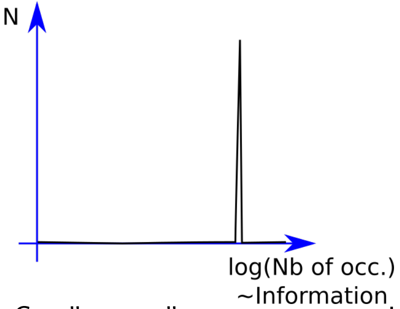
Content farms: pull informative words together



Content farms: pull informative words together



Content farms: pull informative words together



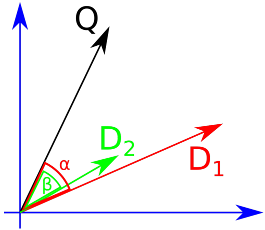
...Can "answer" many more queries

The cosine similarity

How could you correct for content farms cheats?

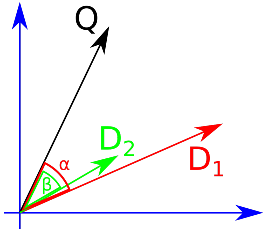
The cosine similarity

How could you correct for content farms cheats?



The cosine similarity

How could you correct for content farms cheats?



Correct by normalizing the similarity:

Consine similarity

$$\text{cosim}(\vec{D}, \vec{Q}) = \frac{\vec{D} \cdot \vec{Q}}{||\vec{D}||_2 . ||\vec{Q}||_2}$$

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

As already pointed out, we could use a semantic approach (ontologies), but need a fixed and manually curated work.

A flexible querying system?

With the vector space model, information of the tokens are now automatically taken into account.
Does it solve the synonymous problem?

Example

Query: `result elections United States`
Doc title: "White House election: live results!"

As already pointed out, we could use a semantic approach (ontologies), but need a fixed and manually curated work.
Can we work directly from the data?

Latent semantics

Special structure of the data: correlations

In practice a tf matrix look like:

Interlude

Video

Special structure of the data: correlations

In practice a tf matrix look like:

Interlude

Video

We observe...

A block structure.

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

If D is a binary vector over documents, what does $M^\top D$ represent?

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.
If Q a binary vector over tokens, what does $M^\top MQ$ represent?

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.
If Q a binary vector over tokens, what does $M^\top MQ$ represent?
The cumulated frequencies of tokens in the (virtual) corpus matching Q.

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?
The fraction of occurrences of tokens of Q in each document.
If D is a binary vector over documents, what does $M^\top D$ represent?
The cumulated frequencies of each token in the corpus D.
If Q a binary vector over tokens, what does $M^\top MQ$ represent?
The cumulated frequencies of tokens in the (virtual) corpus matching Q.

What does it mean that $M^\top MQ = \lambda.Q$?

Reminders from linear algebra

Exercise

If M is a tf matrix and Q a binary vector over tokens, what does MQ represent?

The fraction of occurrences of tokens of Q in each document.

If D is a binary vector over documents, what does $M^\top D$ represent?

The cumulated frequencies of each token in the corpus D.

If Q a binary vector over tokens, what does $M^\top MQ$ represent?

The cumulated frequencies of tokens in the (virtual) corpus matching Q.

What does it mean that $M^\top MQ = \lambda.Q$? What if λ is small? big?

Algebra theorem

Theorem

$M^\top M$ is symmetric and its eigenvectors $\vec{C_i}$ are orthogonal and form a basis of the token space.

$$\vec{D'} = \sum \alpha_i \vec{C_i}$$
$$\vec{Q'} = \sum \beta_i \vec{C_i}$$

We can compare search documents matching query Q using $\vec{D'}.\vec{Q'} = \sum \alpha_i.\beta_i$ or cosim$(\vec{D'}, \vec{Q'})$:)

Low rank approximation

Theorem

$M^\top M$ is symmetric and its eigenvectors are orthogonal and form a basis of the token space.

Theorem (Eckart–Young–Mirsky)

The best[a] r-rank approximation $\hat{M}$ of M is given by the projection on the subspace formed by the eigenvectors of $M^\top M$ corresponding to the r biggest eigen values.

[a] In the sense minimizing $||M - \hat{M}||_F = \sum_{i,j}(m_{i,j} - \hat{m}_{i,j})^2$

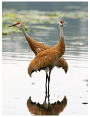
The projection to the low rank space (columns of $V^\top$ in SVD decomposition $M = U\Sigma V^\top$) collapse similar (i.e. *correlated*) tokens to the same component. This space is called the **Latent semantic space**.

Vector model: bright and dark side

The tf-idf vector model is good...

- Similarity based on information carried by tokens
- Flexible querying (latent semantics)
- Naturally rank documents
- Works well in practice

...but still not perfect:

- ignore polysemy  vs. 
- ignore the *truth* of the information 

Extras

How to store already crawled URLs?

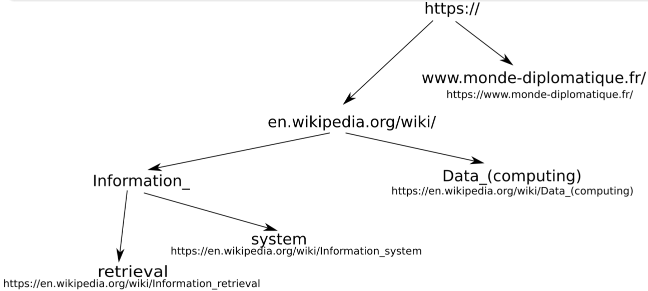
Exercise

What is the cost of checking if the crawler already visited a web page? Is it reasonable?

How to store already crawled URLs?

Exercise

What is the cost of checking if the crawler already visited a web page? Is it reasonable?

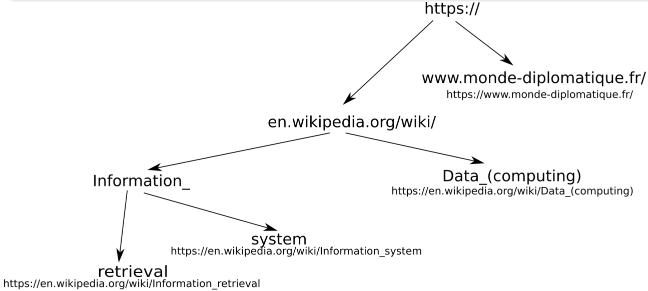


A *trie* structure.

How to store already crawled URLs?

Exercise

What is the cost of checking if the crawler already visited a web page? Is it reasonable?



A *trie* structure.

Exercise

What is the complexity in time?

Examples of (successful) companies in IR

ElasticSearch	Distributed, RESTful[5] search and analytics engine capable of solving a growing number of use cases. [...] centrally stores your data so you can discover the expected and uncover the unexpected.
swiftype	All-in-one relevance, lightning-fast setup and unprecedented control.
blekko	Now in IBM Watson

[5] i.e. based on HTTP