Information Retrieval

LBAW · Databases and Web Applications
MIEIC 20/21

Sérgio Nunes
DEI, FEUP, U.Porto

Agenda

- ➔ Introduction to Information Retrieval
- ➔ Search Engines Overview
- ➔ Information Retrieval Models
- ➔ Retrieval Efficiency
- ➔ Retrieval Evaluation
- ➔ Full Text Search in PostgreSQL

Introduction

Information Retrieval

- ➔ Information Retrieval deals with the representation, storage, organization of, and access to information items
- ➔ IR research includes:
  - ➔ Document and query modeling, web search, text classification, system architecture, user interfaces, data visualization, filtering
- ➔ Early example of *information retrieval systems* –> libraries
  - ➔ Manually built indexes and categories.

Historic Highlights

- ➔ First developments in the area of Information Retrieval started in the 50s, with pioneers such as Hans Peter Luhn and Eugene Garfield.
- ➔ In the 60s, the TF-IDF weighting scheme was developed as a result of work by Karen Spark Jones, Gerard Salton, and others. The probabilistic model was introduced in the 70s and the vector model in the 80s.
- ➔ Libraries were among the first institutions to adopt IR systems for retrieving information.
- ➔ The emergence of the Web, which has become the largest repository of knowledge in human history, put IR at the center of the stage.

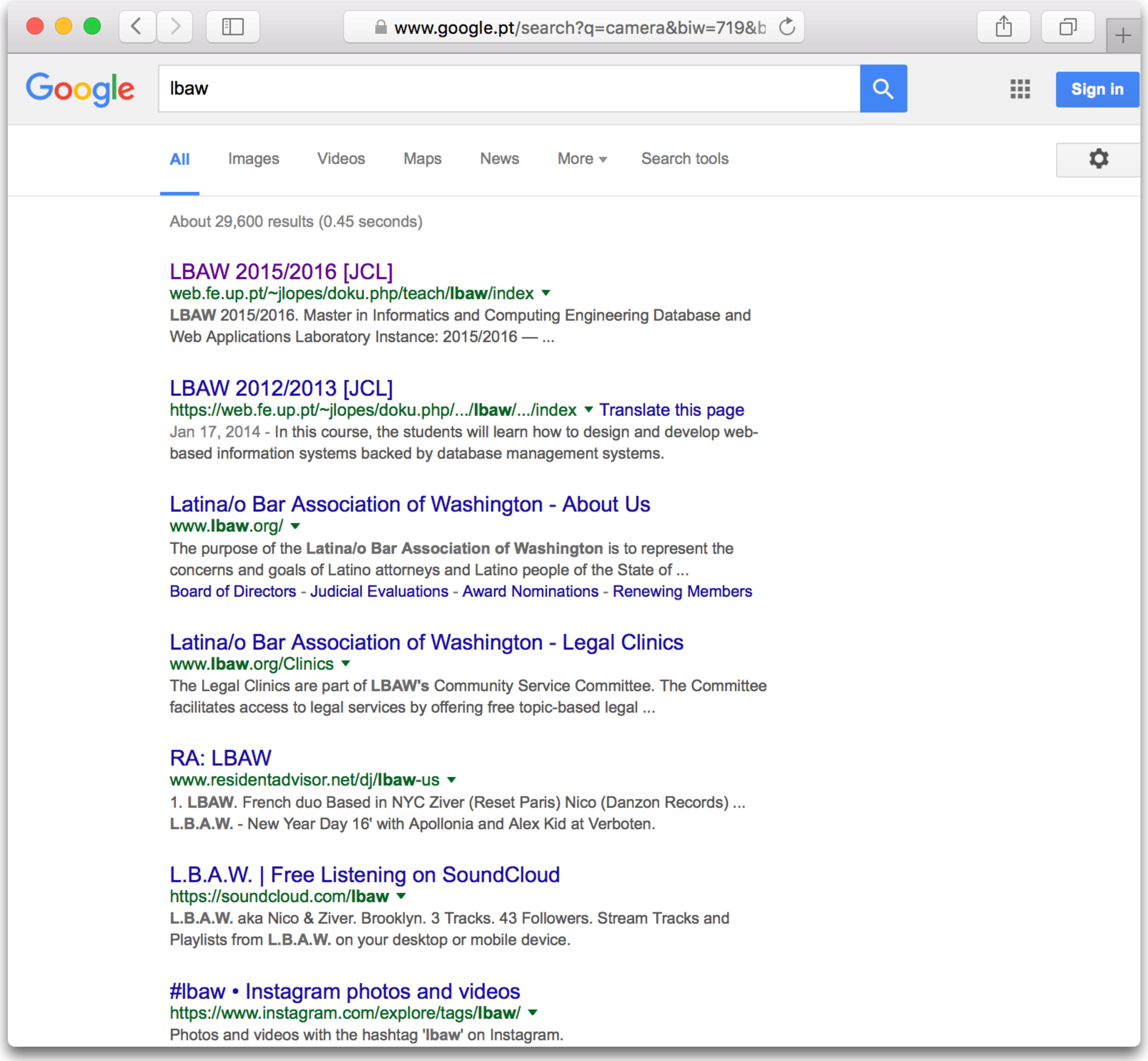
Central Issue

- ➔ The IR Problem
  - ➔ The key goal of an IR system is to retrieval all items that are relevant to a user query, representing an information need, while retrieving as few non relevant items as possible.
- ➔ The central concept in IR is the notion of relevance.

Motivation

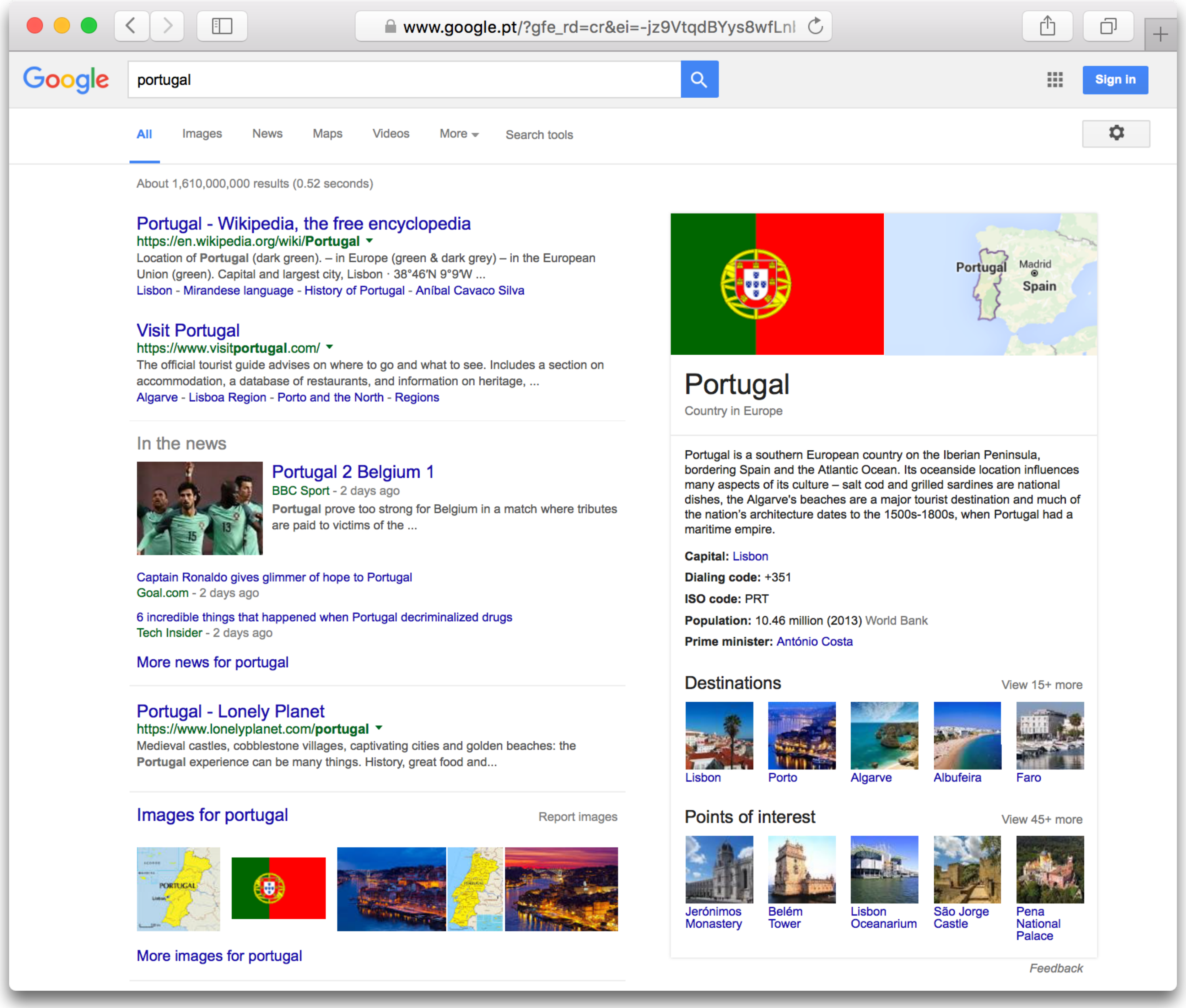
- ➔ RDBS provide set-based or data retrieval.
  - ➔ SELECT title, year FROM book
    WHERE title LIKE '%introduction%html%';
- ➔ **Limitations?**
  - ➔ There is no linguistic support (e.g. intro vs. introduction)
  - ➔ Difficult to search for multiple keywords (e.g. introduction to html vs. html introduction)
  - ➔ Degraded performance when dealing with a large number of documents.
  - ➔ No ranking of results (e.g. order by relevance)

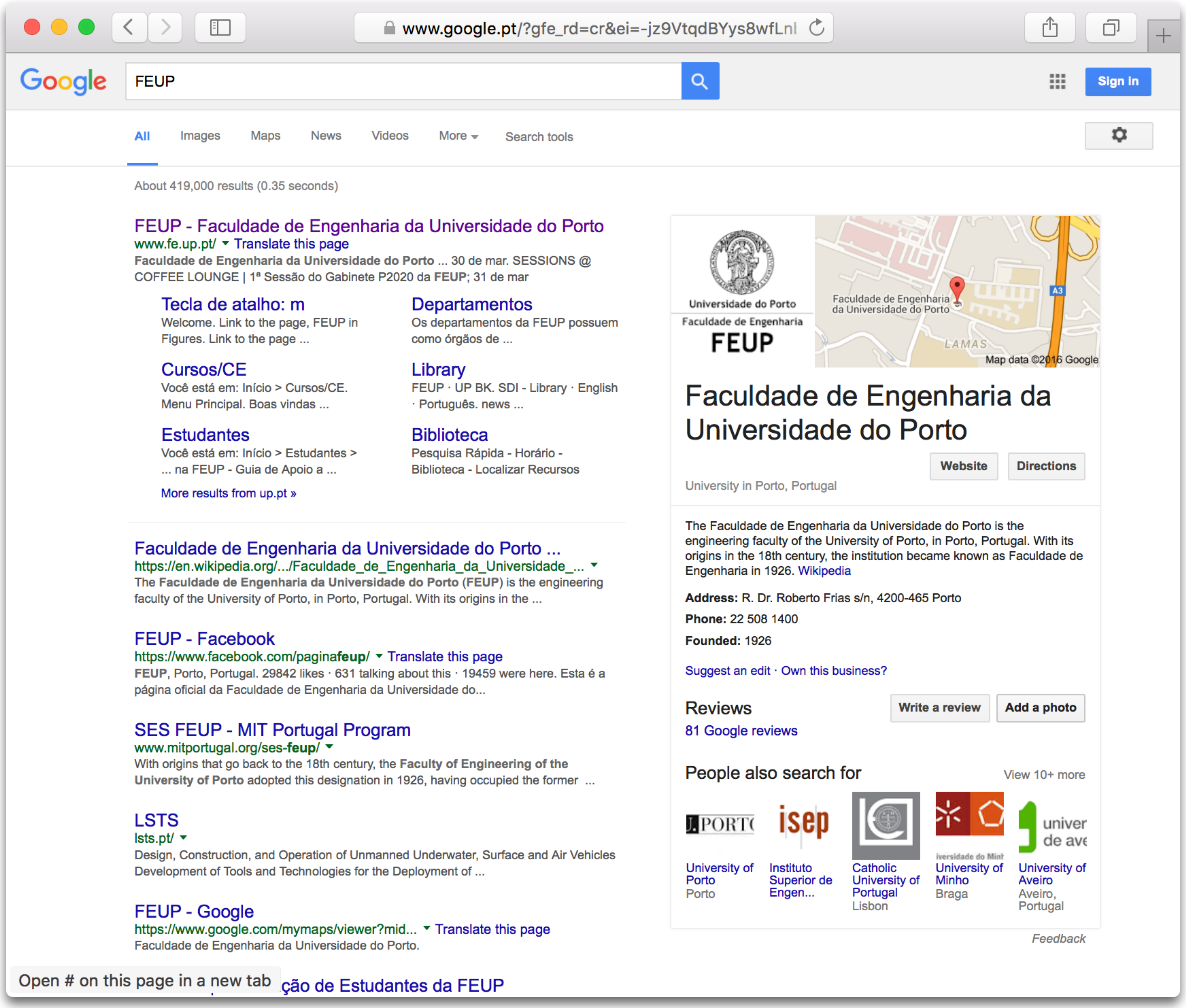
Web Search System

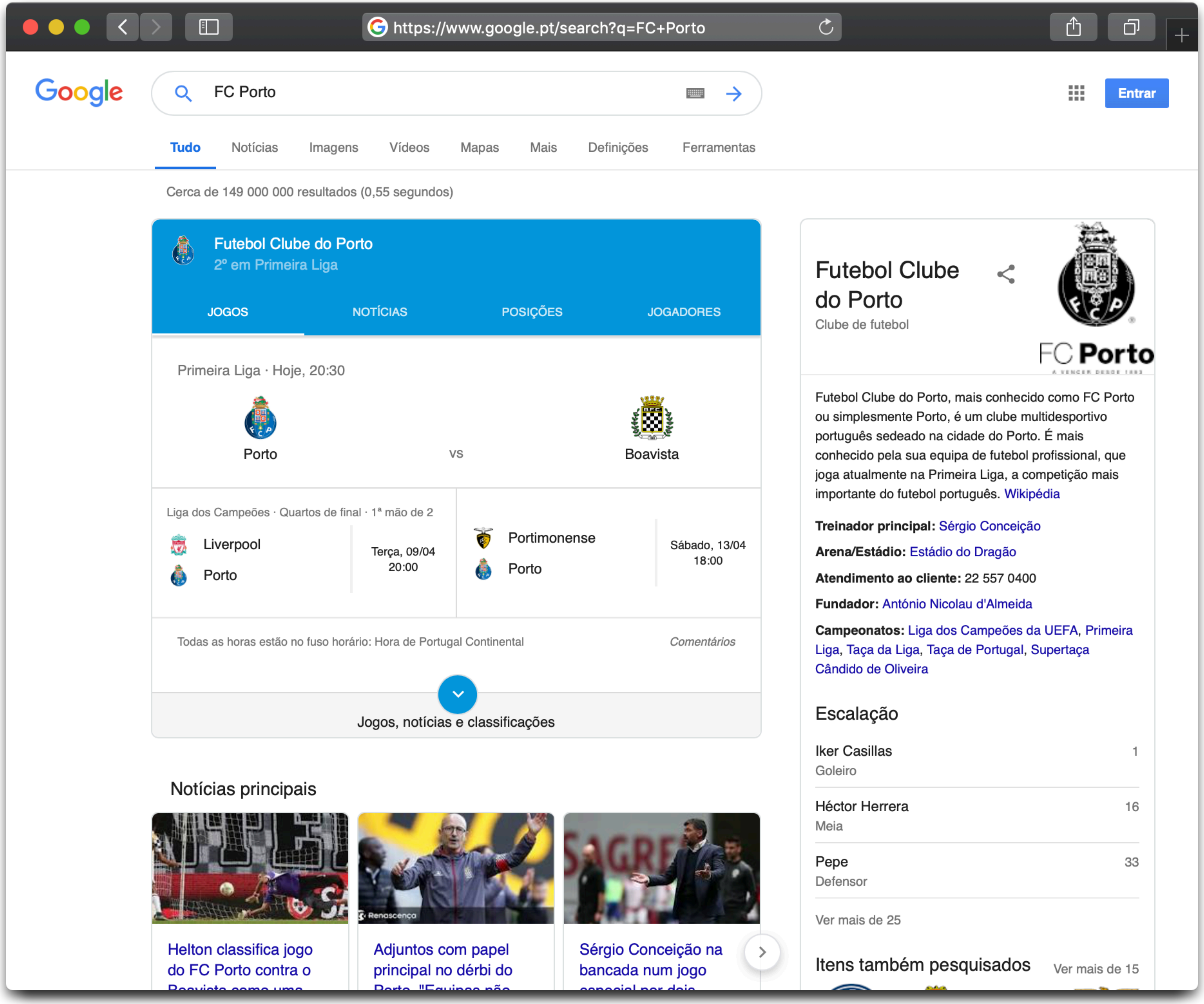


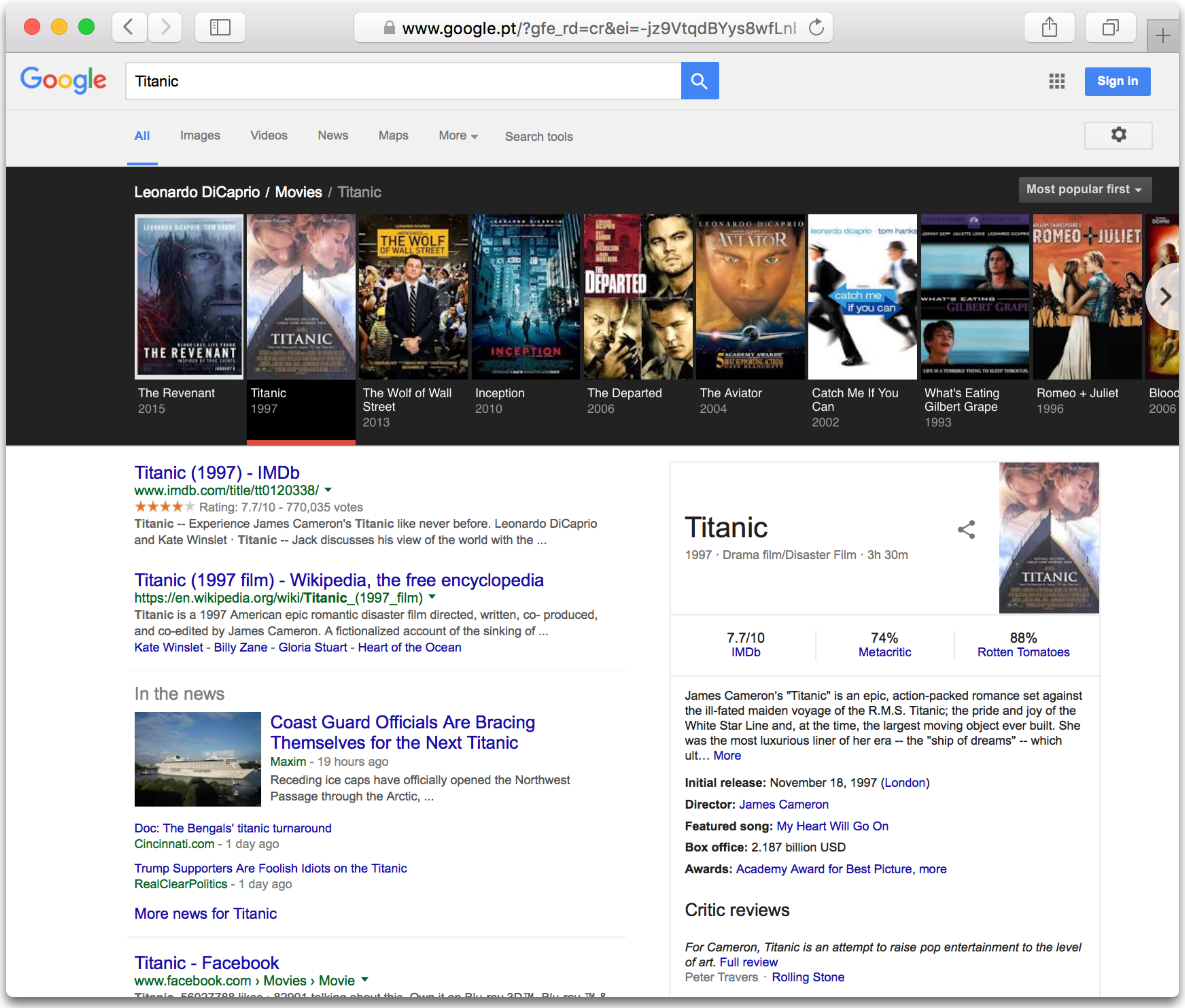
Trends

→ Users expect more than a pointer to a single document for a given information need (e.g. entities, relations).




www.google.pt/?gfe_rd=cr&ei=-jz9VtqdBYys8wfLnl
Google
FEUP
Sign in
All
Images
Maps
News
Videos
More
Search tools
About 419,000 results (0.35 seconds)
FEUP - Faculdade de Engenharia da Universidade do Porto
www.fe.up.pt/ Translate this page
Faculdade de Engenharia da Universidade do Porto ... 30 de mar. SESSIONS @ COFFEE LOUNGE | 1ª Sessão do Gabinete P2020 da FEUP; 31 de mar
Tecla de atalho: m
Welcome. Link to the page, FEUP in Figures. Link to the page ...
Departamentos
Os departamentos da FEUP possuem como órgãos de ...
Cursos/CE
Você está em: Início > Cursos/CE. Menu Principal. Boas vindas ...
Library
FEUP · UP BK. SDI - Library · English · Português. news ...
Estudantes
Você está em: Início > Estudantes > ... na FEUP - Guia de Apoio a ...
Biblioteca
Pesquisa Rápida - Horário - Biblioteca - Localizar Recursos
More results from up.pt »
Faculdade de Engenharia da Universidade do Porto ...
https://en.wikipedia.org/.../Faculdade_de_Engenharia_da_Universidade_...
The Faculdade de Engenharia da Universidade do Porto (FEUP) is the engineering faculty of the University of Porto, in Porto, Portugal. With its origins in the ...
FEUP - Facebook
https://www.facebook.com/paginafeup/ Translate this page
FEUP, Porto, Portugal. 29842 likes · 631 talking about this · 19459 were here. Esta é a página oficial da Faculdade de Engenharia da Universidade do...
SES FEUP - MIT Portugal Program
www.mitportugal.org/ses-feup/
With origins that go back to the 18th century, the Faculty of Engineering of the University of Porto adopted this designation in 1926, having occupied the former ...
LSTS
lsts.pt/
Design, Construction, and Operation of Unmanned Underwater, Surface and Air Vehicles Development of Tools and Technologies for the Deployment of ...
FEUP - Google
https://www.google.com/mymaps/viewer?mid... Translate this page
Faculdade de Engenharia da Universidade do Porto.
Open # on this page in a new tab
ção de Estudantes da FEUP
Universidade do Porto
Faculdade de Engenharia
FEUP
Faculdade de Engenharia da Universidade do Porto
LAMAS
Map data ©2016 Google
Faculdade de Engenharia da Universidade do Porto
Website
Directions
University in Porto, Portugal
The Faculdade de Engenharia da Universidade do Porto is the engineering faculty of the University of Porto, in Porto, Portugal. With its origins in the 18th century, the institution became known as Faculdade de Engenharia in 1926. Wikipedia
Address: R. Dr. Roberto Frias s/n, 4200-465 Porto
Phone: 22 508 1400
Founded: 1926
Suggest an edit · Own this business?
Reviews
Write a review
Add a photo
81 Google reviews
People also search for
View 10+ more
University of Porto
Porto
Instituto Superior de Engen...
Catholic University of Portugal
Lisbon
University of Minho
Braga
University of Aveiro
Aveiro, Portugal
Feedback


https://www.google.pt/search?q=FC+Porto
Google
FC Porto
Entrar
Tudo
Notícias
Imagens
Vídeos
Mapas
Mais
Definições
Ferramentas
Cerca de 149 000 000 resultados (0,55 segundos)
Futebol Clube do Porto
2º em Primeira Liga
JOGOS
NOTÍCIAS
POSIÇÕES
JOGADORES
Primeira Liga · Hoje, 20:30
Porto
vs
Boavista
Liga dos Campeões · Quartos de final · 1ª mão de 2
Liverpool
Porto
Terça, 09/04
20:00
Portimonense
Porto
Sábado, 13/04
18:00
Todas as horas estão no fuso horário: Hora de Portugal Continental
Comentários
Jogos, notícias e classificações
Notícias principais
Helton classifica jogo
do FC Porto contra o
Adjuntos com papel
principal no dérbi do
Sérgio Conceição na
bancada num jogo
Futebol Clube
do Porto
Clube de futebol
FC Porto
Futebol Clube do Porto, mais conhecido como FC Porto ou simplesmente Porto, é um clube multidesportivo português sedeado na cidade do Porto. É mais conhecido pela sua equipa de futebol profissional, que joga atualmente na Primeira Liga, a competição mais importante do futebol português. Wikipédia
Treinador principal: Sérgio Conceição
Arena/Estádio: Estádio do Dragão
Atendimento ao cliente: 22 557 0400
Fundador: António Nicolau d'Almeida
Campeonatos: Liga dos Campeões da UEFA, Primeira Liga, Taça da Liga, Taça de Portugal, Supertaça Cândido de Oliveira
Escalação
Iker Casillas
1
Goleiro
Héctor Herrera
16
Meia
Pepe
33
Defensor
Ver mais de 25
Itens também pesquisados
Ver mais de 15



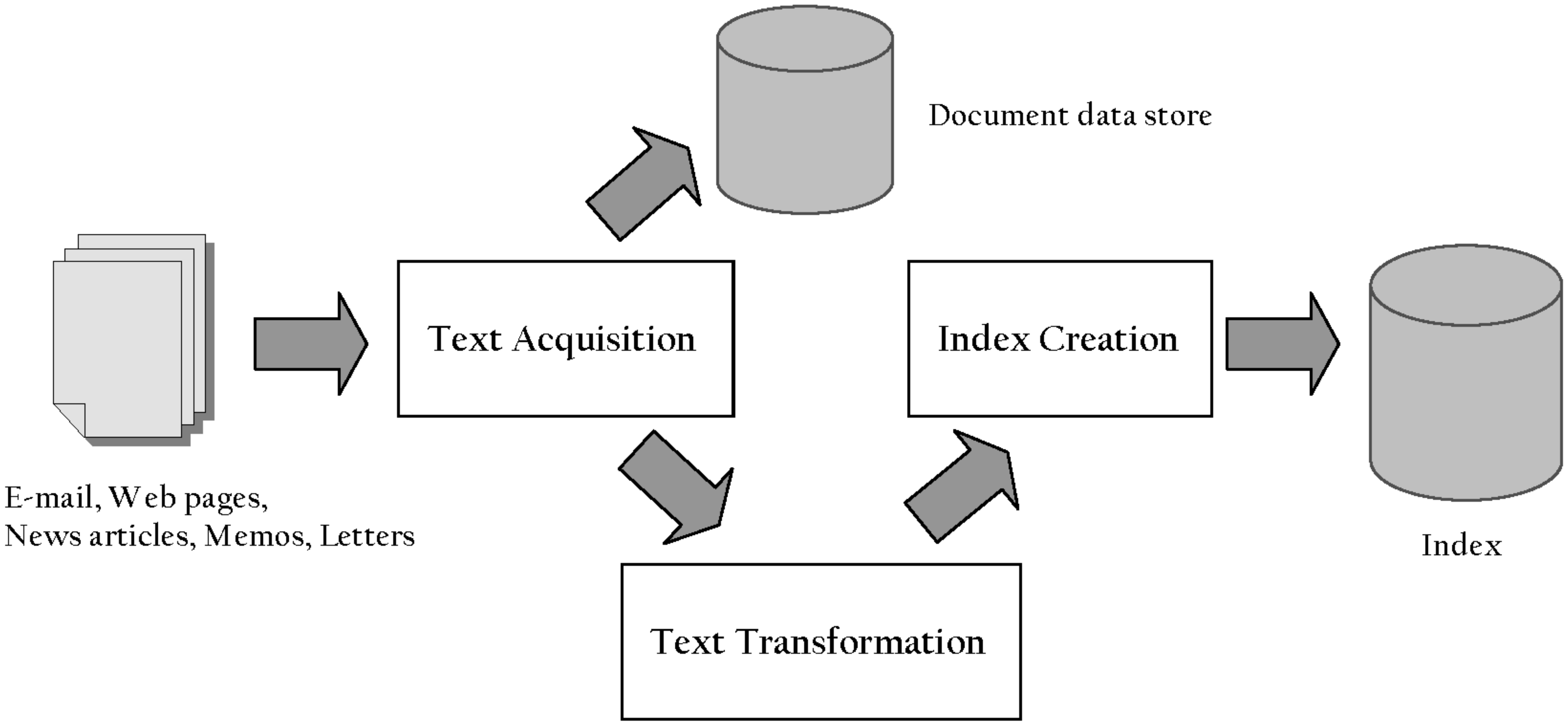
www.google.pt/?gfe_rd=cr&ei=-jz9VtqdBYys8wfLnl
Google
Titanic
Sign in
All
Images
Videos
News
Maps
More
Search tools
Leonardo DiCaprio / Movies / Titanic
Most popular first
The Revenant 2015
Titanic 1997
The Wolf of Wall Street 2013
Inception 2010
The Departed 2006
The Aviator 2004
Catch Me If You Can 2002
What's Eating Gilbert Grape 1993
Romeo + Juliet 1996
Titanic (1997) - IMDb
www.imdb.com/title/tt0120338/
Rating: 7.7/10 - 770,035 votes
Titanic -- Experience James Cameron's Titanic like never before. Leonardo DiCaprio and Kate Winslet · Titanic -- Jack discusses his view of the world with the ...
Titanic (1997 film) - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Titanic_(1997_film)
Titanic is a 1997 American epic romantic disaster film directed, written, co- produced, and co-edited by James Cameron. A fictionalized account of the sinking of ...
Kate Winslet - Billy Zane - Gloria Stuart - Heart of the Ocean
In the news
Coast Guard Officials Are Bracing Themselves for the Next Titanic
Maxim - 19 hours ago
Receding ice caps have officially opened the Northwest Passage through the Arctic, ...
Doc: The Bengals' titanic turnaround
Cincinnati.com - 1 day ago
Trump Supporters Are Foolish Idiots on the Titanic
RealClearPolitics - 1 day ago
More news for Titanic
Titanic - Facebook
www.facebook.com › Movies › Movie
Titanic
1997 · Drama film/Disaster Film · 3h 30m
7.7/10 IMDb
74% Metacritic
88% Rotten Tomatoes
James Cameron's "Titanic" is an epic, action-packed romance set against the ill-fated maiden voyage of the R.M.S. Titanic; the pride and joy of the White Star Line and, at the time, the largest moving object ever built. She was the most luxurious liner of her era -- the "ship of dreams" -- which ult... More
Initial release: November 18, 1997 (London)
Director: James Cameron
Featured song: My Heart Will Go On
Box office: 2.187 billion USD
Awards: Academy Award for Best Picture, more
Critic reviews
For Cameron, Titanic is an attempt to raise pop entertainment to the level of art. Full review
Peter Travers · Rolling Stone

Search Engines

Search Engine Architecture

- ➔ The architecture of search engines can be divided in two main processes
  - ➔ **the indexing process** – offline, when collection changes
  - ➔ **the querying process** – online, in response to user queries

Indexing Process

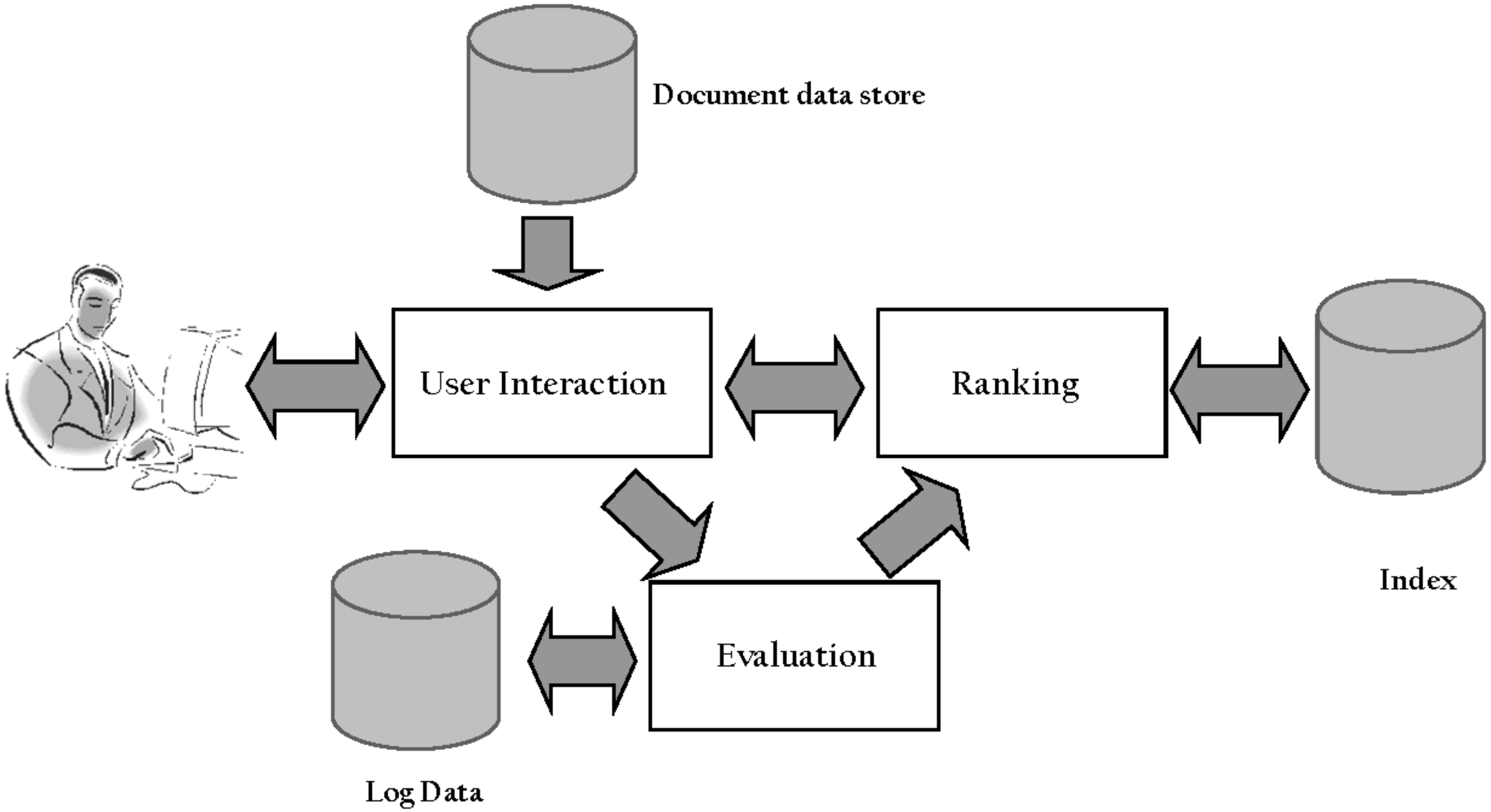


Croft, Metzler, Strohman (2010), Search Engines: Information Retrieval in Practice

Indexing Process

- ➔ Text Acquisition
  - ➔ identifies (finds) and stores documents for indexing
- ➔ Text Transformation
  - ➔ transforms documents into index terms or features
- ➔ Index Creation
  - ➔ takes index terms and creates data structures to support fast searching

Query Process



Croft, Metzler, Strohman (2010), Search Engines: Information Retrieval in Practice

Query Process

- ➔ User Interaction
  - ➔ supports creation and refinement of queries; display of results
- ➔ Ranking
  - ➔ use query and index to generate ranked list of results
- ➔ Evaluation
  - ➔ monitors and measures effectiveness and efficiency

Ranking Signals

- ➔ Estimating each document relevance for a given user query and context is done using various sources of information, usually called signals.
- ➔ **Which signals are used by a search engine?**
  - ➔ Keywords in the document.
  - ➔ Origin of the document (e.g. up.pt, publico.pt, .gov.pt)
  - ➔ References (i.e. links) to the document.
  - ➔ Information about the user (e.g. previous searches and clicks, location, network, browser used, device used).
  - ➔ Much more …

Ranking Signals

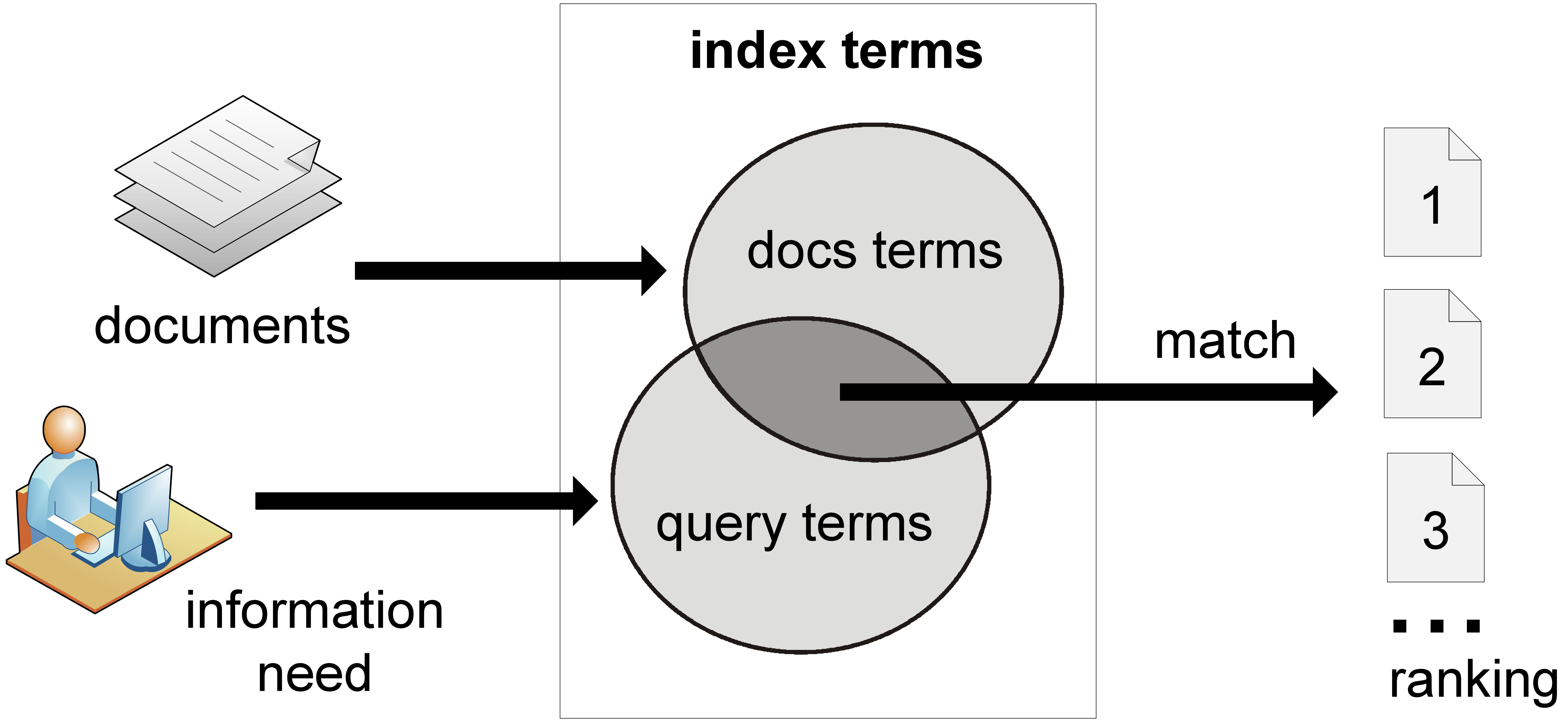
- ➔ Web search engines use hundreds of signals, also called features.
- ➔ These signals can be divided in two groups
  - ➔ static signals that can be computed during the indexing process, e.g. length of document, age of document, number of links to document, etc.
  - ➔ query-dependent signals that are only available at query time, e.g. number of query terms, time of day, query terms in document, etc.
- ➔ Signals can also be divided according to their source:
  - ➔ Document-based, Collection-based, User-based

Information Retrieval Models

Information Retrieval Models

- ➔ Information Retrieval modeling is a process aimed at producing a ranking function
- ➔ The process consists of two main tasks
  - ➔ The conception of a logical framework for representing documents and queries
  - ➔ The definition of a ranking function that allows quantifying the similarities among documents and queries.

Information Retrieval Process



Chap 03: Modeling, Baeza-Yates & Ribeiro-Neto, Modern Information Retrieval, 2nd Edition – p. 4

The Term-Document Matrix

➔ The term-document matrix is a basic concept that represents the relation between indexed terms and collection documents.

➔ Also called incidence matrix.

$$
\begin{array}{cc}
 & \begin{array}{cc} d_1 & d_2 \end{array} \\
\begin{array}{c} k_1 \\ k_2 \\ k_3 \end{array} & \left[\begin{array}{cc} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \\ f_{3,1} & f_{3,2} \end{array}\right]
\end{array}
$$

where each $f_{i,j}$ element stands for the frequency of term k_i in document d_j

Chap 03: Modeling, Baeza-Yates & Ribeiro-Neto, Modern Information Retrieval, 2nd Edition – p. 13

Term Weighting

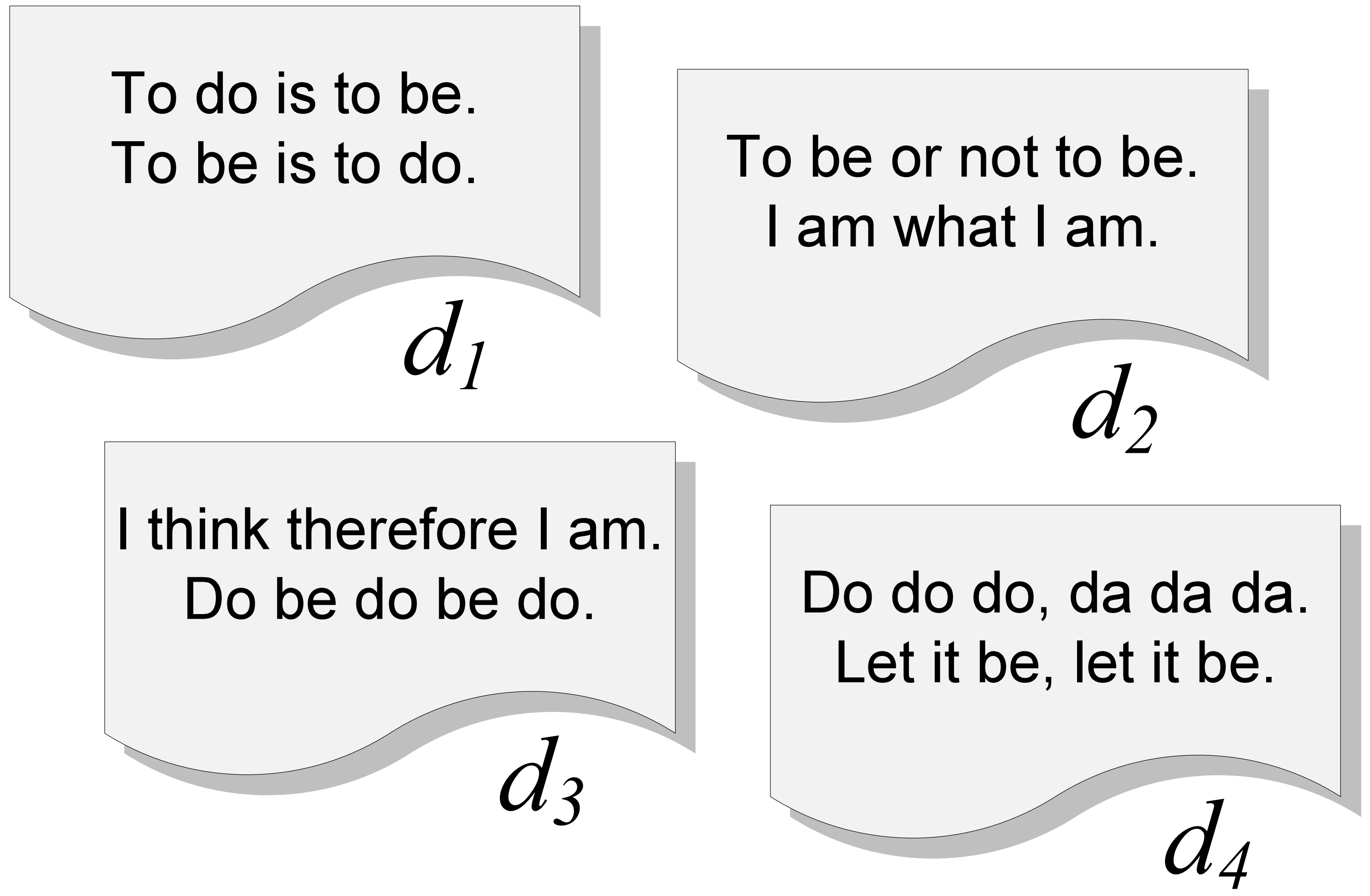
➔ Terms are not equally useful for describing a document.

➔ **Term weights** quantify the importance of a given index term for describing the contents of a document.

$$f(do, d_1) = 2$$
$$f(do, d_2) = 0$$
$$f(do, d_3) = 3$$
$$f(do, d_4) = 3$$

$$F(do) = 8$$

$$n(do) = 3$$



Chap 03: Modeling, Baeza-Yates & Ribeiro-Neto, Modern Information Retrieval, 2nd Edition – p. 27

Term Frequency

➔ Term frequency can be used as an estimation of the term importance for a given document.

➔ However, it can be easily manipulated.

Quasi architecto

Sed ut perspiciatis unde omnis iste natus error sit **flowers** accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo **flowers** veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Nemo enim **flowers** voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

TF("flowers") = 3

Quasi architecto

Sed ut **flowers** unde omnis **flowers** natus error sit **flowers** accusantium **flowers** laudantium, totam rem aperiam, eaque ipsa quae ab illo **flowers** veritatis et quasi **flowers** beatae vitae dicta sunt explicabo.

Nemo enim **flowers** voluptatem quia voluptas sit aspernatur aut **flowers** aut fugit, sed quia **flowers** magni dolores eos qui ratione voluptatem sequi **flowers**.

TF("flowers") = 10

Quasi architecto

flowers ut **flowers** **flowers** omnis **flowers** **flowers** **flowers** sit **flowers** **flowers** **flowers** **flowers**, totam **flowers** aperiam, **flowers** ipsa **flowers** ab **flowers** **flowers** **flowers** et quasi **flowers** **flowers** **flowers** dicta **flowers**.

flowers enim **flowers** **flowers** quia **flowers** **flowers** **flowers** aut **flowers** aut **flowers**, **flowers** quia **flowers** **flowers** dolores **flowers** qui **flowers** **flowers** sequi **flowers**.

TF("flowers") = ∞

Inverse Document Frequency

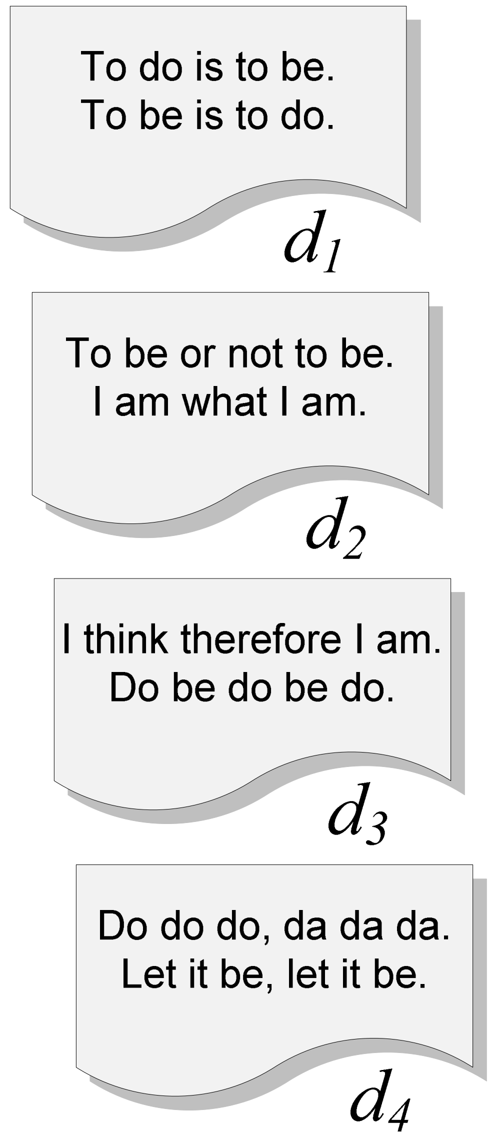
- An important, but less intuitive measure, is the inverse document frequency (IDF) of a term.
- Terms that appear in fewer documents of a collection have more discriminative power, thus are given a higher weight. Also referred to as the specificity of a term.

$$IDF(term) = \frac{|Documents\ in\ collection|}{|Documents\ containing\ term|}$$

TF-IDF

➔ The best known term weighting scheme uses weights that combine term frequency with inverse document frequency, known as TF-IDF.

➔ tf-idf(*term, document, collection*) = tf(*term, document*) x idf(*term, collection*)



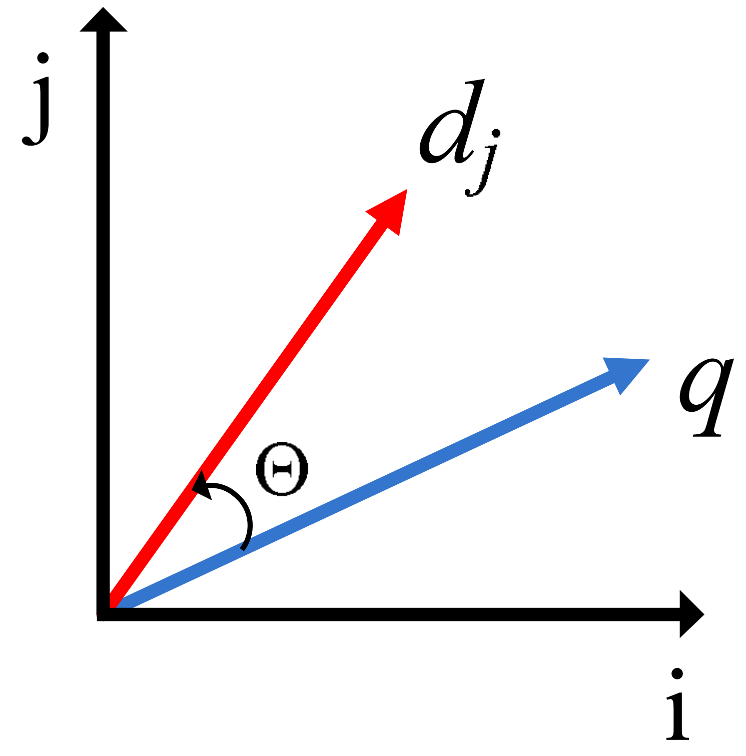
		d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

Chap 03: Modeling, Baeza-Yates & Ribeiro-Neto, Modern Information Retrieval, 2nd Edition – p. 43

Vector Space Model

➔ Binary weights are too limiting. The vector space model proposes a framework in which partial matching is possible.

➔ Documents, and queries, are represented as unary vectors in a n-dimensional space. The similarity between two different documents is obtained using the cosine between these vectors.



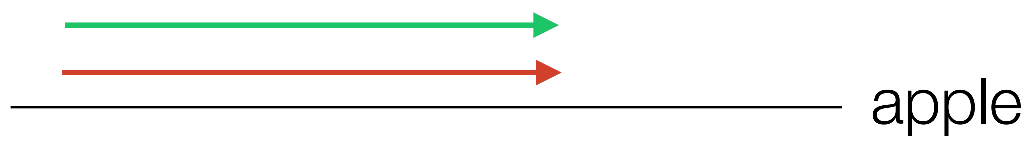
$$cos(\theta) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|}$$

Vector Model Example

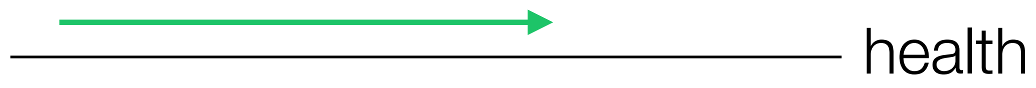
➔ Considering the following two sentences:

- ➔ s1: apples are good for your health
- ➔ s2: apples are fruits that grown on trees

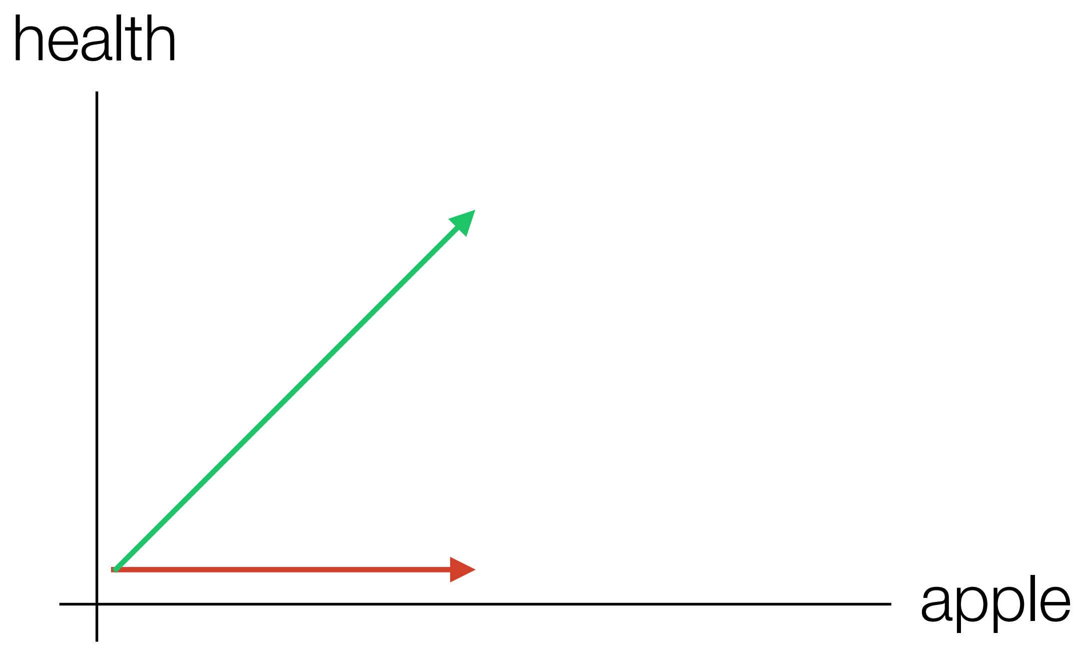
➔ We can represent these two documents in vector spaces, considering n-dimensions.



1-dimension: apple



1-dimension: health

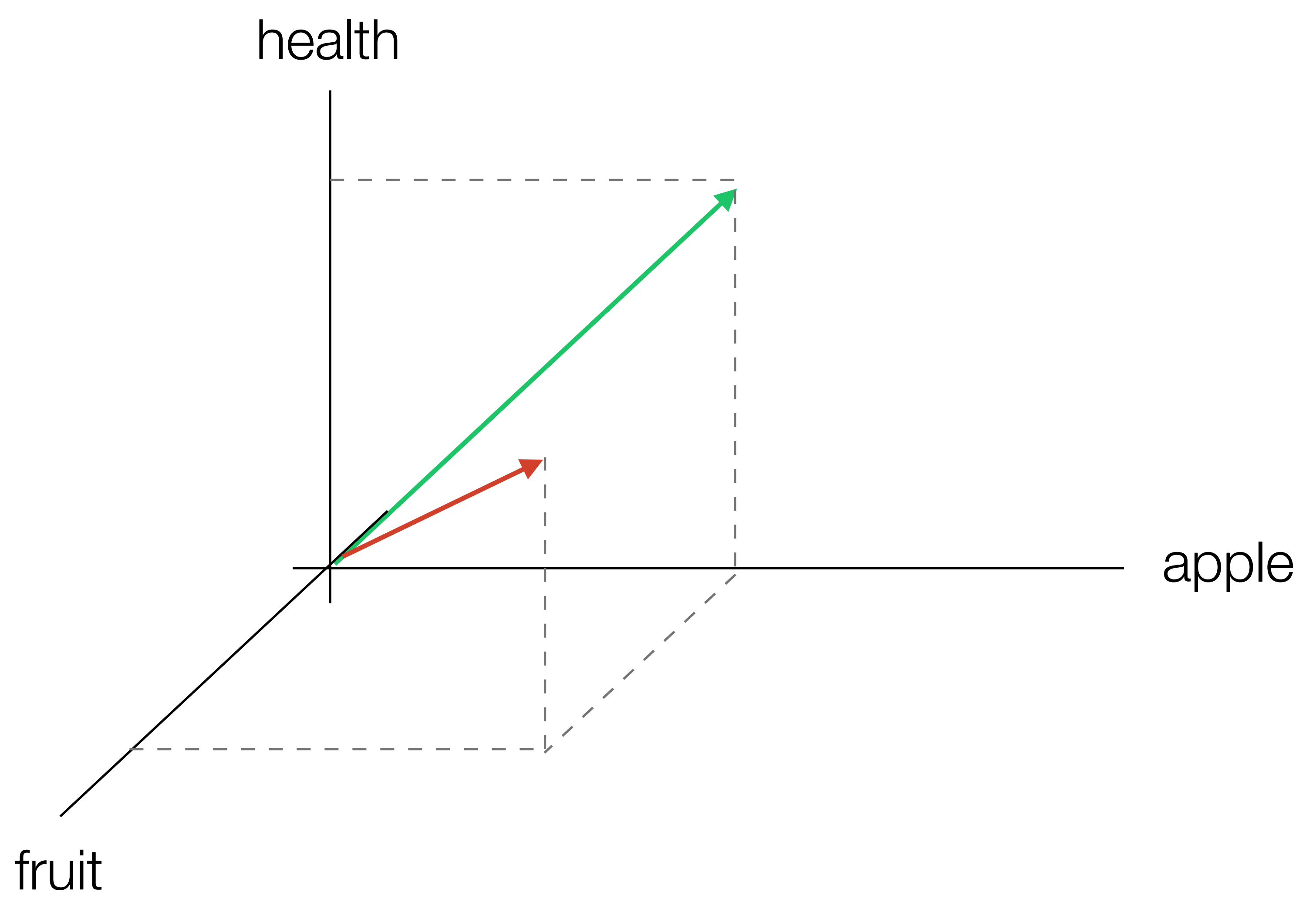


2-dimensions: apple, health

Vector Model Example

➔ Considering the following two sentences:

- ➔ s1: apples are good for your health
- ➔ s2: apples are fruits that grown on trees



3-dimensions: apple, health, fruit

Search Engine Ranking

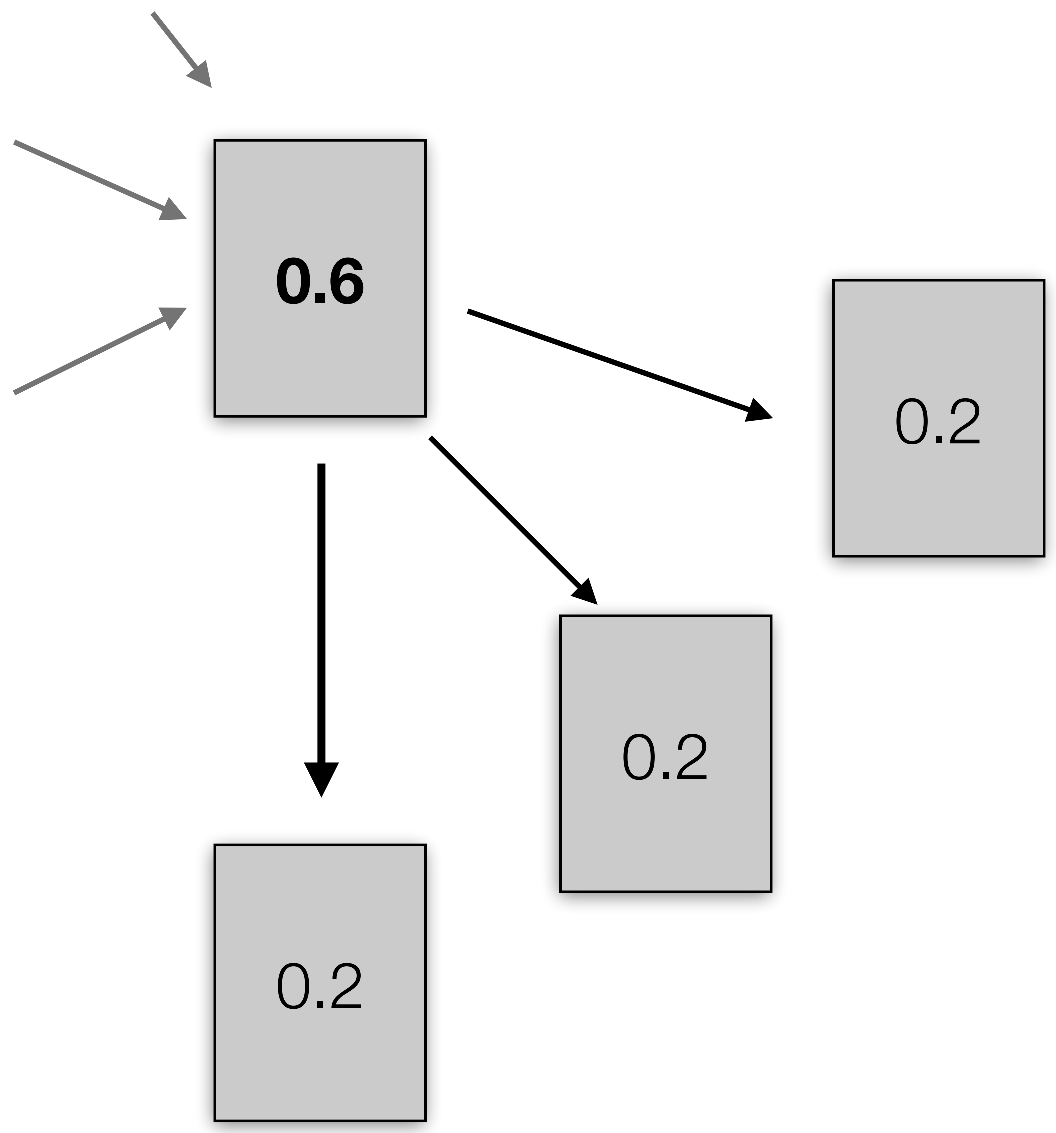
Link-based Signals

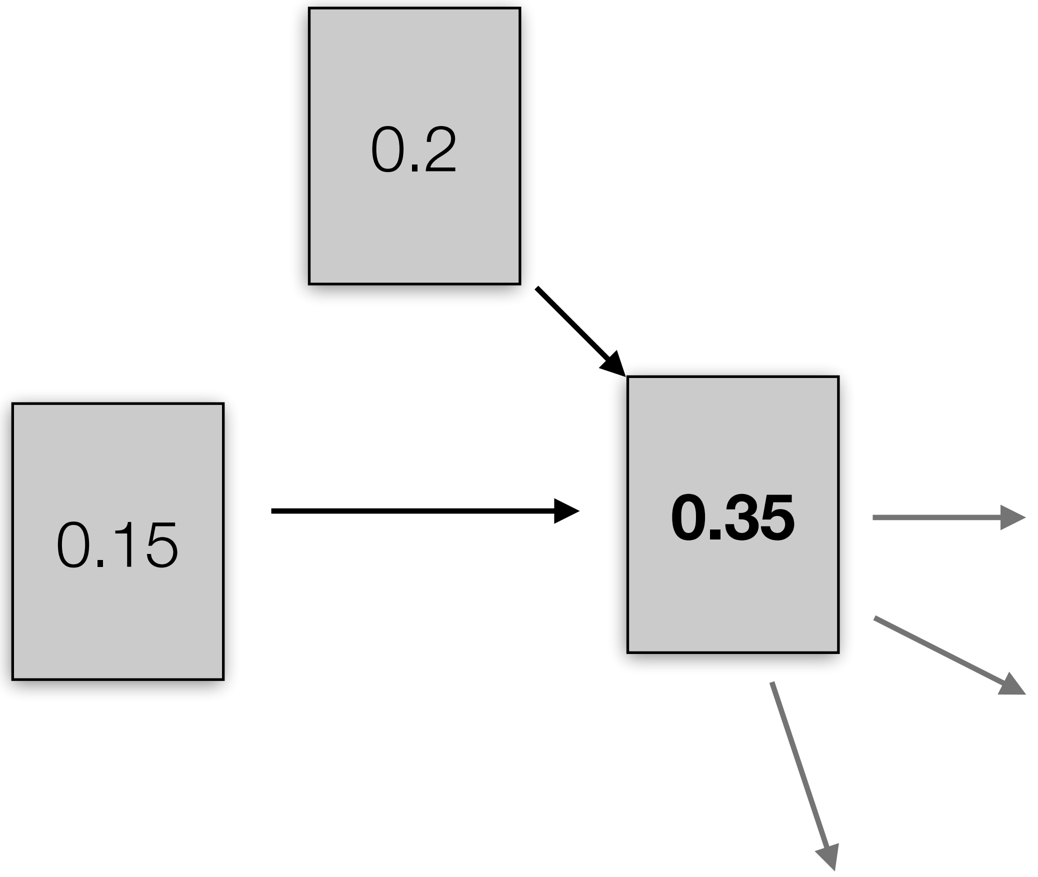
PageRank

- ➔ The web is a directed graph.
- ➔ The hyperlinks pointing to a given page has been used as a measure of quality of that page.
- ➔ Simple approach: use the number of links to a page (i.e. in-degree) as a ranking signal.
- ➔ The best known link-based ranking signal is the PageRank, developed at Stanford (during Larry Page's PhD) and used by Google in their ranking strategy. PageRank is a query-independent score.
- ➔ A link-based, query-dependent alternative, is the HITS algorithm, developed by Jon Kleinberg in 1999. HITS produces two independent scores for each page, an authority score and a hub score.
    - ➔ An authority is a page with many citations from hubs.
    - ➔ A hub is a page that cites a large number of authorities.

PageRank Example

- ➔ PageRank is computed iteratively.
- ➔ All nodes (web documents) start with the same initial value, e.g. 1/N.
- ➔ The score of each node is distributed to the documents that it links to, until the score of each node converges.





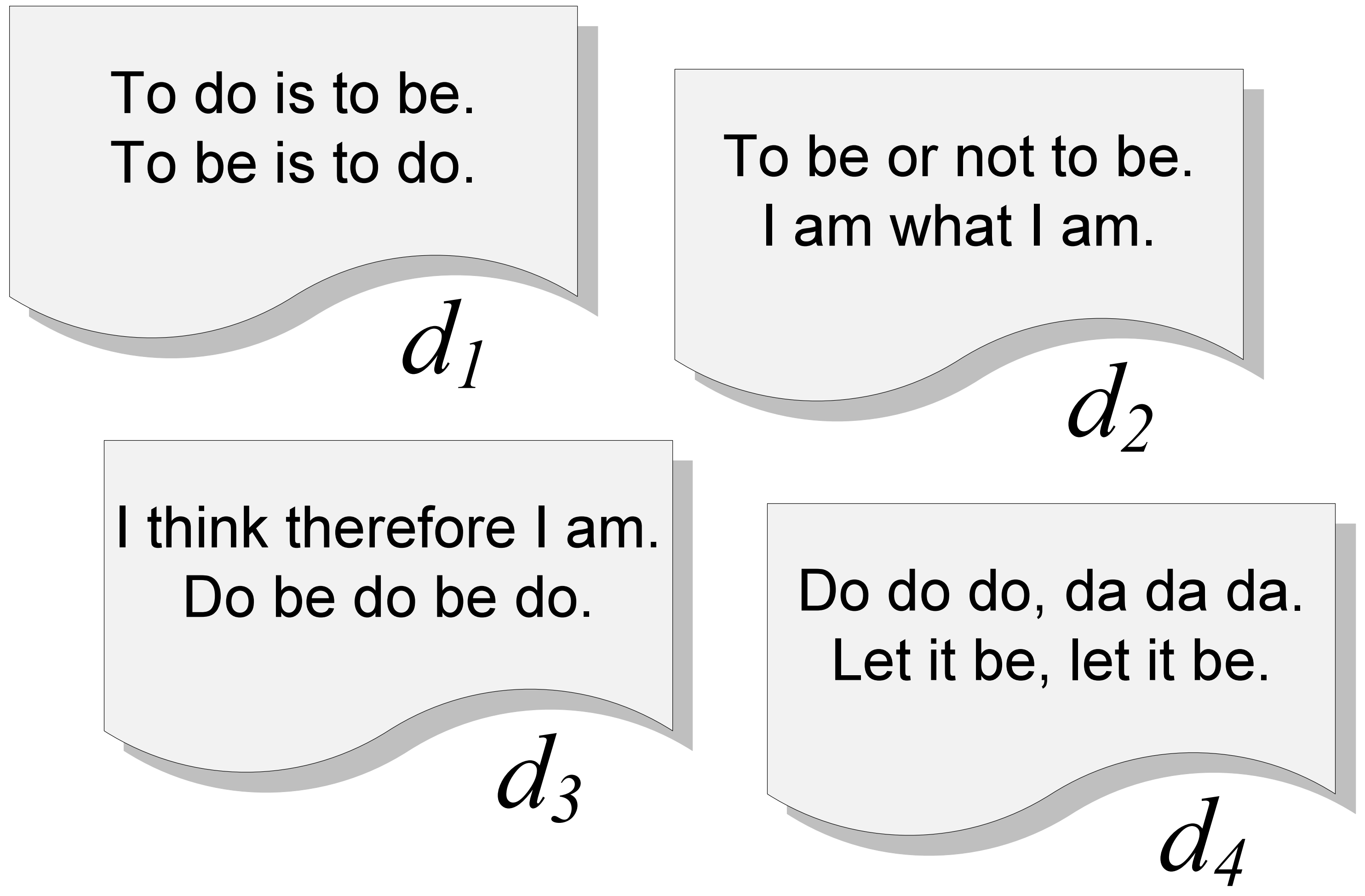
Retrieval Efficiency

Efficiency in Information Retrieval

- ➔ The goal is to process user queries with minimal requirements of computational resources.
- ➔ The inverted index is a word-based data structure built to speed up access.
- ➔ The inverted index structure is composed of two elements: the vocabulary and the occurrences.
  - ➔ The vocabulary is the set of all different words
  - ➔ For each word the index stores the document which contain that word

Basic Inverted Index

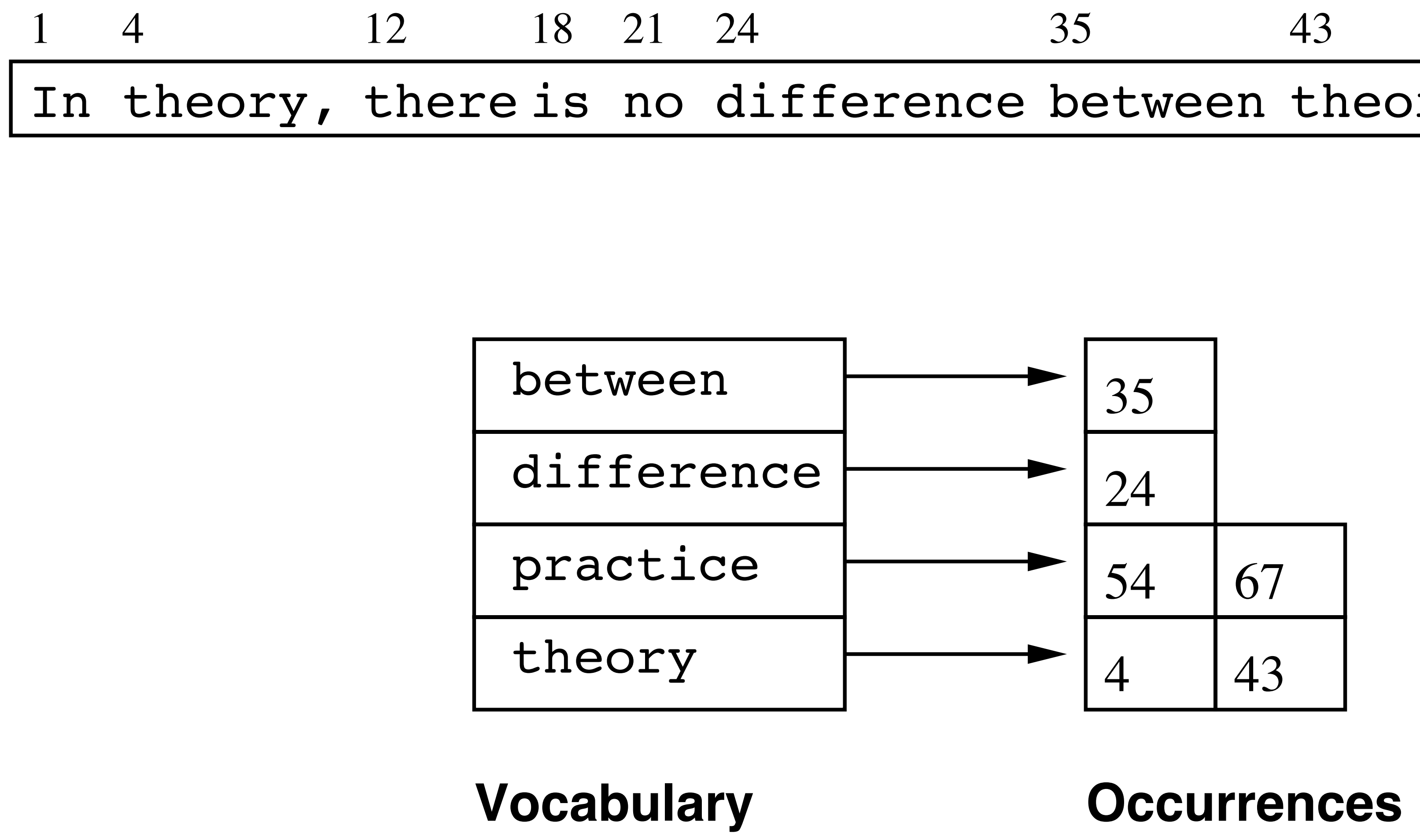
Vocabulary	n_i	Occurrences as inverted lists
to	2	[1,4],[2,2]
do	3	[1,2],[3,3],[4,3]
is	1	[1,2]
be	4	[1,2],[2,2],[3,2],[4,2]
or	1	[2,1]
not	1	[2,1]
I	2	[2,2],[3,2]
am	2	[2,2],[3,1]
what	1	[2,1]
think	1	[3,1]
therefore	1	[3,1]
da	1	[4,3]
let	1	[4,2]
it	1	[4,2]



Full Inverted Index

➔ The basic index is not suitable for answering phrase or proximity queries.

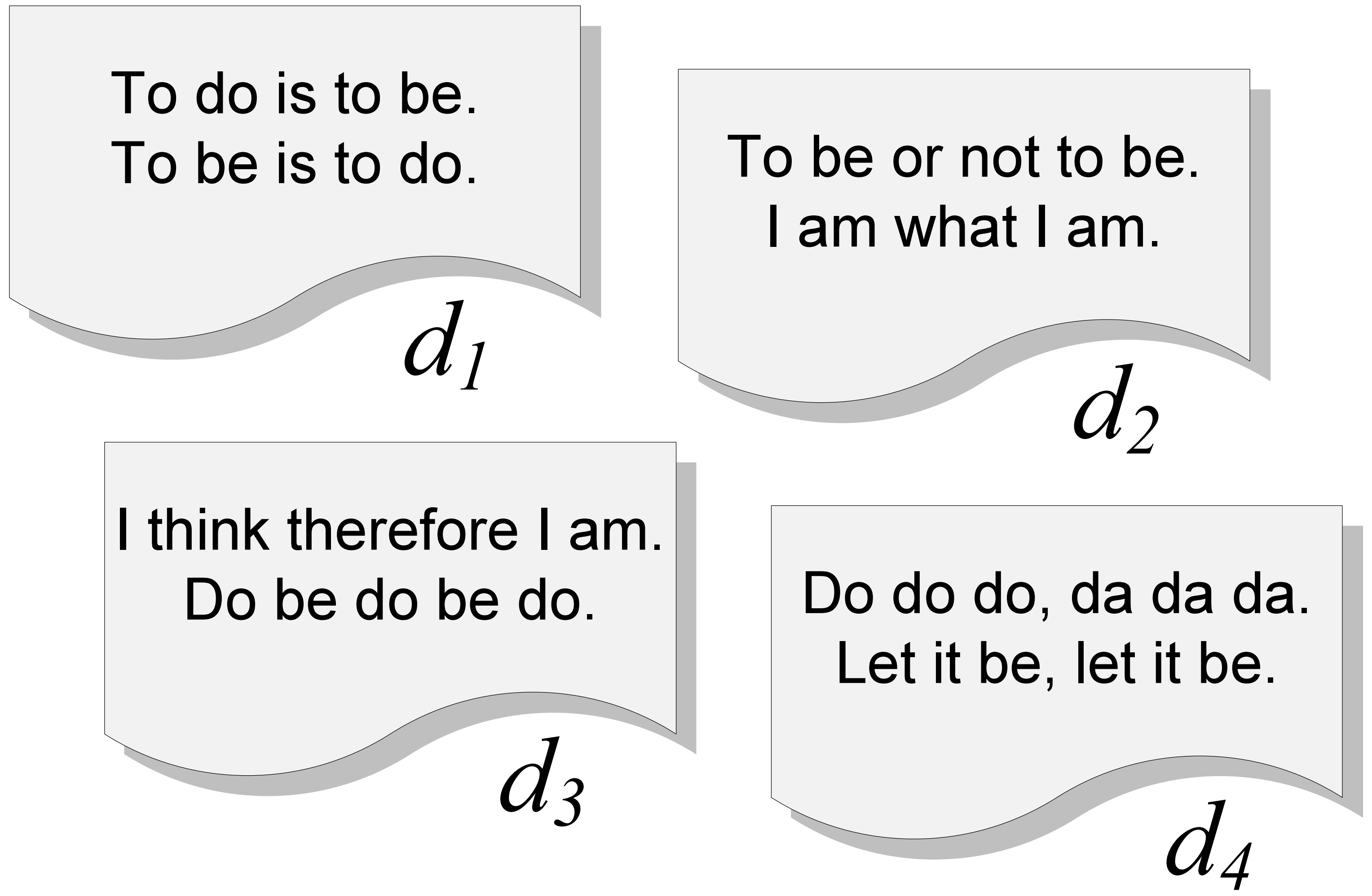
➔ Hence, we need to add the position of each word in each document to the index.



Indexing and Searching, Modern Information Retrieval, Addison Wesley, 2010 – p. 11

Full Inverted Index

Vocabulary	n_i	Occurrences as full inverted lists
to	2	[1,4,[1,4,6,9]],[2,2,[1,5]]
do	3	[1,2,[2,10]],[3,3,[6,8,10]],[4,3,[1,2,3]]
is	1	[1,2,[3,8]]
be	4	[1,2,[5,7]],[2,2,[2,6]],[3,2,[7,9]],[4,2,[9,12]]
or	1	[2,1,[3]]
not	1	[2,1,[4]]
I	2	[2,2,[7,10]],[3,2,[1,4]]
am	2	[2,2,[8,11]],[3,1,[5]]
what	1	[2,1,[9]]
think	1	[3,1,[2]]
therefore	1	[3,1,[3]]
da	1	[4,3,[4,5,6]]
let	1	[4,2,[7,10]]
it	1	[4,2,[8,11]]



Retrieval Evaluation

Retrieval Evaluation

- ➔ How to evaluate how well the system is responding to users' queries?
- ➔ The field of Information Retrieval has a long tradition of measuring and evaluating the performance of retrieval systems. Well-known measures such as Precision and Recall were proposed in this area.
- ➔ Retrieval evaluation is a critical component of any modern search system to:
  - ➔ Determine how well a system is performing and evaluate changes.
  - ➔ Compare the performance of a system with others.
- ➔ Challenging, compared to tradicional areas where performance can be measured using objetive metrics such as space, speed, size, etc.

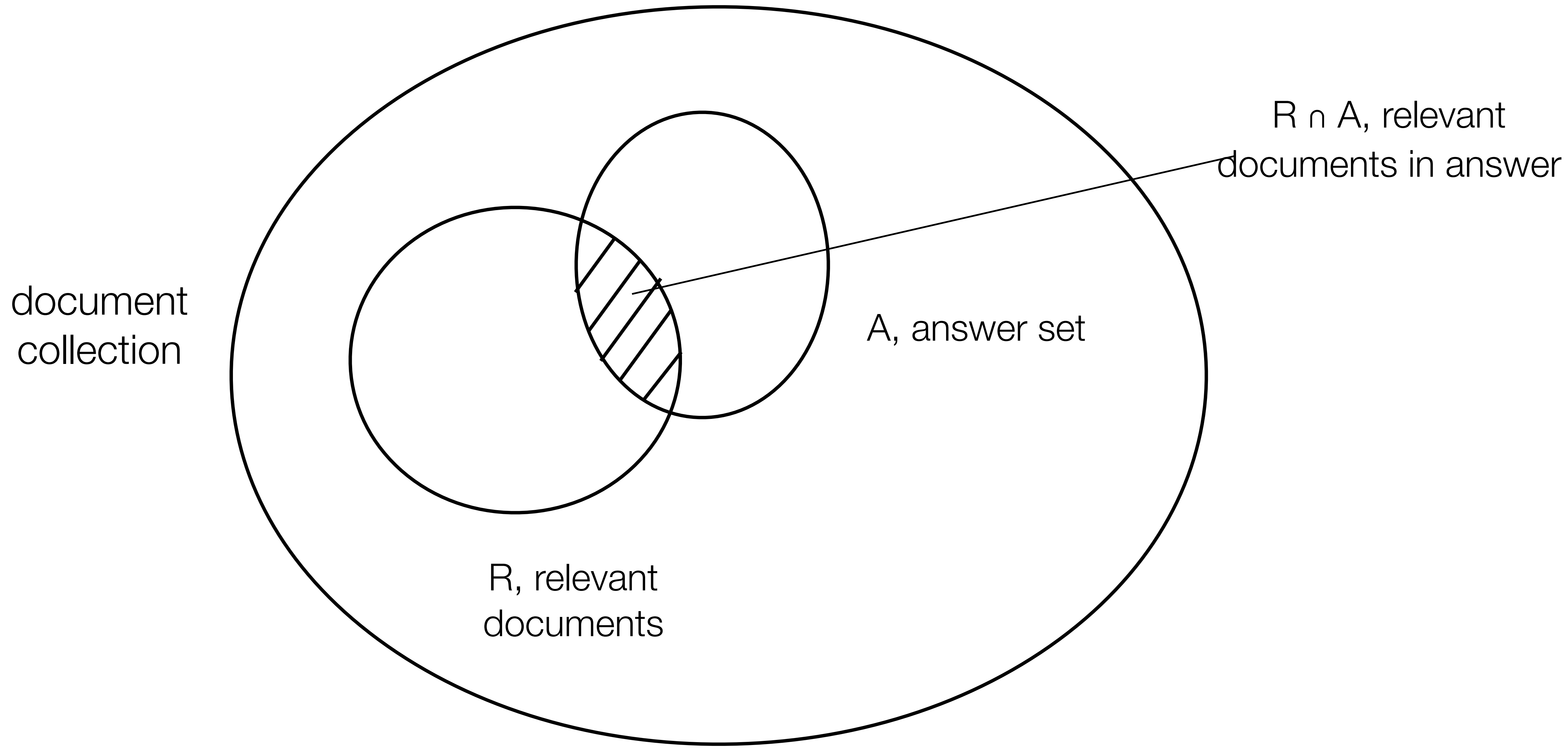
Precision and Recall

- Consider,
  - R, set of relevant documents in the collection.
  - A, set of documents in the retrieved answer.
- We can define the two core measures in IR evaluation,
  - Precision is the fraction of the retrieved documents that are relevant.
  - Recall is the fraction of the relevant documents that are retrieved.

Precision and Recall

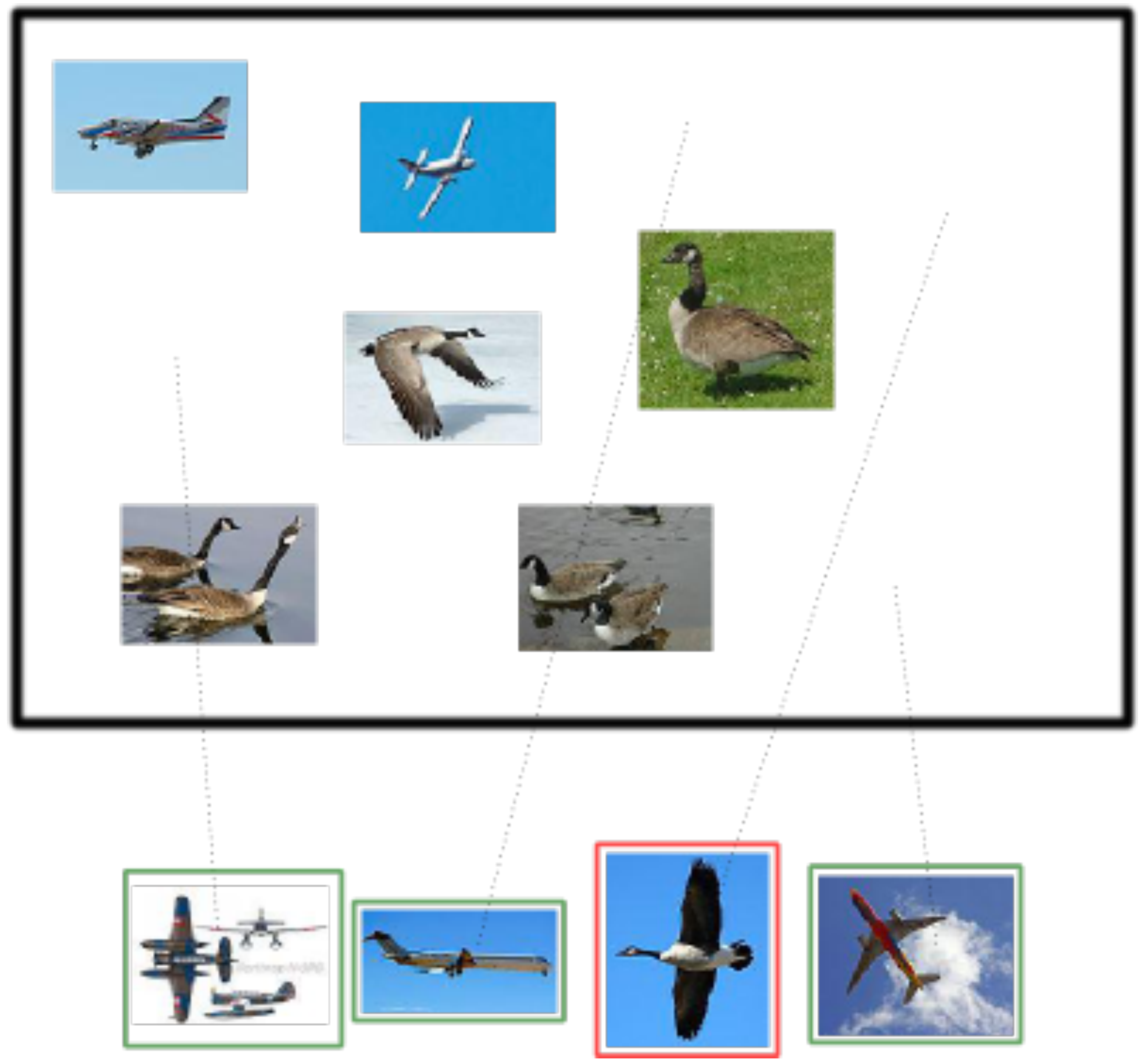
$$Precision = \frac{|R \cap A|}{|A|}$$

$$Recall = \frac{|R \cap A|}{|R|}$$



Precision and Recall Example

➔ For the following system, calculate precision and recall when searching for [airplane].



	relevant	not
retrieved	3	1
not	2	4

Precision = 3 / (3 + 1) = 0.75

Recall = 3 / (3 + 2) = 0.6

P@5 and P@10

- ➔ P@N measures the precision at the top N results.
- ➔ These metrics assume that precision at the top results has the most impact on user experience, e.g. web search.
- ➔ Consider the top 10 results returned by two systems ([X] rel and [–] not rel),
  - ➔ System #1: [X], [–], [–], [X], [X], [X], [–], [X], [X], [X]
  - ➔ System #2: [X], [X], [X], [X], [–], [–], [–], [–], [X], [–]

  - ➔ System #1, P@5 = 0.6 and P@10 = 0.7
  - ➔ System #2, P@5 = 0.8 and P@10 = 0.5

Full Text Search in PostgreSQL

Based on PostgreSQL 9.4.6 Documentation, Chapter 12 - Full Text Search.
http://www.postgresql.org/docs/9.4/static/textsearch.html

Parsing Documents

- ➔ to_tsvector converts a text to a representation optimized for text search.
- ➔ to_tsvector([config regconfig,] document text) returns tsvector
  - ➔ `SELECT to_tsvector('english', 'the old lady sailing an old boat in the sea');`
    - ➔ `'boat':7 'ladi':3 'old':2,6 'sail':4 'sea':10`
  - ➔ `SELECT to_tsvector('portuguese', 'a velha raposa saltou o muro');`
    - ➔ `'mur':6 'rapos':3 'salt':4 'velh':2`

Parsing Queries

- to_tsquery and plainto_tsquery convert a query text to a query structure optimized for search.
- `to_tsquery([ config regconfig, ] querytext text) returns tsquery`
  - `SELECT plainto_tsquery('english','sail boats');`
    - `'sail' & 'boat'`
  - `SELECT plainto_tsquery('portuguese','o velho barco');`
    - `'velh' & 'barc'`

Full Text Search

- Full text searching in PostgreSQL is based on the match operator @@, which returns true if a tsvector (document) matches a tsquery (query).
  - `SELECT to_tsvector('portuguese','o velho barco') @@ plainto_tsquery('portuguese','barca');`
    - `t`
  - `SELECT to_tsvector('portuguese','o velho barco') @@ plainto_tsquery('portuguese','carro');`
    - `f`

Searching a Table

- ➔ Example table: people(id, name)
- ➔

```
CREATE TABLE people (
  id SERIAL PRIMARY KEY,
  name TEXT NOT NULL
);
```

- ➔

```
INSERT INTO people (name) VALUES ('Rui Silva');
INSERT INTO people (name) VALUES ('Pedro Silva');
INSERT INTO people (name) VALUES ('Filipa Silva');
INSERT INTO people (name) VALUES ('Ana Marques');
INSERT INTO people (name) VALUES ('Ana Pinto');
INSERT INTO people (name) VALUES ('Fernando Pinto');
INSERT INTO people (name) VALUES ('Alice Pinto e Pinto');
```

Searching a Table

- ➔
```
SELECT name
FROM people
WHERE to_tsvector('portuguese', name) @@ to_tsquery('portuguese',
'pinto');
```

- ➔
```
      name
----------------
Ana Pinto
Fernando Pinto
Alice Pinto e Pinto
```

- ➔ Although these queries will work without an index, most applications will find this approach too slow. Practical use of text searching usually requires creating an index.

Using Indexes

- ➔ There are two kinds of indexes that can be used to speed up full text searches.
  - ➔ GIN (Generalized Inverted Index) - lossless
  - ➔ GiST (Generalized Search Tree) - lossy
- ➔ As a rule of thumb, GIN indexes are best for static data because lookups are faster. For dynamic data, GiST indexes are faster to update.

Using GIN Index

- ➔ Create a GIN index on the name column.
- ➔ `CREATE INDEX people_name_idx ON people USING gin(to_tsvector('portuguese', name));`

Ranking Results

- ➔ PostgreSQL provides two predefined ranking functions, which take into account lexical, proximity, and structural information:
  - ➔ how often the query terms appear in the document;
  - ➔ how close together the terms are in the document;
  - ➔ how important is the part of the document where they occur.
- ➔ Different applications might require additional information for ranking, e.g., document modification time. The built-in ranking functions are only examples.

Ranking Example

➔
```
SELECT name,
  ts_rank_cd(
    to_tsvector('portuguese', name),
    to_tsquery('portuguese', 'pinto')
  ) AS score
FROM people
ORDER BY score DESC;
```

➔
```
        name         | score
---------------------+-------
Alice Pinto e Pinto |   0.2
Ana Pinto           |   0.1
Fernando Pinto      |   0.1
Rui Silva           |     0
Ana Marques         |     0
Pedro Silva         |     0
Filipa Silva        |     0
```

Obtaining Document Statistics

- The function ts_stat is useful for checking your configuration and for finding stop-word candidates.
- ```
  SELECT *
  FROM ts_stat('
    SELECT to_tsvector(name)
    FROM people
  ');
  ```
- ```
     word   | ndoc | nentry
  ----------+------+--------
   silva    |    3 |      3
   rui      |    1 |      1
   pinto    |    3 |      4
   pedro    |    1 |      1
   marqu    |    1 |      1
   filipa   |    1 |      1
   fernando |    1 |      1
   e        |    1 |      1
   ana      |    2 |      2
   alic     |    1 |      1
  ```

Search Systems

➔ Apache Lucene
https://lucene.apache.org

➔ Solr
https://solr.apache.org

➔ Elasticsearch
https://www.elastic.co/products/elasticsearch

➔ Terrier IR Platform
http://www.terrier.org

➔ Lemur Project
https://www.lemurproject.org

References

- ➔ Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. [online]
- ➔ W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, 2010. [online]
- ➔ Ricardo Baeza-Yates, and Berthier Ribeiro-Neto. *Modern Information Retrieval* (2nd Edition). ACM press, 2012.
- ➔ PostgreSQL. *PostgreSQL 13 Documentation, Chapter 12 - Full Text Search*. [online]