Information Retrieval

COMP90049
Knowledge Technologies

Lea Frermann and Justin Zobel and Karin Verspoor, CIS

Semester 2, 2019

THE UNIVERSITY OF MELBOURNE

Housekeeping

Assignment 1

- Online & ready!
- **small fix**: now with component words in dict.txt!
  (hopefully few) updates & fixes will be announced in LMS and lectures
- Report due: Fri, Sep 13 5pm
- Reviews / reflections due: Wed, Sep 18 5pm
- Questions: discussion board and **consultation hours**
  Monday Sep 2nd, 10am-11am, 803 Doug McDonell
  Monday Sep 9th, 4pm-5.30pm, 803 Doug McDonell

Housekeeping

Mid-semester Exam

- Fri, Aug 30 8.30 – 9.30
- Wilson Hall and Kwon Lee Dow
  assignment based on student ID now in LMS
- (Wilson Hall is cold!)
- also: examples, instructions...

Housekeeping

I'm looking for a course representative!

- attend student-staff meeting on Mon Sep 2, 12-1
- please email me if you're interested
- first come, first serve

Housekeeping

~~I'm looking for a course representative!~~

- ~~attend student-staff meeting on Mon Sep 2, 12-1~~
- ~~please email me if you're interested~~
- ~~first come, first serve~~

Zheng Wei Lim
limz2@student.unimelb.edu.au

Academic Writing

Structure of a research paper

- Introduction: brief summary of problem, main results, **research question**
- Related literature
- Dataset and methods
- Evaluation and Results and critical discussion
- General discussion
- Conclusions
- References

(Some) Characteristics of Academic writing

1. Logical argumentation throughout
   - Introduce problem and research question
   - Motivate your methods
   - Make clear how your experiments are relevant to the research question
   - Critically discuss your findings around the research question
   - Draw conclusions

(Some) Characteristics of Academic writing

2. Formal, objective language

In this paper I'll describe a couple of algorithms which seem to be useful for finding word blends in data.	In this paper, I will discuss three algorithms and their effectiveness in detecting word blend candidates, and their components in Twitter data.

(Some) Characteristics of Academic writing

3. Describe and **analyze** your results

As shown in Table 1, method A achieved m% precision, whereas method B achieved n% precision. Therefore, method A is more effective.

Conclusion

[...]

As shown in Table 1, method A achieved m% precision, whereas method B achieved n% precision. In terms of recall, however, [...] In order to gain a deeper understanding of the shortcomings of the respective methods, Figure 2 displays illustrative examples of mistakes. We observe that method A [...] while method B [...]. We therefore conclude that [...].

Academic Writing

(Some) Characteristics of Academic writing

Useful resources

→ Melbourne University's Academic Skills Unit

Roadmap

Last week: approximate string matching

- methods
- evaluation
- use cases

This week: information retrieval

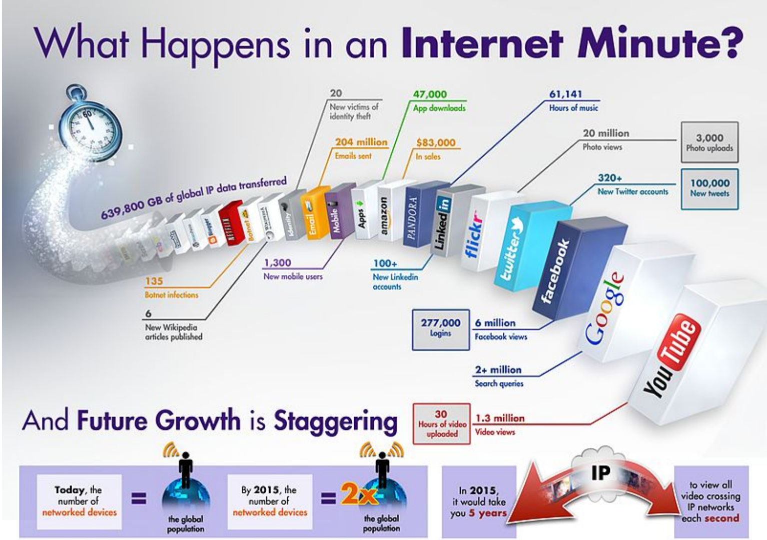
- string matching is (often) not enough – why?
- history and motivation
- methods
- evaluation

Defining “information retrieval”

“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, C. D., Raghavan, P., Schütze, H. (2008). Introduction to Information Retrieval

Defining “information retrieval”



https://commons.wikimedia.org/wiki/File:Internet_Minute_Infographic.jpg

Defining “information retrieval”

“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, C. D., Raghavan, P., Schütze, H. (2008). Introduction to Information Retrieval

Other fields (databases, file structures, . . .) deal with storage and retrieval in general.

What distinguishes IR from other areas of data processing?

Your thoughts on the difference between data retrieval and information retrieval?

Data retrieval versus information retrieval

Conventional database systems, such as relational systems, are designed for data retrieval:

- Prior to storage, the data is transformed into a representation suitable for manipulation by an algebraic query language.

Data retrieval versus information retrieval

Conventional database systems, such as relational systems, are designed for data retrieval:

- Prior to storage, the data is transformed into a representation suitable for manipulation by an algebraic query language.
  For example, the information that “enrolled student Jill Chambers was born on 15 Aug 1989” might be represented in a relational database by
  $\langle$`"Chambers", "Jill", "687651", 1989, 8, 15`$\rangle$

Data retrieval versus information retrieval

Conventional database systems, such as relational systems, are designed for data retrieval:

- Prior to storage, the data is transformed into a representation suitable for manipulation by an algebraic query language.
  For example, the information that “enrolled student Jill Chambers was born on 15 Aug 1989” might be represented in a relational database by
  $\langle$`"Chambers", "Jill", "687651", 1989, 8, 15`$\rangle$
- The information is unambiguous.

Data retrieval versus information retrieval

Conventional database systems, such as relational systems, are designed for data retrieval:

- Prior to storage, the data is transformed into a representation suitable for manipulation by an algebraic query language.
  For example, the information that “enrolled student Jill Chambers was born on 15 Aug 1989” might be represented in a relational database by
  ⟨"Chambers", "Jill", "687651", 1989, 8, 15⟩
- The information is unambiguous.
- Atypical information cannot be represented or queried unless it was anticipated at database-creation time.

Data retrieval versus information retrieval

Conventional database systems, such as relational systems, are designed for data retrieval:

- Prior to storage, the data is transformed into a representation suitable for manipulation by an algebraic query language.
  For example, the information that “enrolled student Jill Chambers was born on 15 Aug 1989” might be represented in a relational database by
  ⟨"Chambers", "Jill", "687651", 1989, 8, 15⟩
- The information is unambiguous.
- Atypical information cannot be represented or queried unless it was anticipated at database-creation time.
- Queries are represented in an algebraic language.
  `select * from Student where Surname = "Chambers"`

Data retrieval versus information retrieval

In IR systems:

- The stored documents are real-world objects that have been created for individual reasons. They do not have to have consistent format, wording, language, length, . . .

Data retrieval versus information retrieval

In IR systems:

- The stored documents are real-world objects that have been created for individual reasons. They do not have to have consistent format, wording, language, length, . . .
- The retrieval system is concerned with the document as originally created, not with a formal representation of the document (such as a list of keywords).

Data retrieval versus information retrieval

In IR systems:

- The stored documents are real-world objects that have been created for individual reasons. They do not have to have consistent format, wording, language, length, . . .
- The retrieval system is concerned with the document as originally created, not with a formal representation of the document (such as a list of keywords).
- Documents are rich and ambiguous, and there is no conceivable automatic method for translating them into an algebraic form.

Data retrieval versus information retrieval

In IR systems:

- The stored documents are real-world objects that have been created for individual reasons. They do not have to have consistent format, wording, language, length, . . .
- The retrieval system is concerned with the document as originally created, not with a formal representation of the document (such as a list of keywords).
- Documents are rich and ambiguous, and there is no conceivable automatic method for translating them into an algebraic form.
- Text in some kinds of collection has structured attributes, but these are only occasionally useful for searching. Examples include `<author>` tags and other metadata.

Data retrieval versus information retrieval

In IR systems:

- The stored documents are real-world objects that have been created for individual reasons. They do not have to have consistent format, wording, language, length, . . .
- The retrieval system is concerned with the document as originally created, not with a formal representation of the document (such as a list of keywords).
- Documents are rich and ambiguous, and there is no conceivable automatic method for translating them into an algebraic form.
- Text in some kinds of collection has structured attributes, but these are only occasionally useful for searching. Examples include `<author>` tags and other metadata.
- Users may not agree on the value of a particular document, even in relation to the same query.

(Knowledge Technology!)

Data retrieval versus information retrieval

Thus a data retrieval system is used to retrieve items based on **facts** that describe them. For example:

- “Get articles from The Age dated 11/8/2017.”
- “Fetch articles filed by Piotr Kulowsky in Kursograd.”
- “Get the article entitled ‘Alta Vista Searching for Success’.”

Data retrieval versus information retrieval

Thus a data retrieval system is used to retrieve items based on **facts** that describe them. For example:

- "Get articles from The Age dated 11/8/2017."
- "Fetch articles filed by Piotr Kulowsky in Kursograd."
- "Get the article entitled 'Alta Vista Searching for Success'."

An IR system is used to retrieve items based on their **meaning**.

- "Find articles that argue for better public transport in rural areas."
- "Is Bosnia a good holiday destination?"
- "Get articles about different kinds of dementia."

Data retrieval versus information retrieval

Thus a data retrieval system is used to retrieve items based on **facts** that describe them. For example:

- “Get articles from The Age dated 11/8/2017.”
- “Fetch articles filed by Piotr Kulowsky in Kursograd.”
- “Get the article entitled ‘Alta Vista Searching for Success’.”

An IR system is used to retrieve items based on their **meaning**.

- “Find articles that argue for better public transport in rural areas.”
- “Is Bosnia a good holiday destination?”
- “Get articles about different kinds of dementia.”

Or, more plausibly: “rural public transport”, “Bosnia holiday”, “dementia senility”.

Defining “information retrieval”

“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, C. D., Raghavan, P., Schütze, H. (2008). Introduction to Information Retrieval

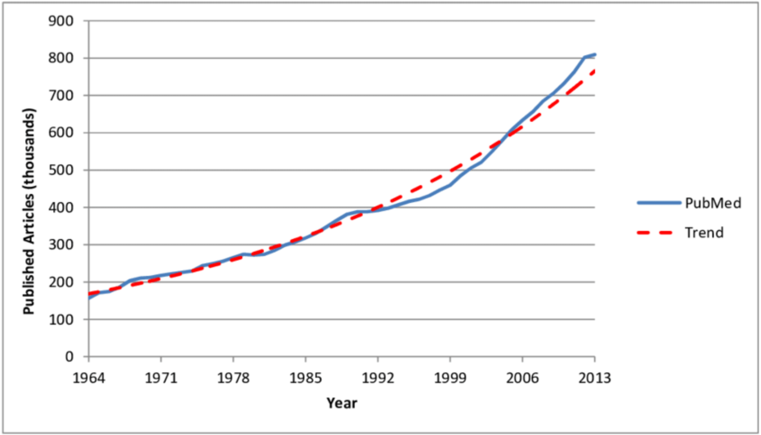
What distinguishes IR from other areas of data retrieval?

- There is an emphasis on the **user**: IR systems as mechanisms for finding documents that are of **value** to an individual.
- The **meaning or content** of a document is of more interest than the specific words used to express the meaning.

IR systems are arguably the primary means of access to stored information in our society.

Information Explosion (very selected example)

PubMed: Biomedical Literature Repository



Malakasiotis, Prodromos, et al. ”Biomedical Question-focused Multi-document Summarization: ILSP and AUEB at BioASQ3.” CLEF (Working Notes). 2015.

IR is everywhere

Search engines are a key part of the management of data such as:

- web sites
- legislation
- corporate documentation
- online retailers
- digital libraries

Google

- handles several thousand million queries a day
- when it was first successful, it was handling 10,000 queries a day
- growth of 8% per month

IR is everywhere

Applications like email management, personal document management

- IR systems are beginning to replace file systems
- traditional role of curator is being marginalized
- IR as a unifying technology, replacing a diversity of prior approaches

IR engines are ubiquitous

- close integration between the desktop and the web
- e.g., help systems mix on-computer with on-line information

IR is everywhere

Societal impact of data and information

- Search is political: data access is a human rights issue.
- Controll of data
- Censorship
- Fake news
- . . .

Text collections

Text collection	Size	
A single document	5 kB	0.05 MB
Complete text of Moby Dick	600 kB	0.6 MB
A researcher's papers – 10 years	10 MB	10 MB
An individual's email – 10 years	100 MB	100 MB
All the web pages at one small university	1 GB	1,000 MB
A single-purpose digital library	20 GB	20,000 MB
All books in a small university library	100 GB	100,000 MB
Govt web pages in English	1 TB	1,000,000 MB
US Library of Congress, 2012	20 TB	20,000,000 MB
Google, 2010	200 TB?	200,000,000 MB

Source for Library of Congress figures: `https://en.wikipedia.org/wiki/List_of_unusual_units_of_measurement#Data_volume`

Document Collections

Typical kinds of document collection include:

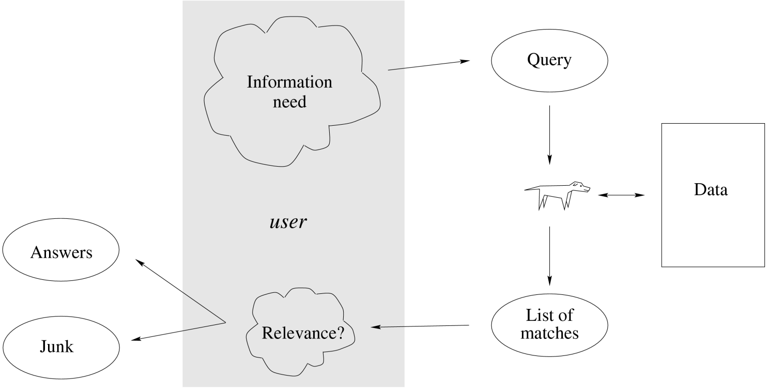
- web pages
- newspaper articles
- intranets
- academic publications
- company reports
- all documents on a PC
- parliamentary proceedings
- bibliographic entries
- historical records
- electronic mail
- court transcripts
- ...

Document Collections

Documents aren’t always text.

- rather: **messages**, i.e., objects that convey information from one person to another
- in the context of IR, “documents” include text, images, music, speech, handwriting, video, and genomes
- There are practical or prototype IR systems for content-based retrieval on each of these kinds of data

Information Seeking – Big picture



Information retrieval in a nutshell

- User has an information need
- User formulates a query
- IR engine retrieves a set of documents
- User evaluates the (ir)relevance of the documents

Information needs

The different kinds of IR system are linked by the concept of **information need**.

- IR system is used by someone because they have an information need they wish to resolve
- Information needs can be highly specific
- Information needs may be difficult to articulate or explain (to a human or a search system)

Information needs

The different kinds of IR system are linked by the concept of **information need**.

- IR system is used by someone because they have an information need they wish to resolve
- Information needs can be highly specific
- Information needs may be difficult to articulate or explain (to a human or a search system)

For example:

- When does the next train depart from Flinders St?
- What are the best travel destinations in Northumberland?
- Do I want to move to Adelaide?
- Are arguments for a space program mature or simplistic?

Many information needs cannot be described succinctly. Depends on who is asking

Searching

People search in a wide variety of ways. (well, somewhat...)

Google query lengths

- 1 word: 21.71%
- 2 words: 23.98%
- 3 words: 19.60%
- 4 words: 13.89%
- 5 words: 8.70%
- 6+ words: 12.12%

https://www.wordstream.com/blog/ws/2019/02/07/google-search-statistics

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.
- Follow links to specific documents.

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.
- Follow links to specific documents.
- Refine or modify the query.

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.
- Follow links to specific documents.
- Refine or modify the query.
- Maybe use advanced querying features.

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.
- Follow links to specific documents.
- Refine or modify the query.
- Maybe use advanced querying features.

Searching

People search in a wide variety of ways.

Perhaps the commonest mode is to:

- Issue an initial query.
- Scan a list of suggested answers.
- Follow links to specific documents.
- Refine or modify the query.
- Maybe use advanced querying features.

The purpose of many searches is to find a starting point for browsing.

Searching

Casual users generally use only the first page or so returned by their favorite search engine.

Professionals use a range of search strategies and are prepared to view hundreds of potential answers.

However, much the same IR techniques work for both kinds of searcher.

Searching ...

To resolve an information need using a search engine, a user **chooses words** and phrases that are **intended to match** appropriate documents, then use these words and phrases to **construct a query**.

If the query is unsuccessful, the user may reformulate it, thus many **different queries can represent the same information need.**

Searching ...

To resolve an information need using a search engine, a user **chooses words** and phrases that are **intended to match** appropriate documents, then use these words and phrases to **construct a query**.

If the query is unsuccessful, the user may reformulate it, thus many **different queries can represent the same information need.**

Consider the query “intel processor” under

- Web,
- News,
- Images,
- Shopping
- Scholar, ...

tabs provided by Google.

A different type of information need is meant in each case.

Searching ...

- Informational: `global warming`
- Factoid: `melting point of lead`
- Topic tracking: `Trump administration`
- Navigational: `university of melbourne`
- Transactional: `Macbook Air`
- Geospatial: `carlton restaurants`

Searching ...

- Informational: `new restaurants in carlton`
- Factoid: `new restaurants in carlton`
- Topic tracking: `new restaurants in carlton`
- Navigational: `new restaurants in carlton`
- Transactional: `new restaurants in carlton`
- Geospatial: `new restaurants in carlton`

Some web queries (Excite, 2001)

action bible
texas state government
interior design institute
reversi othello
ruben hurrican cater the book
toronto sun newspaper
sacramento apartments
the fairmont chateau whistler
forbed global the quiet american
four models of public relations
unlock mobile phone

centerfold galleries
excalibur 1981
free url redirection
lamborghini dioblo
april erikkson
cow hunter
drive pcmcia scsi
ball busting
brass insturments
algebra links
horrible news

Answers

An answer to a query could be defined as

- a document that matches the query according to formal criteria
- e.g., if it contains all the query words

Answers

An answer to a query could be defined as

- a document that matches the query according to formal criteria
- e.g., if it contains all the query words

- does this guarantee the answer to be **helpful** for an information need?
- what about **reliability**? E.g, inconsistent formatting of dates

Answers

An answer to a query could be defined as

- a document that matches the query according to formal criteria
- e.g., if it contains all the query words

- does this guarantee the answer to be **helpful** for an information need?
- what about **reliability**? E.g, inconsistent formatting of dates

The retrieved **answer** should contain **information** that the user is **seeking**.

That is, the document should be **relevant.**

Relevance

The **relevance** of a document to an information need cannot be determined computationally.

- The information need is knowledge held by the user, and is not written down.
- Identifying the topic of a document requires understanding of the text.
- The relevance may be implicit. For example, for the information need “will a US company take over BHP”, a document that states “Enron is bankrupt” is relevant, even though BHP is not mentioned.

Relevance

Relevance can be defined as: a document is relevant (that is, on the right topic) if it **contains knowledge** that helps the user to **resolve the information need**.

There are many other **kinds of relevance**: consider searches for a

- particular fact
- particular document
- particular individual or organization

Answers

Fundamentally, a **response** from a search engine is a **list of documents** of potential relevance.

Possible improvements:

- A **snippet**, which indicates which part(s) of the document is the basis of the answer. (This must be prepared on the fly, as it is specific to the query.)
- **Duplicates** are pruned, or aggregated into a single entry.
- A single **source** might only contribute a single answer.
- Answer types may be augmented with a map or other **infobox**.

Thought experiment

- How does a human judge the relevance of a document to a query?
- Can you describe an algorithm for retrieving documents that would capture the same basic process?

Approaches to retrieval

Consider the **criteria** that a **human** might use to judge whether a document should be returned in response to a query:

- Try and guess what the query might be inspired by, and what kind of information or document is being sought.
- Consider current news or events, or cultural perspectives, or their own experience with the query terms.
- Approach the task of looking through the documents with expectations of what a match is that is based on much more than the terms.
- Be ready to consider a document even if the terminology is completely different.

Approaches to retrieval

That is, a human would see the query as **representative of a topic**, and evaluate documents accordingly.

There is no computational way of approximating this process. Instead, we have to develop methods that use other forms of **evidence** to make a guess as to whether a document is **relevant**.

A simple solution, ‘grepping’

Imagine we wish to search through the texts of Project Gutenberg for **Pangolin**

Can simply use `grep`: linear scan over the text searching for a match (via a regular expression)

- Representation appropriate?
- How will this scale to large collections?

A simple solution, ‘grepping’

Imagine we wish to search through the texts of Project Gutenberg for **Pangolin**

Can simply use `grep`: linear scan over the text searching for a match (via a regular expression)

- Representation appropriate?
- How will this scale to large collections?
- What about handling more complex queries?
  - **Pangolin** AND **ant-eater**
  - **Pangolin** OR **ant-eater**
  - **Pangolin** NEAR **ant-eater**
  - **Pang*in**

Boolean Querying

Until about 1994, all retrieval systems used Boolean querying (and professional searchers) to identify matches.

A typical query might be

```
diabetes & risk & factor & NOT juvenile
```

Documents match if they contain the terms, and don't contain the `NOT` terms.

There is no ordering; matching is yes/no.

Boolean Querying

1. Indexing the Document Collection

→ binary term-document matrix

	doc1	doc2	doc3
juvenile	1	0	1
diabetes	1	1	0
risk	0	1	1
factor	0	1	1

Boolean Querying

1. Indexing the Document Collection

→ binary term-document matrix

documents, e.g., books, web pages, articles, ...

terms

	doc1	doc2	doc3
juvenile	1	0	1
diabetes	1	1	0
risk	0	1	1
factor	0	1	1

Boolean Querying

1. Indexing the Document Collection

→ binary term-document matrix

documents, e.g., books, web pages, articles, ...

terms

	doc1	doc2	doc3
juvenile	1	0	1
diabetes	1	1	0
risk	0	1	1
factor	0	1	1

Ex 1: Query: `diabetes` $\wedge$ `risk`

- Take the bit representations: `diabetes` = **110**   `risk` = **011**
- Perform bitwise AND ($\wedge$): **110** $\wedge$ **011** = **010**
- **010** means: 0 for doc1 (no), 1 for doc2 (yes), 0 for doc3 (no)
- Therefore document 2 is the only match

Boolean Querying

1. Indexing the Document Collection

→ binary term-document matrix

	doc1	doc2	doc3
juvenile	1	0	1
diabetes	1	1	0
risk	0	1	1
factor	0	1	1

terms

documents, e.g., books, web pages, articles, ...

Ex 2: Query: `diabetes` ∧ (NOT `juvenile`)

- Take the bit representations: `diabetes` = **110** `juvenile` = **101**
- Invert `juvenile`=**101** → NOT `juvenile`=**010**
- Perform bitwise AND (∧): **110** ∧ **010** = **010**
- **010** means: 0 for doc1 (no), 1 for doc2 (yes), 0 for doc3 (no)
- Therefore document 2 is the only match

Boolean Querying

1. Indexing the Document Collection

→ binary term-document matrix

documents, e.g., books, web pages, articles, ...

terms

	doc1	doc2	doc3
juvenile	1	0	1
diabetes	1	1	0
risk	0	1	1
factor	0	1	1

Supported operators:

- conjunction, use bitwise AND, $\wedge$
- disjunction, use bitwise OR, $\vee$
- negation, use bitwise complement, ˆ
- (*proximity operators*: `juvenile` /3 `factor` /p `juvenile`)

Boolean querying

Boolean querying is still the method of choice for **legal** and **biomedical** search:

- It is repeatable, auditable, and controllable.
- Boolean queries allow expression of complex concepts.

  ```
  (randomized & controlled & trial)
  or (clinical & study)
  ```

  biomedical queries: sometimes hundreds of terms in dozens of clauses.
- The **time investment** in developing precise queries (**months**) is perceived to be compensated for by reduction in time spent reading (also months).

Boolean querying

For general querying, Boolean querying is unsatisfactory:

- **strict** matching (spelling mistakes, synonyms, ...)
- no sensitivity to **frequency** (or weights)
- there is **no ranking** and no control over result set size
- it is remarkably difficult to do well.

How does ranking work?

Ranked retrieval: A **query** is matched to a **document** by looking for **evidence** in the document that it is on the **same topic** as the query (or the same topic as an **information need** that the query might represent).

How does ranking work?

Ranked retrieval: A **query** is matched to a **document** by looking for **evidence** in the document that it is on the **same topic** as the query (or the same topic as an **information need** that the query might represent).

There are several common terminologies for describing this:

- Is the query **similar** to the document?
- What is the **probability** that the document is relevant to the query?
- Are the document and query **on the same topic**?

For the commonest IR activity, text search, there are many kinds of **evidence of similarity**.

How does ranking work?

Ranked retrieval: A **query** is matched to a **document** by looking for **evidence** in the document that it is on the **same topic** as the query (or the same topic as an **information need** that the query might represent).

There are several common terminologies for describing this:

- Is the query **similar** to the document?
- What is the **probability** that the document is relevant to the query?
- Are the document and query **on the same topic**?

For the commonest IR activity, text search, there are many kinds of **evidence of similarity**.

The **rank** of a document is inversely proportional to its similarity (or probability of relevance)

Evidence of similarity

Some matches to the query “active south american volcano”:

Expedition Chile
. . . highest mountain in Chile and also the highest active volcano in the world, with active . . . We will only attempt this major South American peak . . .

Ray’s Volcano Zone
. . . and Central American Volcanoes Images of South American Volcanoes Images of South . . . Images, maps, movies of Sicilian active . . .

VolcanoWorld Monthly Contest
. . . October 1999. The last eruption of this South American volcano was . . . 1999. This is a North American stratovolcano . . . Also, an active fumarole

Volcanic Activity On The Rise In Central America
A volcano erupted near here, and another crater . . . officials in the two Central American countries said Thursday they had no . . .

Why might these documents have been ranked highly?

Evidence of similarity

Why might these documents have been ranked highly?

- choose documents with words in common with the query.
- some words are more significant than others:
  “volcano” might well find relevant documents by itself, but the query “south” is highly unlikely to do so.
- Significance can be estimated statistically, e.g.,: A word that is rare overall may be common in some documents.
- making effective use of such statistics is a core research activity in IR.

In each of the four matches, the word “volcano” is prominent – almost certainly this is the most significant word. In a collection of 45 gigabytes of web data:

word	active	south	american	volcano
occurrences	185,876	425,912	591,652	16,336

Evidence of similarity

Useful evidence beyond word-match

- documents with the query terms in the title.
- occurrences of the query terms as phrases ("active volcano" and "south america")
- Choose documents that were created recently.
- Attempt to translate between languages.
- Choose authoritative, reliable documents
- Choose pages with appropriately labelled incoming links.

Incorporating these concepts involves varying difficulty.

TF–IDF Models for information Retrieval

Term frequency – inverse document frequency (mostly recap)

Intuition:

- Less weight is given to a term that appears in many documents. (Inverse document frequency or IDF.)
- More weight is given to a document where a query term appears many times. (Term frequency or TF.)
- Less weight is given to a document that has many terms.

We want to favour terms that seem to be **discriminatory**, and reducing the impact of terms that seem to be randomly distributed.

Principles

- ad-hoc development of retrieval algorithms based on matching and counts is hard to justify
- better: **models** to unify observations, make predictions, provide direction, to abstract the essence of a problem, provide a framework
- basis of effective (modern) IR: documents and queries are made up of **indexed terms** and **tokens** (=occurrences)
- use a mathematical model as basis of a similarity measure

The vector-space model (quick recap)

Suppose there are n distinct indexed terms in the collection.

Each document $d = \langle w_{d,1}, w_{d,2}, \ldots, w_{d,t}, \ldots, w_{d,n} \rangle$

- where $w_{d,t}$ is a weight describing the importance of term t in d
- Most $w_{d,t}$ values will be zero, because most documents only contain a tiny proportion of a collection's terms (**sparsity**)

The vector-space model (quick recap)

For example:

- we have a document:

  *W*e few, we happy few, we band of brothers

- we have a dictionary of indexed terms:

  $$\langle \mathrm{a}, \mathrm{aardvark}, \ldots, \mathrm{band}, \ldots, \mathrm{brothers}, \ldots, \mathrm{few}, \ldots, \mathrm{happy}, \ldots \rangle$$

- document vector:

  $$d = \langle 0, 0, \ldots, 1, \ldots, 1, \ldots, 2, \ldots, 1, \ldots \rangle$$

The vector-space model

A vector locates a document (or, equivalently in this context, a query) as a point in *n*-space.

Documents with similar terms have points that are “nearby” in the space.

The vector-space model

In estimating topical similarity, the length of the vector is relatively unimportant.

Consequently, documents with a similar distribution of terms have similar angles in the space.

Typical problems:

- It isn't clear how to (best) choose the weighting function *w*
- Typical formulations of the vector space are orthogonal (Cartesian); there is much evidence that this is incorrect, but there are no clearly better alternatives

The vector-space model

Some **typical information** which might appear in a similarity calculation:

- $f_{d,t}$, the frequency of term t in document d.
- $f_{q,t}$, the frequency of term t in the query.
- f_t, the number of documents containing term t.
- N, the number of documents in the collection.
- n, the number of indexed terms in the collection.
- $F_t = \sum_d f_{d,t}$, the number of occurrences of t in the collection.
- $F = \sum_t F_t$, the number of tokens (occurrences) in the collection.

These statistics are sufficient for computation of the similarity functions underlying highly effective search engines.

The vector-space model

Some **typical information** which might appear in a similarity calculation:

- $f_{d,t}$, the frequency of term t in document d.
- $f_{q,t}$, the frequency of term t in the query.
- f_t, the number of documents containing term t.
- N, the number of documents in the collection.
- n, the number of indexed terms in the collection.
- $F_t = \sum_d f_{d,t}$, the number of occurrences of t in the collection.
- $F = \sum_t F_t$, the number of tokens (occurrences) in the collection.

Back to our heuristics, we wish to find documents d that satisfy:

- Terms t with low f_t, that is, are rare;
- Terms t with high $f_{d,t}$, that is, are common in the document;
- And $|d|$ is low, that is, the document is short.

The cosine measure

Task: score documents by relevance.

- find the cosine of the angle between the document and the query vector

$$sim(q, d) = \frac{\sum_i q_i d_i}{|q||d|}$$

- does this really capture **relevance?**
- Remember: our goal is to find the most relevant documents, not to formally solve the mathematical problems!
- → **Choose an appropriate model!**

The cosine measure

Task: score documents by relevance.

- find the cosine of the angle between the document and the query vector

$$sim(q, d) = \frac{\sum_i q_i d_i}{|q||d|}$$

why cosine?

- does this really capture **relevance?**
- Remember: our goal is to find the most relevant documents, not to formally solve the mathematical problems!
- → **Choose an appropriate model!**

The cosine measure

Many choices for a TF-IDF model are consistent with our heuristics!

For example,

$$\begin{array}{cl} \text{TF} & w_{d,t} = f_{d,t} \\ \text{IDF} & w_{q,t} = \dfrac{N}{f_t}, \quad \text{if } f_{q,t} > 0, \text{ otherwise } w_{q,t} = 0 \\ \text{Length} & |r| = \sqrt{\displaystyle\sum_t w_{r,t}^2} \end{array}$$

Cosine with this TF-IDF weighting model:

$$S(q,d) = \frac{\sum_t w_{q,t} \times w_{d,t}}{|q||d|}$$

The cosine measure

Many choices for a TF-IDF model are consistent with our heuristics!

For example,

TF	$w_{d,t} = 1 + \log_2 f_{d,t}$, if $f_{d,t} > 0$, otherwise $w_{d,t} = 0$
IDF	$w_{q,t} = \log_2 \dfrac{N}{f_t}$, if $f_{q,t} > 0$, otherwise $w_{q,t} = 0$
Length	$\|r\| = \sqrt{\sum_t w_{r,t}^2}$

Cosine with this TF-IDF weighting model:

$$S(q,d) = \frac{\sum_t w_{q,t} \times w_{d,t}}{|q||d|}$$

The cosine measure

Many choices for a TF-IDF model are consistent with our heuristics!

For example,

$$\text{TF} \times \text{IDF:} \qquad w_{d,t} = f_{d,t} \times \frac{N}{f_t}$$

$$\text{Query is binary:} \qquad w_{q,t} \in \{0, 1\}$$

$$\text{Length} \qquad |r| = \sqrt{\sum_t w_{r,t}^2}$$

“Cosine” with this TF-IDF weighting model:

$$S(q, d) = \frac{\sum_{t \in q} \text{TF-IDF}_{d,t}}{|d|}$$

Cosine Example

Term–document matrix (vector space model)

	doc1	doc2	doc3
juvenile	2	0	0
diabetes	1	2	0
risk	0	3	1
factor	0	1	2

Query: `diabetes risk`

TF: $w_{d,t} = f_{d,t}$

IDF: $w_{q,t} = \frac{N}{f_t}$, if $f_{q,t} > 0$, otherwise $w_{q,t} = 0$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

Cosine Example

Term–document matrix (vector space model)

	doc1	doc2	doc3
juvenile	2	0	0
diabetes	1	2	0
risk	0	3	1
factor	0	1	2

Query: `diabetes risk`

TF: $w_{d,t} = f_{d,t}$

IDF: $w_{q,t} = \frac{N}{f_t}$, if $f_{q,t} > 0$, otherwise $w_{q,t} = 0$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{q \cdot d}{|q||d|}$$

$$S(q,d_1) = \frac{\langle 0, \frac{3}{2}, \frac{3}{2}, 0\rangle \cdot \langle 2,1,0,0\rangle}{\sqrt{0^2 + \frac{3}{2}^2 + \frac{3}{2}^2 + 0^2}\sqrt{2^2 + 1^2 + 0^2 + 0^2}}$$

$$S(q,d_1) = \frac{1.5}{(2.12)(2.24)} \approx 0.316$$

Cosine Example

Term–document matrix (vector space model)

	doc1	**doc2**	doc3
juvenile	2	0	0
diabetes	1	2	0
risk	0	3	1
factor	0	1	2

Query: `diabetes risk`

TF: $w_{d,t} = f_{d,t}$

IDF: $w_{q,t} = \frac{N}{f_t}$, if $f_{q,t} > 0$, otherwise $w_{q,t} = 0$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{q \cdot d}{|q||d|}$$

$$S(q,d_2) = \frac{\langle 0, \frac{3}{2}, \frac{3}{2}, 0 \rangle \cdot \langle 0,2,3,1 \rangle}{\sqrt{0^2 + \frac{3}{2}^2 + \frac{3}{2}^2 + 0^2}\sqrt{0^2 + 2^2 + 3^2 + 1^2}}$$

$$S(q,d_2) = \frac{7.5}{(2.12)(3.74)} \approx 0.945$$

Cosine Example

Term–document matrix (vector space model)

	doc1	doc2	**doc3**
juvenile	2	0	0
diabetes	1	2	0
risk	0	3	1
factor	0	1	2

Query: `diabetes risk`

TF: $w_{d,t} = f_{d,t}$

IDF: $w_{q,t} = \frac{N}{f_t}$, if $f_{q,t} > 0$, otherwise $w_{q,t} = 0$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{q \cdot d}{|q||d|}$$

$$S(q,d_3) = \frac{\langle 0, \frac{3}{2}, \frac{3}{2}, 0\rangle \cdot \langle 0,0,1,2\rangle}{\sqrt{0^2 + \frac{3}{2}^2 + \frac{3}{2}^2 + 0^2}\sqrt{0^2+0^2+1^2+2^2}}$$

$$S(q,d_3) = \frac{7.5}{(2.12)(3.74)} \approx 0.316$$

Cosine Example

Term–document matrix (vector space model) — weighted by TF-IDF

	doc1	doc2	doc3
juvenile	$2 \times \frac{3}{1}$	0	0
diabetes	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$	0
risk	0	$3 \times \frac{3}{2}$	$1 \times \frac{3}{2}$
factor	0	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$

Query: `diabetes risk`

TF-IDF: $w_{d,t} = f_{d,t} \times \frac{N}{f_t}$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{\sum_{t \in q} w_{d,t}}{|d|}$$

$$S(q,d_1) = \frac{1 \times \frac{3}{2} + 0}{\sqrt{6^2 + 1.5^2 + 0^2 + 0^2}} \approx 0.242$$

Cosine Example

Term–document matrix (vector space model) — weighted by TF-IDF

	doc1	**doc2**	doc3
juvenile	$2 \times \frac{3}{1}$	0	0
diabetes	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$	0
risk	0	$3 \times \frac{3}{2}$	$1 \times \frac{3}{2}$
factor	0	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$

Query: `diabetes risk`

TF-IDF: $w_{d,t} = f_{d,t} \times \frac{N}{f_t}$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{\sum_{t \in q} w_{d,t}}{|d|}$$

$$S(q,d_2) = \frac{2 \times \frac{3}{2} + 3 \times \frac{3}{2}}{\sqrt{0^2 + 3^2 + 4.5^2 + 1.5^2}} \approx 1.86$$

Cosine Example

Term–document matrix (vector space model) — weighted by TF-IDF

	doc1	doc2	**doc3**
juvenile	$2 \times \frac{3}{1}$	0	0
diabetes	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$	0
risk	0	$3 \times \frac{3}{2}$	$1 \times \frac{3}{2}$
factor	0	$1 \times \frac{3}{2}$	$2 \times \frac{3}{2}$

Query: `diabetes risk`

TF-IDF: $w_{d,t} = f_{d,t} \times \frac{N}{f_t}$

$$|r| = \sqrt{\sum_t w_{r,t}^2}$$

$$S(q,d) = \frac{\sum_{t \in q} w_{d,t}}{|d|}$$

$$S(q,d_3) = \frac{0+1\times\frac{3}{2}}{\sqrt{0^2+0^2+1.5^2+3^2}} \approx 0.447$$

Evaluation Metrics

Recall evaluation in Approximate String Search:

- We have one (or more) probably misspelled token(s) of interest
- Our system returns one (or more) item(s) from the dictionary
- We examine whether the returned dictionary item(s) are “correct” (the intended word)

  → Accuracy
  → Precision
  → Recall

Evaluation Metrics

Evaluation in Information Retrieval:

- We have one (or more) queries
- Our system returns ~~one (or more)~~ **many** documents from the collection
- We examine whether the returned documents are relevant (meet the user's information need)

~~→ Accuracy~~
→ Precision
→ Recall

Evaluation Metrics

Some differences between evaluation in the two applications:

- Typically many more results in IR than Approx. Search

  → The collection is larger

- IR has multiple “correct” (relevant) results; Approx. Search only one

  → The collection is larger, and redundant
  → User’s need can potentially be met in many different ways
  → Accuracy isn’t meaningful

- IR results are ranked, Approx. Search typically not

  → Boolean querying typically more like Approx. Search evaluation
  → Approx. Search could be ranked, but typically many ties

Evaluation Metrics in an IR Context

Precision: $$\frac{\text{number of returned relevant results}}{\text{number of returned results}} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

Recall: $$\frac{\text{number of returned relevant results}}{\text{total number of relevant results}} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

(often useless in an IR context)

(suitably averaged across multiple queries)

Evaluation Metrics in an IR Context

Precision at k ($P@k$): $\frac{\text{number of returned relevant results in top k}}{\text{k}}$

(Recall at k usually not meaningful)

Evaluation Metrics in an IR Context

Average Precision: $\frac{1}{R}\sum_{k|d(k)\text{is relevant}} P@k$

where R is the total number of relevant documents for the query (denominator of Recall)

Typically averaged over many queries: MAP (Mean Average Precision)

Evaluation Metrics in an IR Context

...But where do the judgements come from???

The TREC experiments

US National Institute of Standarts and Technologies (**NIST**) established the Text REtrieval Conference (**TREC**) & framework

- to compare search engines in a systematic, unbiased way
- data set creation
- annually run since 1992!
- major reason for success of IR research and development

Data

- 1992: 2GB of newswire (huge back then!)
- 1990s: additional 50 queries evaluated each year
- today: ≈0.5TB (25,000,000 web pages)
- video, bioinformatics, differend languages, ...

Some limitations and extensions

query expansion: towards deeper text understanding

- query: “flight costs Italy”; Document: “[...] vacation [...] Italy [...] airplane fees [...]”
- option 1: reformulate query with a thesaurus (WordNet)
- option 2: *learn* a thesaurus from word co-occurrences in the document collection

relevance feedback: towards ‘mind-reading’ information need

- iteratively, improve the results
- maybe hard to initially formulate a good query... but easy to judge results given initial query (e.g., through clicks)
- IR model improves query representation based on user feedback

Rocchio Algorithm intuition: The optmal query will be most similar to *relevant documents* and least similar to *irrelevant documents*.

Summary

- Text search is a key computational technology.
- Search is much broader than the web and is used on vastly different scales. Specific search tasks require specific tools.
- Queries are distinct from information needs; the former are the written approximation of the latter. Search is one component, but not the only one, of the task of resolving an information need.
- Search can be Boolean or ranked. Boolean search is only appropriate for heavyweight applications such as deep exploration of a collection.
- Ranking involves assessment of evidence, including many features of documents but in particular term significance.
- There are many models for encapsulating evidence, including the TF-IDF weighting for the vector-space model.
- Measurement of effectiveness depends on the concept of relevance, and requires large-scale assessment of queries and documents.

Background Readings

Zobel, Justin and Alistair Moffatt (2006). “Inverted Files for Text Search Engines”. ACM Computing Surveys 38 (2): 1–56. doi:10.1145/1132956.1132959

Manning, Christopher D., Prabhakar Raghavan, Heinrich Schütze (2008). “Introduction to Information Retrieval”. Chapters 1, 6. Cambridge University Press.