Text Search

COMP90049 Knowledge Technologies

Justin Zobel and Rao Kotagiri, CIS

Semester 1, 2015

Defining “information retrieval”

Information retrieval (IR) is “the subfield of computer science that deals with storage and retrieval of documents” (Frakes & Baeza-Yates, 1992).

This definition emphasises *documents*. Other fields (databases, file structures, . . .) deal with storage, retrieval and also computation using the retrieved data in general.

What distinguishes IR from these other areas?

- There is an emphasis on the *user*. IR systems can be characterized 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.

Defining “information retrieval”

Typical kinds of document collection include web pages, newspaper articles, intranets, academic publications, company reports, all documents on a PC, research grant applications, parliamentary proceedings, bibliographic entries, historical records, electronic mail, and court transcripts.

Documents aren’t always text. They can be defined as *messages*: an object that conveys 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. E.g. Finding images that are similar to the query image.

Data retrieval versus information retrieval

Conventional databases 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 Jack Chambers was born in 1990 on March 17th” might be represented in a relational database by
  $\langle$`"Chambers", "Jack", "287651", 1990, 3, 17`$\rangle$
- 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 ATTRIBUTES from RELATIONS where CONDITIONS
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, . . .
- 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).
- Users may not agree on the value of a particular document, even in relation to the same query.
- Documents are rich and ambiguous, and there is no conceivable automatic method for translating them into an algebraic form. That is representing the meaning of the document with unique meaning.
- Text in some kinds of collection has structured attributes, but these are only occasionally useful for searching. Examples include `<author>` tags and other metadata. Examples are XML documents.

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 15/7/2002."
- "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."
- "Is Bosnia a good holiday destination?"
- "Get articles about different kinds of dementia."

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

The past

The study of IR originated in schools of librarianship, during the nineteenth century.

(Special-purpose mechanisms for IR were in existence much earlier, such as the hand-crafted concordances that were used to index the Bible in the 1200s.)

The Dewey Decimal system (1876) can be seen, in retrospect, as a contribution to information retrieval. Many of the achievements in the area were practical developments in management of documents: card indexes, loan systems, subject categories, personnel files, . . .

Dewey Decimal system

Categories of Dewey Decimal system

- 000 General works, Computer science and Information
- 100 Philosophy and psychology
- 200 Religion
- 300 Social sciences
- 400 Language
- 500 Science
- 600 Technology
- 700 Arts & recreation
- 800 Literature
- 900 History & geography

Dewey Decimal system

Sub-categories for Natural Science in the Dewey Decimal system

- 500 Natural sciences and mathematics
- 510 Mathematics
- 516 Geometry
- 516.3 Analytic geometries
- 516.37 Metric differential geometries
- 516.375 Finsler Geometry

Before the 1990s, the most widely-used information retrieval “systems” were clerks and librarians: that is, trained searchers who could expertly match an interest to a book, article, or file.

The recent past

Investigation of computerized IR began during the 1950s, and grew slowly until about 1990, driven by the accumulation of information in electronic form: medical files, corporate repositories, large-scale bibliographies.

In 1980, a 50 megabyte (size of a "washing-machine") hard drive cost was around $50,000 (median 4-bedroom house price in Melbourne was $50,000 in 1980 and 2012 it was $567,500!).

A single search using a commercial IR system cost $20–$100.

Typical online collections in 1990 were tens to hundreds of megabytes.

The documents were plain text, marked up in languages developed for libraries. Typically they were quality-controlled to ensure that meta-data was correct and fields were filled in.

Search was typically over abstracts and manually-allocated keywords, rather than full documents.

The recent past

A document search system was implemented on cards and tape in 1953–1957. It was less effective than manual search.

Workable prototype IR systems such as SMART, forerunners of current search engines, appeared from about 1962 – despite the tiny storage capacities of computers at that time.

Modern systems began to appear around 1990, for the large volumes of data that CD-ROM, high-density disks, and high-bandwidth networks made possible.

(In 1990, a 400 megabyte "shoebox" disk cost around $10,000. A 300 megabyte magneto-optical rewritable disk cost $300, but the drive cost $30,000, and disks could not be left in the drive.)

Most text collections had been carefully curated, such as legislation or bibliographic data, but text collections were becoming increasingly anarchic: Usenet news archives, emails, and ad hoc repositories of research papers.

The present

Increasing disk capacities created the opportunity for personal-computer document repositories, such as online help systems and drug databases.

The spread of the internet created the opportunity for search on remote computers, initially via services such as interlibrary data exchanges, ARCHIE, and WAIS.

Search engines based on IR techniques are a key computing technology. Search via web search engines is the primary access method used to find pages, even familiar pages – they are widely preferred to bookmarks.

The web was originally an adjunct to other mechanisms for disseminating information: URLs were distributed by word-of-mouth, email, browsing, or Usenet news. The web took off with the release of Mosaic in 1993 (University of Illinois). Lycos (Carnegie Mellon University) launched in 1994 with an index of 54,000 web pages.

The first truly successful web search engine, Alta Vista (Yahoo), was developed to advertise the power of the underlying hardware.

The present

Search engines are a key part of the management of data such as web sites, legislation, corporate documentation, online retailers, digital libraries, and intelligence services.

In some applications – email management, personal document management – IR systems are beginning to replace file systems, and the traditional role of curator is being marginalized. Thus IR is an example of a unifying technology that is replacing a diversity of prior approaches.

Search engines are used to search over a wide range of scales of data.

They are ubiquitous, with close integration between the desktop and the web – for example, help systems mix on-computer with on-line information.

Search is political: data access is a human rights issue.

Google now processes over 4.58 billion searches per day (40,000 search queries every second, 1.7 trillion searches per year worldwide). It has grown by 14% per month! Based on Googles report a typical web search would, on average, produce around 0.2g of CO_2, compared with the 7g of CO_2 generated by a boiling a kettle.

Text collections

Collection	Size	
A single document	5 kB	5 kB
Complete text of *Moby Dick*	600 kB	600 kB
A researcher's papers – 10 years	10 MB	10,000 kB
An individual's email – 10 years	100 MB	100,000 kB
All the web pages at one small university	1 GB	1,000,000 kB
A single-purpose digital library	20 GB	20,000,000 kB
All books in a small university library	100 GB	100,000,000 kB
Govt web pages in English	1 TB	1,000,000,000 kB
Google, June 2004	20 TB	20,000,000,000 kB
# Web pages	19 March, 2015	4.58 billion pages

Sizes

Prifix	10^n	Size
Kilo	2^{10}	10^3
Mega	2^{20}	10^6
Giga	2^{30}	10^9
Terra	2^{40}	10^{12}
Peta	2^{50}	10^{15}
Exa	2^{60}	10^{18}
Zetta	2^{70}	10^{21}
Yotta	2^{80}	10^{24}

Information needs

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

An IR system is used by someone because they have an information need they wish to resolve. Information needs can be highly specific, but 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. For example, whether a travel destination is interesting depends on who is asking – some people like nightlife, other people like wildlife.

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.
- Use advanced querying features.

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

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.

Consider the query “intel processor” under the web, news, groups, images, video, shopping, and scholar tabs provided by Google. A different information need is meant in each case.

There are also different query intents in each category.

- Requests for information: “global warming”
- Factoid questions: “what is the melting point of lead?”
- Topic tracking: “what is the history of this news story?”
- Navigational: “University of Melbourne”
- Service or transaction: “Mac powerbook”
- Geospatial: “Carlton restaurant”

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: if it contains all the query words, for example, then it could be described as a match.

But this does not mean that the document is a helpful response for that particular information need.

Moreover, such matching criteria are likely to be simplistic and unreliable. For example, documents often contain information such as a title or date, but not in a consistent way, and such content is often not helpful for retrieval.

What is required is that the document should contain information that the user is seeking.

That is, the document should be *relevant*.

Answers

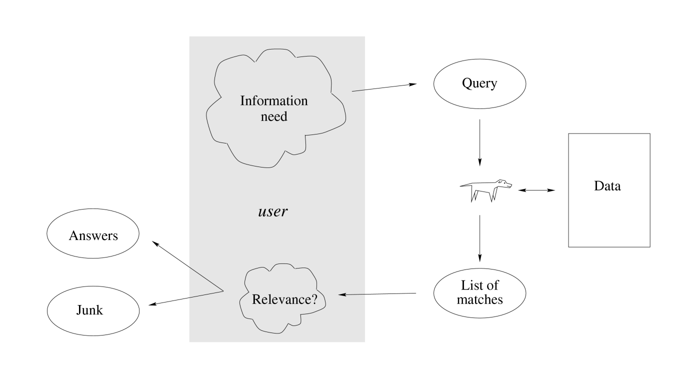
The relevance of a document to an information need cannot be determined computationally.

- 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 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, or a particular document, or a particular individual or organization.

Answers


Information need
user
Query
Data
Answers
Junk
Relevance?
List of matches

Answers

Superficially, we perceive the response from a search engine as a list of web pages of potential relevance.

But, to begin with, they are not web pages, they are descriptions of web pages. In particular, each description includes a *snippet* which must be prepared on the fly, as it is specific to the particular query.

Duplicates are pruned, or aggregated into a single entry.

Only a limited number of answers is shown from each web site.

Answer types may be augmented with a map or other infobox.

Approaches to retrieval

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

They would:

- 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.

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.

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 grey; matching is yes/no.

Such matching can be made more sophisticated by use of mechanisms such as fields (title, abstract) and proximity.

(In the 1980s, if you, as an academic, wanted to find a research paper not held at your institution, the 'search interface' was the librarian who listened to your need and used it to assemble a Boolean query, which was used to produce a printout that you looked through. When the paper finally arrived in your pigeonhole, it would have a receipt indicating how much the search, copying, postage and labour had amounted to – typically $30 to $100.)

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)
  ```

  It is common for biomedical queries to contain 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).

For general querying, Boolean querying is unsatisfactory in several respects: there is no ranking and no control over result set size, and it is difficult to incorporate factors such as monotonicity. And it is remarkably difficult to do well.

How does ranking work?

In principle, the idea of *ranked retrieval* is simple. 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*?

The more similar or likely a page is, relative to the other documents in the collection, the higher its *rank*.

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

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 . . .

Evidence of similarity

Why might these pages have been ranked highly?

- Choose documents with words in common with the query.

This is obvious, but some words are more significant than others. The query “volcano” might well find relevant documents by itself, but the query “south” is highly unlikely to do so.

Significance can be estimated statistically. Investigation of methods for 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

A strategy based on words-in-the-document is a contrast to the kinds of retrieval used until recently.

In a published book held in a library, an expert has assigned a subject, other search terms (or index terms), and a Dewey Decimal number.

A book can be searched for on the basis of title, author, or search terms. It can be found by browsing under the Dewey Decimal ordering.

The content – that is, the actual words used in writing the book – is not indexed at all. Thus, if a use of a book is not anticipated at filing time, it cannot be found at search time.

E.g., A book with the tittle:

Finite State Machines is

wrongly filed as a mechanical engineering book when it belongs to the area of theory of computer science and electronic engineering.

Evidence of similarity

Evidence in addition to word-match is being used to select documents.

- Choose documents with the query terms in the title or URL.
- Look for occurrences of the query terms as phrases.

For example, the first match contains “active volcano” and “south america”.

- Choose pages that were created recently.
- Attempt to translate between languages.
- Choose authoritative pages with large numbers of incoming links.
- Choose pages with appropriately labelled incoming links.

Evidence of similarity

- Use synonyms or related terms.

For example, a “volcano” document that additionally mentions “lava” is more likely to be relevant than one that doesn’t.

Such additional terms can be found by inspecting matching documents, using processes called query expansion and relevance feedback.

- *Query expansion*: fetch similar documents, find the terms these have in common, add these to the query.
- *Relevance feedback*: ask the user which documents were right, then proceed as for query expansion.

These mechanisms help some queries but ruin others.

Additional terms can also be found, but even less reliably, in a thesaurus or a resource such as WORDNET.

Evidence of similarity

Query: “extra terrestrial life/intelligence”

Expansion terms derived from a corpus:
extraterrestrials, planetary society, universe, civilization, planet, radio signal, seti, sagan, search, earth, extraterrestrial intelligence, alien, astronomer, star, radio receiver, nasa, earthlings, e.t., galaxy, life, intelligence, meta receiver, radio search, discovery, northern hemisphere, national aeronautics, jet propulsion laboratory, soup, space, radio frequency, radio wave, klein, receiver, comet, steven spielberg, telescope, scientist, signal, mars, moises bermudez, extra terrestrial, harvard university, water hole, space administration, message, creature, astronomer carl sagan, intelligent life, meta ii, radioastronomy, meta project, cosmos, argentina, trillions, raul colomb, ufos, meta, evidence, ames research center, california institute, history, hydrogen atom, columbus discovery, hypothesis, third kind, institute, mop, chance, film, signs

Monotonicity

Effective similarity measures combine information about queries and documents so that three monotonicity observations are enforced:

- Less weight is given to terms that appear in many documents. (Inverse document frequency or IDF.)
- More weight is given to terms that appear many times in a document. (In-document frequency or TF.)
- Less weight is given to documents that have many terms.

Global frequency:

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

$weight(volcano) > weight(active) > weight(south) > weight(american)$

The intention is to bias the score towards relevant documents by favouring terms that seem to be discriminatory, and reducing the impact of terms that seem to be randomly distributed.

Principles

The observation that word matching and word counts can be used to find answers provides a basis for ad hoc development of retrieval algorithms, but such a piecemeal approach is hard to justify as research.

Models are used throughout science to unify observations, make predictions, and provide direction. IR is no exception. Models of retrieval are used to abstract the essence of a problem, to reason about a problem, and to provide a framework for application of existing solutions we can use tools such as logic and statistics.

The basis of the effective IR models in use today is that documents and queries are made up of terms or tokens. (In early IR these might have been manually assigned index terms. In web IR they could include many things in addition to full document content.)

A mathematical model can then be used as the basis of a similarity measure.

The vector-space model

One of the earliest models proposed for IR (in 1962) was the vector-space model.

Suppose there are n distinct indexed terms in the collection. Then each document d can be thought of as a vector

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

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.)

Intuitively, if some other document d' has a vector

$$\langle w_{d',1}, w_{d',2}, \ldots, w_{d',t}, \ldots, w_{d',n} \rangle$$

where the weights are close to those of d – in particular, if the non-zero w values are for much the same set of terms – then d and d' are likely to be similar in topic.

The vector-space model

A vector locates a document (or, equivalently in this context, a query) as a point in *n*-space. The angle of the vector is determined by the relative weight of each term.

Something that is not easy to derive from the principles of the vector space model is how to compute $w_{d,t}$ values. There have been many proposals of alternatives, some rather arbitrary.

In the vector space model, the space is orthogonal (Cartesian) – that is, the terms are treated as if they occur independently.

- This is obviously false. Some words (say “Saudi”) positively influence the likelihood of observing others (“Arabia”, “prince”, “oil”).
- Yet there has been no (clear) successful use of dependency models in IR, under the vector space model or the alternatives.

The vector-space model

Most similarity formulations require values such as:

- $f_{d,t}$, 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 occurrences in the collection.

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

To link back to monotonicity: we wish to find documents d that have

- Term t with low f_t, that is, are rare;
- But t has high $f_{d,t}$, that is, is common in the document;
- And $\sum_{w \in d} f_{d,w}$ is low, that is, the document is short.

The cosine measure

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

An alternative is to measure the angle between vectors. A typical older formulation that works well in practice calculates the cosine of the angle in n-dimensional space between a query vector $\langle w_{q,t} \rangle$ and a document vector $\langle w_{d,t} \rangle$.

$$w_{q,t} = \ln\left(1 + N/f_t\right) \qquad w_{d,t} = 1 + \ln f_{d,t}$$

$$W_d = \sqrt{\sum_t w_{d,t}^2} \qquad W_q = \sqrt{\sum_t w_{q,t}^2}$$

$$S(q,d) = \left(\sum_{t \in q \wedge d} w_{d,t} \cdot w_{q,t}\right) / \left(W_d \cdot W_q\right)$$

Note that documents can score highly even if some query terms are not present.

Information theory

Consider a message M composed of distinct symbols $w_1, \ldots, w_n$, where each symbol w_i has a frequency f_i. The total length of the message is $|M| = \sum_i f_i$.

Information theory tells us that the minimum length encoding of the message is to allocate $-\log_2 \frac{f_i}{|M|}$ bits to symbol w_i.

That is, common symbols (high f_i) get a small number of bits and rare symbols get a large number of bits. The sum

$$E = \sum_i -f_i \times \log_2 \frac{f_i}{|M|}$$

is the *entropy* of the message; this is the theoretical minimum length of the message in the context of the provided information.

Relationship to information retrieval: we are interested in terms that have high entropy, and documents in which these terms are a significant component of the document's 'message'.

Language models

Another approach is to estimate the entropy of the query against the document, or equivalently to see which document most closely models the distribution of the (tiny number of) terms in the query.

$$S(q,d) = \prod_{t \in q} \left(\frac{|d|}{|d| + \mu} \cdot \frac{f_{d,t}}{|d|} + \frac{\mu}{|d| + \mu} \cdot \frac{F_t}{F} \right)$$

$$\overset{rank}{=} \sum_{t \in q} \log \left(\frac{f_{d,t}}{|d| + \mu} + \frac{\mu}{|d| + \mu} \cdot \frac{F_t}{F} \right)$$

The per-term information is an estimate of the term's likelihood in the document, smoothed with information from the collection as a whole. The value μ is a tuning constant specifying how two models (document and background) are mixed.

This approach seems very different to the others – there is no explicit weighting for terms that are rare in the collection, for example.

Term-oriented language models

The language-model formulation can be transformed to a rank-equivalent version. Observe that

$$S(q,d) = \prod_{t \in q \wedge d} \left(\frac{f_{d,t}}{|d| + \mu} + \frac{\mu}{|d| + \mu} \cdot \frac{F_t}{F} \right) \times \prod_{t \in q - d} \left(\frac{\mu}{|d| + \mu} \cdot \frac{F_t}{F} \right),$$

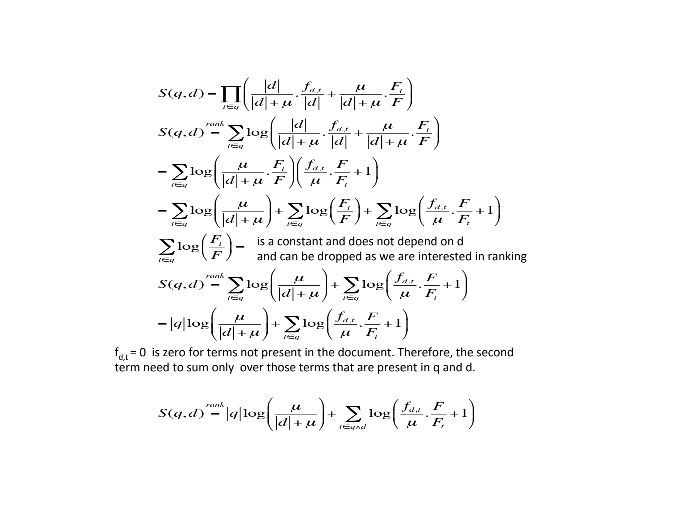
and $\prod_{t \in q} \frac{\mu}{|d|+\mu} \cdot \frac{F_t}{F} \times \prod_{t \in q} \frac{|d|+\mu}{\mu} \cdot \frac{F}{F_t} = 1$ and $\prod_{t \in q} \frac{F_t}{F}$ is constant for a query. Then

$$S(q,d) \overset{rank}{=} |q| \log \frac{\mu}{|d| + \mu} + \sum_{t \in q \wedge d} \log \left(\frac{f_{d,t}}{\mu} \cdot \frac{F}{F_t} + 1 \right)$$

This version has a document weight ($|q| \log \frac{\mu}{|d|+\mu}$), in-document frequency ($\frac{f_{d,t}}{\mu}$), and inverse document frequency ($\frac{F}{F_t}$).

All the weights can be directly derived from information theory.

Answers



$$S(q,d) = \prod_{t\in q}\left(\frac{|d|}{|d|+\mu}\cdot\frac{f_{d,t}}{|d|} + \frac{\mu}{|d|+\mu}\cdot\frac{F_t}{F}\right)$$

$$S(q,d) \overset{rank}{=} \sum_{t\in q}\log\left(\frac{|d|}{|d|+\mu}\cdot\frac{f_{d,t}}{|d|} + \frac{\mu}{|d|+\mu}\cdot\frac{F_t}{F}\right)$$

$$= \sum_{t\in q}\log\left(\frac{\mu}{|d|+\mu}\cdot\frac{F_t}{F}\right)\left(\frac{f_{d,t}}{\mu}\cdot\frac{F}{F_t}+1\right)$$

$$= \sum_{t\in q}\log\left(\frac{\mu}{|d|+\mu}\right) + \sum_{t\in q}\log\left(\frac{F_t}{F}\right) + \sum_{t\in q}\log\left(\frac{f_{d,t}}{\mu}\cdot\frac{F}{F_t}+1\right)$$

$\sum_{t\in q}\log\left(\frac{F_t}{F}\right) =$ is a constant and does not depend on d and can be dropped as we are interested in ranking

$$S(q,d) \overset{rank}{=} \sum_{t\in q}\log\left(\frac{\mu}{|d|+\mu}\right) + \sum_{t\in q}\log\left(\frac{f_{d,t}}{\mu}\cdot\frac{F}{F_t}+1\right)$$

$$= |q|\log\left(\frac{\mu}{|d|+\mu}\right) + \sum_{t\in q}\log\left(\frac{f_{d,t}}{\mu}\cdot\frac{F}{F_t}+1\right)$$

$f_{d,t} = 0$ is zero for terms not present in the document. Therefore, the second term need to sum only over those terms that are present in q and d.

$$S(q,d) \overset{rank}{=} |q|\log\left(\frac{\mu}{|d|+\mu}\right) + \sum_{t\in q\wedge d}\log\left(\frac{f_{d,t}}{\mu}\cdot\frac{F}{F_t}+1\right)$$

The TREC experiments

NIST (National Institute of Standards and Technology) established the large-scale TREC framework in 1992 to compare search engines in a systematic, unbiased way.

The first year of TREC used two gigabytes of newswire – a huge volume of data for its day. (Two gigabytes of disk might have cost around $20,000.)

Throughout the 1990s, an additional 50 queries were evaluated each year. Most of the document collections were re-used over several years.

The largest current collection is half a terabyte (25,000,000 web pages). About 100 groups participate each year.

Tasks have included video and bioinformatic retrieval as well as different languages and different aspects of text retrieval (named pages, home pages, topic coverage).

The TREC experiments

- Define *relevance* carefully (topic search, named-page search, multi-aspect search . . .)
- Identify a set of systems that are to be compared.
- Given a set of queries, use *pooling* to find a set of interesting pages from a collection. In pooling, each system returns its top *k* answers for each query, which are then combined into per-query pools.
- Assess the documents in each pool for relevance – if the pool is large, it is reasonable (most of the time) to assume that documents outside the pool are irrelevant.
- Compare the ability of engines to find these pages.

The TREC experiments

In a typical year, 1998,

- The document pools were (a) 2 gigabytes of newswire-type data, or about 0.5 million documents, and (b) 100 gigabytes of web data (massive at the time), or about 7 million documents.
- On the newswire data there was 50 queries.
- On the newswire data, about 30 groups participated with 61 systems, each reporting the top 1000 documents for each query.
- The top 100 answers for each system were pooled, giving about 3,000 documents per query or 150,000 documents overall.
- Humans assessed each of the 150,000 documents for relevance to the queries, finding an average of about 70 relevant documents per query.

The TREC experiments

The appearance of effective web-scale search systems would have been delayed without the evaluation framework given by a large volume of shared and robust test data, and by the opportunity it provided to share knowledge about system implementation.

In a typical year around 100 groups participate with hundreds of systems, each exploring new avenues towards improving retrieval.

There are now several other “TRECs”, including TRECVID for video, TREC Legal, TREC Biomedical, INEX for XML documents, CLEF for cross-language information retrieval, TDT for topic detection and tracking, and the Japanese NTCIR for Asian languages.

Summary

- Text search is a key computational technology. But it has a long pre-computational history.
- 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 expression 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. The key concept is of monotonicity.
- There are many models for encapsulating evidence, including the vector-space model and language models.
- Measurement of effectiveness depends on the concept of relevance, and requires large-scale assessment of queries and documents.

Reading: Manning et al., chapters 6, 11, & 12.