Information Retrieval I

David Hawking

30 Sep 2010

Machine Learning Summer School, ANU
Session Outline

- Ranking documents in response to a query
- Measuring the quality of such rankings
- Case Study: Tuning 40 parameters at the London School of Economics
- Coffee Break
- Web Search Engineering
- Field Work: how do Web search engines really work?
- Stretch Break
- Discussion: Other IR problems for machine learning
- Historical context
Start a Machine Learning Run to discuss later
Information Retrieval

- Information Need
- IRS
- Documents
- Query
- Results

- Ranked retrieval → ranked list of results
Measuring/comparing the quality of rankings
Mean average precision (MAP) = area under curve.
Normalised Discounted Cumulative Gain

|                   | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 20 |
| Perfect System    | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | - |
| Real System A     | 1 | 2 | 3 | 4 | 5 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Real System B     | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 5 | 4 | 3 | 2 | 1 | - |

(Relevance judged on a 5 point scale)

\[
DCG[r] = \begin{cases} 
G[1] & \text{if } r = 1 \\
DCG[r - 1] + G[r]/ \log_b r & \text{otherwise}
\end{cases}
\]
But where do the utility judgments come from?

(We’ll return to this later on.)
Probability Ranking Principle

Maron & Kuhns, JACM, 1960

"... technique called “Probabilistic Indexing”, allows a computing machine, given a request for information, to derive a number (called the “relevance number”) for each document, which is a measure of the probability that the document will satisfy the given request. The result of the search is an ordered list of those documents which satisfy the request ranked according to their probable relevance."

- Cooper (1977) produced a counter example, based on sub-classes of users with different criteria submitting the same query ⇒ need to model diversity.
Modern Ranking Functions

\[ RSV = \alpha_0 D_0 + \ldots + \alpha_n D_n + \beta_0 S_0 + \ldots + \beta_n S_n \] (1)

- Machine learned combination of:
  - dynamic scores – probability of relevance given doc and query text
  - static priors, independent of the query
Dynamic factors
Key Concepts

▶ **Term** — Basic unit of indexing: e.g. a word, a word-stem, a phrase. Could be any discrete feature, not necessarily derived from text.

▶ **Term Coordination.**

▶ **tf** — Term frequency.

▶ **N** — Number of documents in the collection.

▶ **V** — Vocab – distinct terms in the collection.

▶ **n_i** — Number of documents with i-th term present.

▶ **idf** — Inverse document frequency. Spärck Jones, J Doc, 1972: \( \lceil \log_2 N \rceil - \lceil \log_2 n_i \rceil + 1. \)

▶ **Relevance** — Often modelled as dichotomous variable.

\( Rel | \overline{Rel} \)
Probabilistic Retrieval

(From Robertson and Zaragoza tutorial, SIGIR 2007) Starting with the probability ranking principle:

\[ P(\text{Rel}|d, q) \propto q \frac{P(\text{Rel}|d, q)}{P(\text{Rel}|d, q)} \frac{P(d|\text{Rel}, q)}{P(d|\text{Rel}, q)} \]

transform to odds (2)

\[ \propto q \frac{P(d|\text{Rel}, q)}{P(d|\text{Rel}, q)} \]

Bayes rule (3)

\[ \approx \prod \frac{P(tf_i|\text{Rel}, q)}{P(tf_i|\text{Rel}, q)} \]

Assume independence (4)

\[ \approx \prod_{t \in q} \frac{P(tf_i|\text{Rel}, q)}{P(tf_i|\text{Rel}, q)} \]

Restrict to query terms (5)

\[ \propto q \sum_{t \in q} \log \frac{P(tf_i|\text{Rel}, q)}{P(tf_i|\text{Rel}, q)} \]

So we can add weights (6)
Okapi BM25 (Robertson et al, 1994)

\[ w_t = tf_d \times \frac{\log(\frac{N-n+0.5}{n+0.5})}{k_1 \times ((1-b) + b \times \frac{dl}{avdl}) + tf_d} \]  

\[ S_d = \sum_{t \in q} w_t \]  

- \( S_d \) is not a probability but should be rank-equivalent to it.
Modelling saturation is important.

\[
\frac{tf}{tf + k}
\]  

(9)
Length normalisation

Need for normalisation of $tf_i$ depends upon why some documents are longer than others. Make it tunable:

$$tf_i' = \frac{tf_i}{B} \quad (10)$$

$$B = (1 - b) + b \frac{dl}{dl} \quad (11)$$
BM25F - Extension to fields

- Weight term frequencies prior to non-linear combination in BM25.
- Robertson, Zaragoza & Taylor, CIKM 2004
Other Retrieval Models

- Vector Space
- Language Models
- Divergence from Randomness (parameter free!)
Using an inverted file to generate dynamic scores

Postings (uncompressed).
(2,3)(7,1)(11,2)(17,1)(22,6)

Term Dictionary

Index

Document Table

Term  count  postings
aaaaa  1
oboe   5
oblong 3
zzzzz  2

Score  DocID  Length  QIE  Snippet
0.145  doc001 2106  0.6
0.212  doc002 5327  0.735  Arist...
0.707  doc003 4108  0.33
0.009  doc004 2999  0.1
0.031  doc005 101  0.2
0.100  doc006 27111  0.7
External Textual Evidence
Important target pages have many incoming links, each with its own brief description of the target. Appropriately weighted, these annotations can be used to index the page they target. The text highlighted in your browser to indicate a link you can click on, is called anchor text.

Can you guess the URL?

The text highlighted in your browser to indicate a link you can click on, is called anchor text.
Click-associated queries

When a searcher enters a query and clicks on a document, we can associate that query with the document. Associated queries can be weighted by click frequency and used in indexing and retrieval.
A collaborative bookmarking tool can be used to tag a document, image or other resource with an annotation which is shared with other users.

Important resources receive many tags. The frequency of a tag -- indicated by the type size in a "tag cloud" display -- can be used as an indexing weight.

▶ See e.g. Dubinko et al, WWW 2006
Collecting tags
Should these external texts be treated as document fields?
Static factors

Adapted from Richardson, Prakash and Brill, WWW 2006

- **Incoming hyperlinks**
  - e.g. raw count, PageRank, Kleinberg Hub/Authority
- **Searcher behaviour data**
  - e.g. Frequency of visits to page (from toolbars, or proxy logs);
    Frequency of clicks when this page appears in search results;
    Average dwell time on the page
- **Query independent use of anchor text**
  - Amount of referring anchor text; Size of anchor text vocabulary
- **Page properties**
  - e.g. Word count; Frequency of most common term;
- **URL properties** (Kraiij & Westerveld, SIGIR 2002)
  - e.g. Length, depth of URL; type (root, subroot, page, dynamic)
- **Domain properties**
  - e.g. Average outlink count for pages in this domain.
- **Spam rating.**
  - e.g. Presence of AdSense ads!
- **Adult content score.**
PageRank

Initial PR value for all 15 nodes: 1/15

After convergence, which of A, B, C, D has highest PR?

- random surfer
- start with equal probability for all bookmarked pages
- follow outgoing links with equal probability
- teleport to a bookmark with probability $d$
There's no query!
Machine-Learning the Overall Ranking Function
\[ RSV = \alpha_0 D_0 + \ldots + \alpha_n D_n + \beta_0 S_0 + \ldots + \beta_n S_n \]  \hspace{1cm} (12)

- We need to be able to compute ranking quality for gezillions of combinations of the \(\alpha\)s and \(\beta\)s.
- Ranking quality is highly dependent upon the query so at each point we need to run very large numbers of queries and measure the quality of results.
Thoughts on a loss function

(Except for nerds like me, people don’t actually search for the fun of it. They do it in order to complete a task.)

- What we really want to optimise:
  - Proportion of search-facilitated tasks that people complete
  - How satisfactorily they complete them
  - How fast they complete them

- That’s very difficult. How can we do it?
User Studies

- Bring large numbers of human subjects into a laboratory and ask them to do search tasks.
- Measure their task performances.
- But:
  - Expensive
  - Not a real task – do the subjects do it properly?
  - Huge sources of variance to be controlled
    - individual differences
    - order effects
    - interactions
  - Results are set level – not reusable
In-Situ Studies

- Ask representative human subjects to use a two-panel search tool instead of their normal search engine.
- Controls for many of the problems
- Still not re-usable
- Explicit or implicit judgments.

Results are still set level – not reusable.
Observing natural user behaviour

- Via search engine or browser logs
- Where do people click?
  - Trust bias
  - Interpreting no-click
  - Increased frequency of clicks before and after page boundaries and “the fold”
- Can get preference judgments:
  - If someone skips over $D_n$ and clicks on $D_{n+1}$ we have evidence that they prefer $D_{n+1}$ to $DD_n$ for
- That could be input into a machine learning system.
Manipulating Rankings

- Reordering results
- Interleaving results
- Inserting results
- Observe behavioural differences
  - Flights and Buckets
- GYB do lots of this.
Cranfield? TREC? Huh?

<num> Number: 151

topic: Coping with overcrowded prisons

desc: Description:
The document will provide information on jail and prison overcrowding and how inmates are forced to cope with those conditions; or it will reveal plans to relieve the overcrowded condition.

narr: Narrative:
A relevant document will describe scenes of overcrowding that have become all too common in jails and prisons around the country. The document will identify how inmates are forced to cope with those overcrowded conditions, and/or what the Correctional System is doing, or planning to do, to alleviate the crowded condition.

/top>
Employ judges to assign relevance / utility scores to all documents (or for a large pool of documents which might possibly be relevant to the query).

- TREC pools—Union of top 100 docs for participating systems
- Results in re-usable test sets, modulo:
  - completeness
  - judging errors and disagreements
- TREC studies of stability of rankings across strict/l lenient judging
- GYB have large budgets for this.
  - Bing: 5 point scale, Gains are $2^n$
Issues with TREC style test sets

- Of what population are the TREC topics a representative sample?
- No penalty for duplicates – they’re very common
- No reward for diversification
- Solution: es.csiro.au/C-TEST
  - Interpretations
  - Differential utilities
  - Equivalence sets
C-TEST: CSIRO TOOLKIT FOR THE EVALUATION OF SEARCH TECHNOLOGY

David Hawking and Tom Rowlands

(draft four; not yet complete)

C-TEST implements a simple but generalised test-collection framework for evaluating the effectiveness of search systems. It is designed primarily for the evaluation of web and enterprise web search but is more generally applicable. It comprises:

- two XML file formats specifying:
  - queries, interpretations, weighted answers
  - results of a "run"
- a collection of tools to evaluate based on various metrics, sample from existing files, pool results for evaluation and use a web based tool for evaluation
- a library, written in Perl, to work with the files from your own tools

It is capable of appropriately evaluating the Homepage Finding, Named Page Finding and Topic Distillation tasks from the TREC Web Track. The framework effectively penalises the return of duplicate results and has a prominent recognition of the one query test having many potential, variably-weighted, interpretations.

C-TEST is distributed free under the Mozilla Public Licence. CSIRO welcome your suggestions, criticism and patches. We do, however, ask that if you are to use C-TEST in the development of an academic publication, you cite our C-TEST paper:


REQUIREMENTS AND INSTALLATION

C-TEST is tested on Ubuntu, Mac OS X and Windows XP systems. It almost certainly runs without modification on other systems as well. Packages that are required are Perl, LibXML, XML::Statistics::Distributions and File::Slurp. Directions for installing the latter two are included in the download.

DOWNLOAD

C-TEST pre-release Not for re-distribution (yet).

SITE
C-TEST Example

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<testfile name="Airline\u003eHomepage\u003eFinding">
  <query id="1" text="Qantas\u003eairways" weight="1.0" depth="10">
    <interpretation comment="homepage\u003efinding" weight="0.9">
      <eset util="10" comment="qantas\u003ehomepage">
        <docid>qantas.com</docid>
        <docid>www.qantas.com.au</docid>
        <docid>www.qantas.com/index.html</docid>
      </eset>
    </interpretation>
    <interpretation comment="share\u003eprice" weight="0.1">
      <eset util="10" comment="qantas\u003estock\u003exchange\u003utraading">
        <docid>asx.com.au/companylisting=QAN</docid>
      </eset>
      <eset util="3" comment="QANTAS\u003eshare\u003information\u003eprice">
        <docid>qantas.com.au/info/about/investors/shareholderInfo</docid>
      </eset>
    </interpretation>
  </query>
</testfile>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<results label="Airways\u003cHomepage\u003cFinding,\u003crun\u003cwith\u003calpha =0,\u003cbeta=2"">
  <query id="1" text="Qantas\u003cairways" weight="1.0" depth 
  ="10">
    <docid rank="1">www.qantas.com.au</docid>
    <docid rank="3">www.anzac.com/qantas/qantas.htm</docid>
    <docid rank="4">www.qantas.com/index.html</docid>
    <docid rank="5">www.quantas.com.au</docid>
    <docid rank="6">en.wikipedia.org/wiki/Qantas</docid>
    <docid rank="7">www.airlinequality.com/Forum/qantas. 
    htm</docid>
    <docid rank="8">www.travelmood.com/qantas.asp</docid>
    <docid rank="9">www.oneworld.com/ow/member-airlines/
    qantas</docid>
    htm</docid>
  </query>
</results>
C-TEST: Tools for

- Creating testfiles
  - From a spreadsheet
  - From TREC topics
  - By searching and browsing
  - By sampling a query log and judging

- Computing measures and significance testing of differences
LSE Case Study
Sources of testfiles at LSE

- A-Z Sitemap (≈500 entries)
  - Biased toward anchortext
- Keymatches file (≈500 entries)
  - Pessimistic
- Click data (≈250 queries with ≈t clicks)
  - Biased toward clicks - can achieve 100% success
- Random sample of workload, post-judged
- Popular/Critical queries (134 manually judged)
  - Optimising for searchers or for publishers
Tuning problem

▶ Approximately 40 parameters, some continuous, some binary, some integer
▶ Not much idea about the shape of the function.
  ▶ Pretty sure that there are multiple points of inflection.
▶ Some combinations make no sense
▶ Obviously brute-force grid search is impossible
▶ Even so, millions of query executions are needed.
Dimension at a time tuning
Where have we got with our tuning run?
LSE Tuning results (failure rates)

- Out-of-the-box: 24.63%
- As configured: 22.39%
- After tuning (DAAT mode): 8.21%
On the flipside of coffee ...

- Web SearchEngineering
- Field Work: how do Web search engines really work?
- Stretch Break
- Discussion: Other IR problems for machine learning
- Historical context