Introduction to Information Retrieval

Evaluation & Result Summaries
Evaluation issues

- Presenting the results to users
- Efficiently extracting the k-best
- Evaluating the quality of results
  - qualitative evaluation
  - quantitative evaluation
Presenting the Information to users

- Results summaries:
  - Making our “good results” usable to a user

Looking vs. Clicking

- Users view results one and two more often / thoroughly
- Users click most frequently on result one
Result Summaries

• Having ranked the documents matching a query, we wish to present a results list.

• Most commonly, a list of the document titles plus a short summary.

**John McCain**
John McCain 2008 - The Official Website of John McCain's 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for McCain; Americans with ...
www.johnmccain.com • Cached page

**JohnMcCain.com - McCain-Palin 2008**
John McCain 2008 - The Official Website of John McCain's 2008 Campaign for President ... African American Coalition; Americans of Faith; American Indians for McCain; Americans with ...
www.johnmccain.com/Informing/Issues • Cached page

**John McCain News- msnbc.com**
Complete political coverage of John McCain. ... Republican leaders said Saturday that they were worried that Sen. John McCain was heading for defeat unless he brought stability to ...
www.msnbc.msn.com/id/16438320 • Cached page

**John McCain | Facebook**
Welcome to the official Facebook Page of John McCain. Get exclusive content and interact with John McCain right from Facebook. Join Facebook to create your own Page or to start ...
www.facebook.com/johnmccain • Cached page
Summaries

• The title is typically automatically extracted from document metadata. What about the summaries?
  – This description is crucial.
  – User can identify good/relevant hits based on description.

• Two basic kinds:
  – Static
  – Dynamic

• A static summary of a document is always the same, regardless of the query that hit the doc

• A dynamic summary is a query-dependent attempt to explain why the document was retrieved for the query at hand
Static summaries

• In typical systems, the static summary is a subset of the document

• Simplest heuristic: the first 50 (or so – this can be varied) words of the document
  – Summary cached at indexing time

• More sophisticated: extract from each document a set of “key” sentences
  – Simple NLP heuristics to score each sentence
  – Summary is made up of top-scoring sentences.

• Most sophisticated: NLP used to synthesize a summary
  – Seldom used in IR; cf. text summarization work
Dynamic summaries

- Present one or more “windows” within the document that contain several of the query terms
  - “KWIC” snippets: Keyword in Context presentation
- Generated in conjunction with scoring
  - If query found as a phrase, all or some occurrences of the phrase in the doc
  - If not, document windows that contain multiple query terms
Generating dynamic summaries

- If we have only a positional index, we cannot (easily) reconstruct context window surrounding hits
- Remember **positional index**: for each term in the vocabulary, we store postings of the form docID: <position1, position2, ...>
- If we cache the documents at index time, can find windows in it, cueing from hits found in the positional index
  - E.g., positional index says “the query is a phrase/term in position 4378” so we go to this position in the cached document and stream out the content
- Most often, cache only a fixed-size prefix of the doc
  - Note: Cached copy can be outdated!
Dynamic summaries

• Producing good dynamic summaries is a tricky optimization problem
  – The real estate for the summary is normally small and fixed
  – Want short item, so show as many matches as possible, and perhaps other things like title
  – Want snippets to be long enough to be useful
  – Want linguistically well-formed snippets: users prefer snippets that contain complete phrases
  – Want snippets maximally informative about doc

• But users really like snippets, even if they complicate IR system design
Alternative results presentations?

• An active area of HCI research
• An alternative: http://www.searchme.com / copies the idea of Apple’s Cover Flow for search results
Efficiently extracting the k-best

- Heuristics for finding the top $k$ out of $N$ faster
  - Use a **binary min heap**
    - A binary min heap is a binary tree in which each node’s value is less than the values of its children.
    - It takes $O(N \log k)$ operations to construct the $k$-heap containing the $k$ largest values (where $N$ is the number of documents).
  - Essentially **linear in $N$** for small $k$ and large $N$. 
Binary min heap

0.6

0.85

0.9

0.97

0.7

0.8

0.95
Selecting $k$ top scoring documents in $O(N \log k)$

- **Goal:** Keep the $k$ top documents seen so far
- **Use a binary min heap**
- **To process a new document $d'$ with score $s'$:**
  - Get current minimum $h_m$ of heap (in $O(1)$)
  - If $s' \leq h_m$ skip to next document
  - If $s' > h_m$ heap-delete-root (in $O(\log k)$)
  - Heap-add $d'/s'$ (in $O(1)$)
  - Reheapify (in $O(\log k)$)
Evaluating the quality of results

1. Unranked evaluation (evaluation that does not take into account the rank of a document)

3. Ranked evaluation

4. Evaluation benchmarks

5. Result summaries
Measures for a search engine

- How fast does it index
  - e.g., number of bytes per hour
- How fast does it search
  - e.g., latency as a function of queries per second
- What is the cost per query?
  - In currency (dollars/euros)
Measures for a search engine

- All of the preceding criteria are measurable: we can quantify speed / size / money
- However, the key measure for a search engine is user happiness.
- What is user happiness?
- Factors include:
  - Speed of response
  - Size of index
  - User Interface
  - Most important: relevance
- (actually, maybe even more important: it’s free)
- Note that none of these is sufficient: blindingly fast, but useless answers won’t make a user happy.
- How can we quantify user happiness?
Your turn

• Measures of happiness?
Who is the user?

- Who is the user we are trying to make happy?
- Web search engine: *searcher*. Success: Searcher finds what he was looking for. **Measure**: rate of return to this search engine
- Web search engine: *advertiser*. Success: Searcher clicks on ad. **Measure**: clickthrough rate
- Ecommerce: *buyer*. Success: Buyer buys something. **Measures**: time to purchase, fraction of “conversions” of searchers to buyers
- Ecommerce: *seller*. Success: Seller sells something. **Measure**: profit per item sold
- Enterprise: *CEO (chief executive)*. Success: Employees are more productive (because of effective search). **Measure**: profit of the company
Most common definition of user happiness: Relevance

- User happiness is equated with the relevance of search results to the query.
- But how do you measure relevance?
- Standard methodology in information retrieval consists of three elements.
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of the relevance of each query-document pair
Relevance: query vs. information need

- Relevance to what?
- First take: relevance to the query
- “Relevance to the query” is very problematic.
- Information need $i$: “I am looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.”
- This is an information need, not a query.
- Query $q$: [red wine white wine heart attack]
- Consider document $d'$: At heart of his speech was an attack on the wine industry lobby for downplaying the role of red and white wine in drunk driving.
- $d'$ is an excellent match for query $q$ . . .
- $d'$ is not relevant to the information need $i$. 
Relevance: query vs. information need

- User happiness can only be measured by relevance to an information need, not by relevance to queries.
- Our terminology is sloppy in these slides and in IR: we talk about query-document relevance judgments even though we mean information-need-document relevance judgments.
Standard measures of relevance: Precision and Recall

- **Precision** ($P$) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant} | \text{retrieved})$$

- **Recall** ($R$) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved} | \text{relevant})$$
Precision and recall

<table>
<thead>
<tr>
<th></th>
<th>Relevant</th>
<th>Nonrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>true positives (TP)</td>
<td>false positives (FP)</td>
</tr>
<tr>
<td>Not retrieved</td>
<td>false negatives (FN)</td>
<td>true negatives (TN)</td>
</tr>
</tbody>
</table>

\[ P = \frac{TP}{TP + FP} \]
\[ R = \frac{TP}{TP + FN} \]
Precision/recall tradeoff

- You can increase recall by returning more docs.
- Recall is a non-decreasing function of the number of docs retrieved.
- A system that returns all docs has 100% recall!
- The converse is also true (usually): It’s easy to get high precision for very low recall.
- Suppose the document with the largest score is relevant. How can we maximize precision?
A combined measure: $F$

- $F$ allows us to trade off precision against recall.

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

where $\beta^2 = \frac{1 - \alpha}{\alpha}$

- $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, \infty]$
- Most frequently used: balanced $F$ with $\beta = 1$ or $\alpha = 0.5$
  - This is the harmonic mean of $P$ and $R$: $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$
F: Example

<table>
<thead>
<tr>
<th></th>
<th>relevant</th>
<th>not relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieved</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>not retrieved</td>
<td>60</td>
<td>1,000,000</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1,000,040</td>
</tr>
</tbody>
</table>

- $P = \frac{20}{20 + 40} = \frac{1}{3}$
- $R = \frac{20}{20 + 60} = \frac{1}{4}$
- $F_1 = 2\frac{1}{\frac{1}{3} + \frac{1}{4}} = \frac{2}{7}$
Exercise

- Compute precision, recall and $F_1$ for this result set:
  
<table>
<thead>
<tr>
<th></th>
<th>Relevant</th>
<th>Not Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>Not Retrieved</td>
<td>82</td>
<td>1,000,000,000</td>
</tr>
</tbody>
</table>

- The snoogle search engine below always returns 0 results (“0 matching results found”), regardless of the query. Why does snoogle demonstrate that accuracy is not a useful measure in IR?
Accuracy

- Why do we use complex measures like precision, recall, and $F$?
- Why not something simple like accuracy?
- Accuracy is the fraction of decisions (relevant/nonrelevant) that are correct.
- In terms of the contingency table above, accuracy = \( \frac{TP + TN}{TP + FP + FN + TN} \).
- Why is accuracy not a useful measure for web information retrieval?
Why accuracy is a useless measure in IR

- Simple trick to maximize accuracy in IR: always say no and return nothing
- You then get 99.99% accuracy on most queries.
- Searchers on the web (and in IR in general) want to find something and have a certain tolerance for junk.
- It’s better to return some bad hits as long as you return something.
- → We use precision, recall, and $F$ for evaluation, not accuracy.
F: Why harmonic mean?

- Why don’t we use a different mean of $P$ and $R$ as a measure?
  - e.g., the arithmetic mean
- The simple (arithmetic) mean is 50% for “return-everything” search engine, which is too high.
- Desideratum: Punish really bad performance on either precision or recall.
- Taking the minimum achieves this.
- But minimum is not smooth and hard to weight.
- $F$ (harmonic mean) is a kind of smooth minimum.
$F_1$ and other averages

- We can view the harmonic mean as a kind of soft minimum
Difficulties in using precision, recall and $F$

- We need relevance judgments for information-need-document pairs – but they are expensive to produce.
- For alternatives to using precision/recall and having to produce relevance judgments – see end of this lecture.
Outline

1. Unranked evaluation
2. Ranked evaluation (what if we consider ranking?)
3. Evaluation benchmarks
4. Result summaries
Precision, recall, and the F measure are set-based measures. They are computed using unordered sets of documents.

We need to extend these measures (or to define new measures) if we are to evaluate the ranked retrieval results that are now standard with search engines.

We can easily turn set measures into measures of ranked lists.

Just compute the set measure for each “prefix”: the top 1, top 2, top 3, top 4 etc results.

Doing this for precision and recall gives you a precision-recall curve.
Jig-saw precision/recall curve

Precision-recall curves have a distinctive saw-tooth shape: if the document retrieved is nonrelevant then recall is the same as for the top documents, but precision has dropped. If it is relevant, then both precision and recall increase, and the curve jags up and to the right.
Jig-saw precision-recall curve

- Each point corresponds to a result for the top $k$ ranked hits ($k = 1, 2, 3, 4, \ldots$).
- Interpolation (in red): Take maximum of all future points.
- Rationale for interpolation: The user is willing to look at more stuff if both precision and recall get better.

K=1 outcomes: $+$ or $-$ $P=1$ or $0$, $R=1/N$ or $0$

K=1 outcomes: $++$, --, +-,+- $P=1$ or $0.5$ or $0$, $R=1/N$ or $2/N$ or $0$
11-point interpolated average precision

<table>
<thead>
<tr>
<th>Recall</th>
<th>Interpolated Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.67</td>
</tr>
<tr>
<td>0.2</td>
<td>0.63</td>
</tr>
<tr>
<td>0.3</td>
<td>0.55</td>
</tr>
<tr>
<td>0.4</td>
<td>0.45</td>
</tr>
<tr>
<td>0.5</td>
<td>0.41</td>
</tr>
<tr>
<td>0.6</td>
<td>0.36</td>
</tr>
<tr>
<td>0.7</td>
<td>0.29</td>
</tr>
<tr>
<td>0.8</td>
<td>0.13</td>
</tr>
<tr>
<td>0.9</td>
<td>0.10</td>
</tr>
<tr>
<td>1.0</td>
<td>0.08</td>
</tr>
</tbody>
</table>

It is often useful to remove these jiggles and the standard way to do this is with an interpolated precision: the *interpolated precision* at a certain recall level $r$ is defined as the highest precision found for any recall level $r' > r$

$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

11-point average: $\approx 0.425$
Averaged 11-point precision/recall graph

- Compute interpolated precision at recall levels 0.0, 0.1, 0.2, . . .
- Do this for each of the queries in the evaluation benchmark
- Average over queries
- This measure measures performance at all recall levels.
- Note that performance is not very good!
Variance of measures like precision/recall

- For a test collection, it is usual that a system does badly on some information needs (e.g., $P = 0.2$ at $R = 0.1$) and really well on others (e.g., $P = 0.95$ at $R = 0.1$).
- Indeed, it is usually the case that the variance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones.
Outline

1. Unranked evaluation
2. Ranked evaluation
3. Evaluation benchmarks
4. Result summaries
What we need for a benchmark

- A collection of documents
  - Documents must be representative of the documents we expect to see in reality.

- A collection of information needs
  - . . .which we will often incorrectly refer to as queries
  - Information needs must be representative of the information needs we expect to see in reality.

- Human relevance assessments
  - We need to hire/pay “judges” or assessors to do this.
  - Expensive, time-consuming
  - Judges must be representative of the users we expect to see in reality.
Standard relevance benchmark: Cranfield

- Pioneering: first testbed allowing precise quantitative measures of information retrieval effectiveness
- Late 1950s, UK
- 1398 abstracts of aerodynamics journal articles, a set of 225 queries, exhaustive relevance judgments of all query-document-pairs
- Too small, too untypical for serious IR evaluation today
Standard relevance benchmark: TREC

- **TREC** = Text Retrieval Conference (TREC)
- Organized by the U.S. National Institute of Standards and Technology (NIST)
- TREC is actually a set of several different relevance benchmarks.
- Best known: TREC Ad Hoc, used for first 8 TREC evaluations between 1992 and 1999
- 1.89 million documents, mainly newswire articles, 450 information needs
- No exhaustive relevance judgments – too expensive
- Rather, NIST assessors’ relevance judgments are available only for the documents that were among the top k returned for some system which was entered in the TREC evaluation for which the information need was developed.
Text REtrieval Conference (TREC)
...to encourage research in information retrieval from large text collections.

Overview

Publications

Other Evaluations

Information for Active Participants

Frequently Asked Questions

Tracks

Data

Past TREC Results

Contact Information

TREC 2008 Call for Participation
Contextual Suggestion Track

The Contextual Suggestion track investigates search techniques for complex information needs that are highly dependent on context and user interests.

Track coordinators:
Charles L A Clarke, University of Waterloo
Adriel Dean-Hall, University of Waterloo
Jaap Kamps, University of Amsterdam
Paul Thomas, CSIRO

Track Web Page:
http://sites.google.com/site/treccontext/

Mailing list:
Send a mail message to listproc (at) nist.gov such that the body consists of the line subscribe trec-context <FirstName> <LastName>

Crowdsourcing Track

The Crowdsourcing track investigates emerging crowd-based methods for search evaluation and/or developing hybrid automation+crowd search systems.

Track coordinators:
Gabriella Kazai, Microsoft Research
Matt Lease, University of Texas at Austin
Mark Smucker, University of Waterloo

Track Web Page:
https://sites.google.com/site/treccrowd

Mailing list:
http://groups.google.com/group/trec-crowd
Contextual Suggestion Track

- Future IR systems should anticipate user needs without asking the user to specify all the details.
- For example, imagine a group of information retrieval researchers with a November evening to spend in beautiful Gaithersburg, Maryland. A contextual suggestion system might recommend a beer at the Dogfish Head Alehouse (www.dogfishalehouse.com), dinner at the Flaming Pit (www.flamingpitrestaurant.com), or even a trip into Washington on the metro to see the National Mall (www.nps.gov/nacc).
Contextual Suggestion Track (2)

• The goal of the Contextual Suggestion track is to provide a venue for the evaluation of such systems.

• For context suggestions, systems are provided with user preferences in relation with a set of suggestions (previous choices) and must learn “generalised” preference model for users. Furthermore, they are provided with contexts

```xml
<context number="1">
  <city>Portland</city>
  <state>Oregon</state>
  <lat>45.5</lat>
  <long>-122.7</long>
  <day>weekday</day>
  <time>evening</time>
  <season>fall</season>
</context>
```
Federated search is the approach of querying multiple search engines simultaneously, and combining their results into one coherent search engine result page.

KBA seeks to help humans expand knowledge bases like Wikipedia by automatically recommending edits based on incoming content streams. This open evaluation measures an automatic system's ability to filter a large stream of text for new knowledge about entities.

1) Initialize with a target entity and info need
2) Iterate over stream of text items
3) For each, output confidence between 0, 1

- Content Stream
  - 462M texts, 40% English
  - 4,973 hourly chunks of a $10^5$ docs/hour
  - News, blogs, forums, and link shortening

KBA System
Microblog Track

The Microblog track examines the nature of real-time information needs and their satisfaction in the context of microblogging environments such as Twitter.

Track coordinators:
Miles Efron, University of Illinois
Jimmy Lin, University of Maryland

Track Web Page:
https://sites.google.com/site/microblogtrack/

Mailing list:
http://groups.google.com/group/trec-microblog

Session Track

The Session track aims to provide the necessary resources in the form of test collections to simulate user interaction and help evaluate the utility of an IR system over a sequence of queries and user interactions, rather than for a single "one-shot" query.

Track coordinators:
Ben Carterette, University of Delaware
Evangelos Kanoulas, Google Zurich
Mark Sanderson, RMIT University
Paul Clough, University of Sheffield

Track Web Page:
http://ir.cis.udel.edu/sessions

Mailing list:
Use the link given on the track web page to join the email list.
Temporal Summarization Track

The goal of the Temporal Summarization track is to develop systems that allow users to efficiently monitor the information associated with an event over time.

Track coordinators:
Javad Aslam, Northeastern University
Fernando Diaz, Microsoft Research
Matthew Ekstrand-Abueg, Northeastern University
Virgil Pavlu, Northeastern University
Tetsuya Sakai, Microsoft Research Asia

Track Web Page:
http://www.trec-ts.org/

Mailing list:
http://groups.google.com/group/temporalsummarization2013

Web Track

The goal of the Web track is to explore and evaluate Web retrieval technologies that are both effective and reliable.

Track coordinators:
Kevyn Collins-Thompson, Microsoft Research
Paul N. Bennett, Microsoft Research
Fernando Diaz, Microsoft Research
Charles Clarke, University of Waterloo

Track Web Page:

Mailing list:
Send a mail message to listproc (at) nist.gov such that the body consists of the line subscribe trec-web <FirstName> <LastName>
Standard relevance benchmarks: Others

- GOV2
  - Another TREC/NIST collection
  - 25 million web pages
  - Used to be largest collection that is easily available
  - But still 3 orders of magnitude smaller than what Google/Yahoo/MSN index

- NTCIR
  - East Asian language and cross-language information retrieval

- Cross Language Evaluation Forum (CLEF)
  - This evaluation series has concentrated on European languages and cross-language information retrieval.

- Many others
Validity of relevance assessments

- Relevance assessments are only usable if they are consistent.
- If they are not consistent, then there is no “truth” and experiments are not repeatable.
- How can we measure this consistency or agreement among judges?
- → Kappa measure
Kappa measure

- Kappa is a measure of how much judges agree or disagree.
- Designed for categorical judgments.
- Corrects for chance agreement.
- $P(A) = \text{proportion of time judges agree}$
- $P(E) = \text{what agreement would we get by chance}$

\[
\kappa = \frac{P(A) - P(E)}{1 - P(E)}
\]
Kappa measure (2)

- Values of $k$ in the interval $[2/3, 1.0]$ are seen as acceptable.
- With smaller values: need to redesign relevance assessment methodology used etc.
Calculating the kappa statistic

$$\begin{array}{c|ccc}
\text{Judge 2 Relevance} & \text{Yes} & \text{No} & \text{Total} \\
\hline
\text{Yes} & 300 & 20 & 320 \\
\text{No} & 10 & 70 & 80 \\
\text{Total} & 310 & 90 & 400 \\
\end{array}$$

- **Judge 1 Relevance**
  - Yes
  - No
  - Total

**Observed proportion of the times the judges agreed**

\[ P(\text{Agreement}) = \frac{300 + 70}{400} = \frac{370}{400} = 0.925 \]

**Pooled marginals**

\[ P(\text{nonrelevant}) = \frac{80(J1) + 90(J2)}{400 + 400} = \frac{170}{800} = 0.2125 \]
\[ P(\text{relevant}) = \frac{320(J1) + 310(J2)}{400 + 400} = \frac{630}{800} = 0.7878 \]

**Probability that the two judges agreed by chance** \[ P(E) = P(\text{nonrelevant})^2 + P(\text{relevant})^2 = 0.2125^2 + 0.7878^2 = 0.665 \]

**Kappa statistic** \[ \kappa = \frac{P(A) - P(E)}{1 - P(E)} = \frac{0.925 - 0.665}{1 - 0.665} = 0.776 \text{ (still in acceptable range)} \]
Inter-judge agreement at TREC

<table>
<thead>
<tr>
<th>Information need</th>
<th>number of docs judged</th>
<th>disagreements</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>211</td>
<td>6</td>
</tr>
<tr>
<td>62</td>
<td>400</td>
<td>157</td>
</tr>
<tr>
<td>67</td>
<td>400</td>
<td>68</td>
</tr>
<tr>
<td>95</td>
<td>400</td>
<td>110</td>
</tr>
<tr>
<td>127</td>
<td>400</td>
<td>106</td>
</tr>
</tbody>
</table>
Impact of inter-judge disagreement

- Judges disagree a lot. Does that mean that the results of information retrieval experiments are meaningless?
- No.
- Large impact on absolute performance numbers
- Virtually no impact on ranking of systems
- Suppose we want to know if algorithm A is better than algorithm B
- An information retrieval experiment will give us a reliable answer to this question . . .
- . . . even if there is a lot of disagreement between judges.
Evaluation at large search engines

- Recall is difficult to measure on the web
- Search engines often use precision at top $k$, e.g., $k = 10$ . . .
- . . . or use measures that reward you more for getting rank 1 right than for getting rank 10 right.
- Search engines also use non-relevance-based measures.
  - Example 1: *clickthrough* on first result
  - Not very reliable if you look at a single clickthrough (you may realize after clicking that the summary was misleading and the document is nonrelevant) . . .
  - . . . but *pretty reliable* in the aggregate.
A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.

- Probably the evaluation methodology that large search engines trust most (e.g. Google)
Critique of pure relevance

- We’ve defined relevance for an isolated query-document pair.
- Alternative definition: marginal relevance
- The **marginal relevance** of a document at position $k$ in the result list is the additional information it contributes over and above the information that was contained in documents $d_1 \ldots d_{k-1}$. 
## Example

### System A: Ranked pages (C=correct W=Wrong)

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>.....</th>
<th>Qn</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A2</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A3</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A4</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A5</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
</tbody>
</table>

### System B: Ranked pages

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>.....</th>
<th>Qn</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A2</td>
<td>C</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>.....</td>
<td>C</td>
</tr>
<tr>
<td>A3</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A4</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
<tr>
<td>A5</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>.....</td>
<td>W</td>
</tr>
</tbody>
</table>
Mean Reciprocal Rank (MRR)

• Score for an individual question:
  – The reciprocal of the rank at which the first correct answer is returned
  – 0 if no correct response is returned

• The score for a run:
  – Mean over the set of questions in the test
**MRR in action**

**System A:** \[ MRR = \frac{(.2+1+1+.2)}{10} = 0.24 \]

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th></th>
<th>Qn</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A2</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A3</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A4</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A5</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>

**System B:** \[ MRR = \frac{(.5+.33+.5+.25+1+.5+.5+.5+.5)}{10}=0.42 \]

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th></th>
<th>Qn</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A2</td>
<td>C</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>A3</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A4</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
<tr>
<td>A5</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>