Rapid Deployment of High-quality Search: Removing the Evaluation Bottleneck

Einat Amitay
David Carmel
Ronny Lempel
Aya Soffer

IBM Haifa Research Lab

Introduction

- In order to deliver high quality search, one must be able to measure and quantify search quality.
- Likewise, when contemplating which search system to buy, customers would like to measure the quality provided by candidate systems.
- Current IR methodology for measuring search quality is extremely labor intensive and cumbersome.
- Goal: ease the quality evaluation process, essentially removing the bottleneck it imposes today on search system deployment.

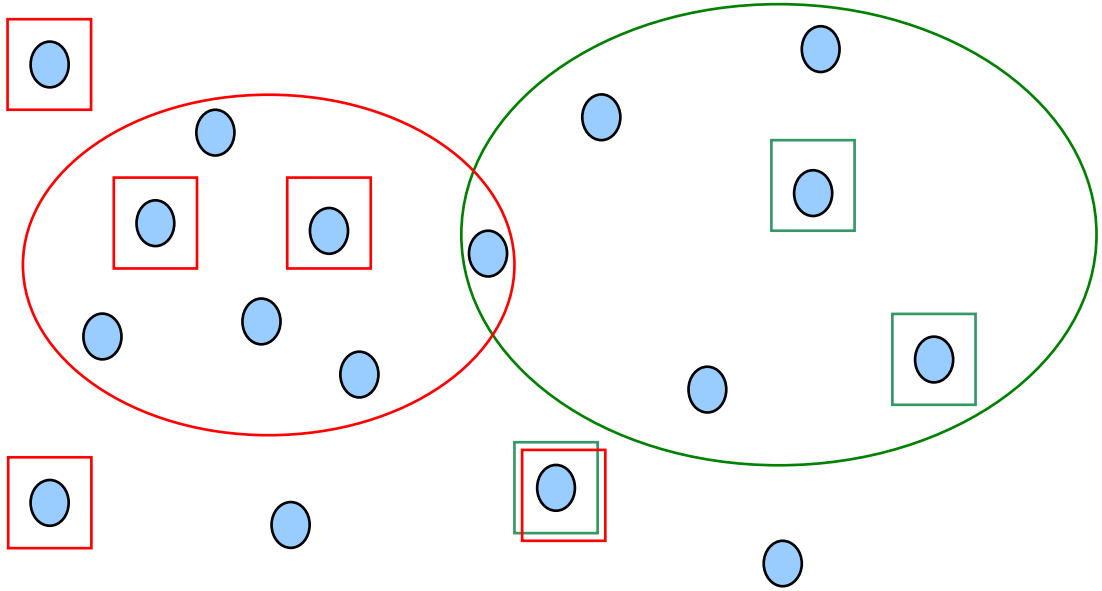
Talk Outline

- Current accepted methodology for search quality evaluation
  - Deficiencies of said methodology
- Related work
- Proposed methodology: evaluation using Term Relevance Sets
  - Does it solve the problems of the traditional approaches?
  - Experiments: is the new scheme reliable? Is it robust?
- Conclusions and future work

Quality Evaluation by Relevance Judgments – Legacy Scheme

- Fix a corpus (a collection of documents), and a set of queries.
- Scan the entire corpus, and mark each document as relevant/irrelevant with respect to each query.
- Given a search system, run the queries and examine number (and rank) of returned relevant documents.
- Assign a score to the search system, aggregating its achievements over the set of queries.

Legacy Scheme - Example

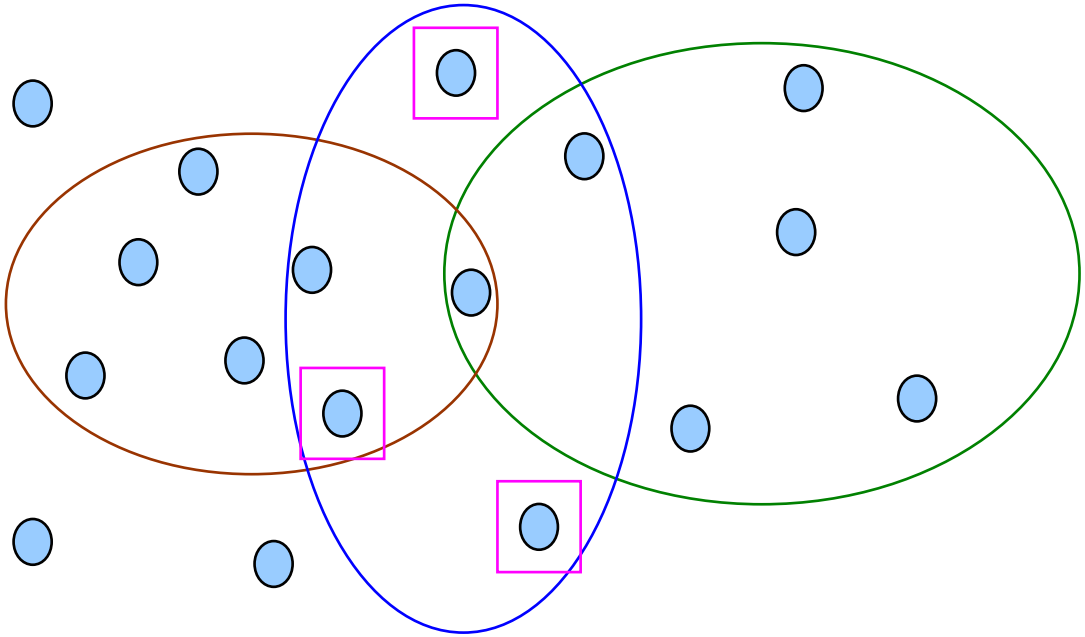


Quality Evaluation by Relevance Judgments, Using Pools

Following approach is useful in IR competitions, when several search systems are pitted against each other.

- Fix a corpus and a set of queries. For each query:
  - run all systems and form a “pool” of results – the union of the results returned by all systems.
  - Scan the entire pool, and mark each document as relevant/irrelevant with respect to the query.
  - Re-examine the results returned by each individual system, and score the system with respect to number (and rank) of returned relevant documents.
- Assign a score to each search system, aggregating its achievements over the set of queries.

Pooling - Example



Problem #1: Scale

- Judging the relevance of each document to each query is clearly not feasible for modern-day collections of millions of documents.
- Judging the relevance of a large pool of documents per query, while feasible, is labor intensive.
- BTW, how can one get hold of a varied enough pool of results in a competitive market scenario?

Problem #2: Moving Targets

- Many interesting collections today are extremely dynamic:
  - New documents are added daily
  - Existing documents are modified and deleted at high rates
- Keeping relevance judgments up to date in such a volatile environment is extremely challenging: documents that matched a query yesterday
  - May not exist today
  - May exist but no longer be relevant
  - May still be relevant, but less so than newly added content
- In particular, what if our score function explicitly favors new documents, naturally pushing yesterday's marked results to low-ranking spots?

Problem #3: Bad Coverage vs. Inappropriate Score Function

- Index may not cover entire collection.
- Documents marked as relevant may be in non-covered part of collection.
- When relevant documents fail to appear in search results, it is unclear whether problem stems from bad coverage, or from bad treatment of indexed content.
- Quality evaluation should separate the two issues, to better pinpoint what needs to be fixed (coverage policy vs. rank function).

Related Work – Avoiding Manual Inspection of Pooled Documents

- Several research efforts have tried to circumvent the manual inspection of pooled documents.
- Common theme: given a pool, automatically identify relevant documents, and continue as if those were manually marked.
- Most approaches have positive correlation with original, labor-intensive approach – about 0.5 similarity between the rankings of competing systems (i.e., about 75% of all pairs of systems are ranked in the same relative order).

Related Work – Avoiding Manual Inspection of Pooled Documents

Example: Wu & Crestani, 2003

- Given a pool of returned results for some query, count the number of systems that returned each document.
- Score each system by summing the counts achieved by the documents it returned.
- Basically, a system scores well when the results it returns overlap with results returned by other systems.
- Favors conservative systems, and punishes new retrieval techniques that tend to return unique documents.

Quality Evaluation Using Term Relevance Sets (Trels)

- Measure quality of returned results based on their content (appearance of some terms), rather on their identifiers.
- An IR system is evaluated over a set of queries.
- Each query is associated with two sets of terms:
  - On-topic terms – terms related to the query that are likely to appear in relevant documents and that are indicators of relevancy.
  - Off-topic terms – terms related to some of the query's words. They are not likely to appear in relevant documents, and are typically indicators of irrelevancy or topic-drift.

Quality Evaluation Using Term Relevance Sets (cont.)

- The IR system executes each query, returning a set of result documents per query.
- Returned documents are scored: appearances of on-topic terms will contribute to the score, while off-topic terms will lower it.
- The scores of all documents returned for a certain query are aggregated into a score for the query.
  - Aggregation may take into account the rank/score attributed by the search system to each document.
- An overall score of an IR system is the average of its scores over all queries.

Term Relevance Sets - Examples

- Query: recycle automobile tires
- On-topic: “rubberized asphalt”, “door mats”, playground, fish habitats
- Off-topic: traction, air-pressure, paper, plastic, glass, “all terain”, “winter tires”, Goodyear

- Query: Stirling engine
- On-topic: pressure*gas, hot*cold, “Robert Stirling”, temperature*difference, “displacer piston”
- Off-topic: “internal combustion”, search, “University of Stirling”, Stirling*district, Scotland

Creation of Trels

- Choosing the queries:
  - Queries should ideally contain some ambiguous terms, along with a disambiguating term, or a general term with a more precise modifier.
  - Such queries naturally test an IR system's ability to fully exploit the information contained in the query.
- Collecting on-topic terms:
  - Non-experts can submit the query to a search engine, and examine several good results. Then, they should select terms that distinctly identify the precise topic of the query.
  - An expert on a topic can often produce on-topic terms without even resorting to examining relevant documents.
  - Some natural on-topic terms include acronym expansion.

Creation of Trels (cont.)

- Collecting off-topic terms:
  - Submit “partial” queries (without some disambiguating terms) to a search engine.
  - Identify documents relevant to the partial queries but irrelevant to the whole query.
  - Identify common terms in such documents, that usually do not appear in relevant documents.
- Connection to query refinement:
  - On-topic terms are often those one would add to the query when refining it.
  - Off-topic terms are often those one would add to the query with a preceding minus, to exclude documents containing them.

Experiments

- Used official TREC datasets, queries, and relevance judgments.
- Measured effectiveness of a variety of IR systems.
- Compared Trel-based scores to official TREC measures based on relevance judgments.
- Found very high correlation between TREC-based and Trel-based scores.
- Found high correlation between the rankings induced by the TREC measures and those induced by Trels.

First Experiment - Data

- TREC-8 collection (528K text documents, assembled in the early 90's).
- 50 TREC topics, defined by short "Web-like" queries.
- For each of the 50 queries, results of 128 IR systems.
- Relevance judgments on all returned results, from which two official TREC measures (p@10, MAP) can be derived per system.

First Experiment - Setup

- Selected 27/50 queries, and constructed Trels using documents pertaining to the queries found on the Web in late 2003 (no overlap with TREC documents).
- On average: 32.5 on-topic terms, 8.5 off-topic terms.
- Trel-scored each of the 128 IR systems w/respect to the results each returned for the 27 queries.
- Calculated the TREC-based measures p@10, MAP for each of the 128 systems, based on the results each returned for the 27 queries.
- Compared the correlation of the Trel-based scores and rankings with those of the two official TREC measures.

First Experiment - Results

	Correlation of scores	Correlation of rankings
p@10-Trels	0.952	0.732
MAP-Trels	0.938	0.738
MAP-p@10	0.956	0.842

- Trel-based scores are as correlated with Trec's official scores as the two Trec measures are correlated with each other.
- Rank-correlation somewhat lower, but still much better than in previous works that avoided collecting manual relevance judgments.

Second Experiment - Data

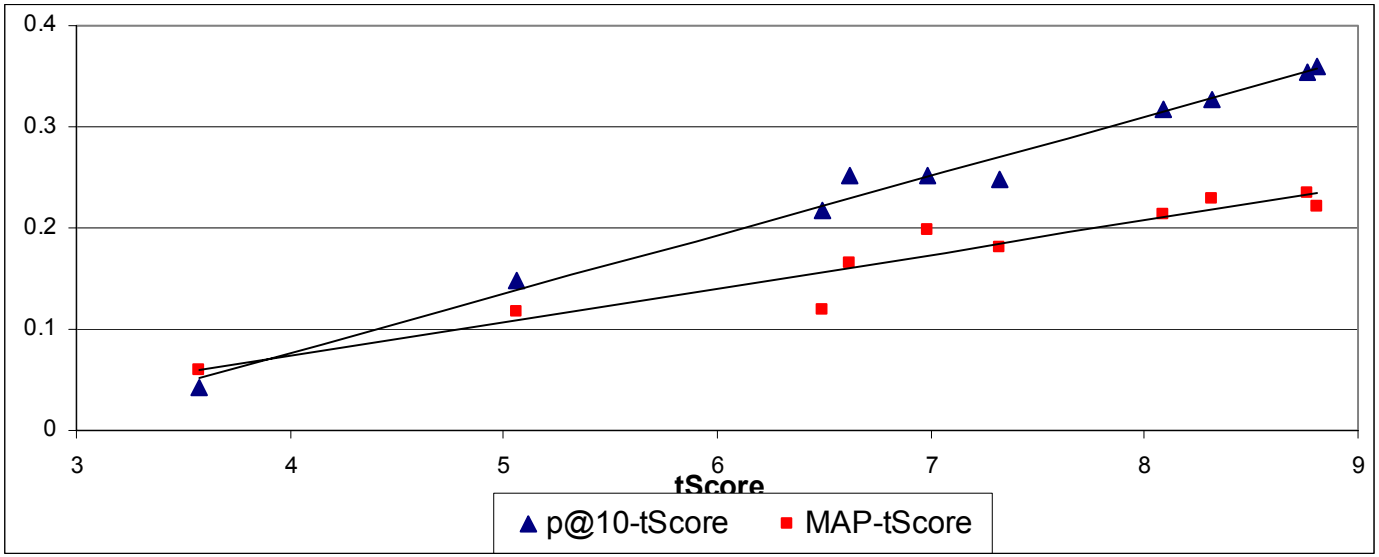
- WT10g collection (1.69M Web pages, assembled in 2001).
- 50 WT topics, defined by short “Web-like” queries, as well as the 27 queries with Trels of the first experiment.
- Relevance judgments on all returned results by any of the participants in TREC’s Web Track, 2001.
- 10 variants/flavors of our Juru search engine.

Second Experiment - Setup

- Each Juru variant executed the 50 WT queries over the WT10g collection, and was assigned TREC-based scores using the known relevance judgments.
- Each Juru variant executed the 27 Trec-8 queries over the WT10g collection, and was assigned Trel-based scores.
- Again, compared the correlation of the Trel-based scores and rankings with those of the two official TREC measures.

Second Experiment - Results

	Correlation of scores	Correlation of rankings
p@10-Trels	0.993	0.866
MAP-Trels	0.960	0.822
MAP-p@10	0.961	0.866



Summary - Advantages of Trels

- Trels are scale-free: quality Trels enable quality evaluation on collections of arbitrary sizes.
- Trels are forever: for many queries, they expire very slowly, if at all.
- Trels are global: can be developed independently of any specific corpus, and reused over many collections.
- Trels are feasible: choosing appropriate queries and composing Trels for them, while not trivial, is by far less laborious than collecting relevance judgments.

Future Work

- Facilitate the creation of Trels (methodology for selection of proper on/off topic terms).
- Examine and quantify the robustness of Trels – how many terms are needed, how sensitive are measurements to specific term sets.
- Weighted Trels – certain terms in the on/off topic sets might be considered as more indicative of topicality. Can such weights be assigned automatically?