Relevancy Testing Outline

Introduction

It's nice to be able to compare one search engine to another in some unbiased way. You point a couple search engines at the same documents and run some searches. The test tool then pops a numerical grade. These grades can either compare each engine to what humans have said are the correct answers, or compares one engine directy to another, and at least determins which one did better. **TREC** has been one of the groups trying to perfect this type of testing over the years.

As simple as this sounds, quite a few issues come up that make this more difficult than you might think:

- You need to get a large set of documents and searches to test with. This can be a problem for copyright reasons, among other things.
- You need to come up with searches that represent the types of things real users would search for, and get everybody to agree on what's a fair test search.
- Getting humans to sit down and record what the best matches should be is very time consuming for large data sets.
- Then you need to decide which mathematical formulas you'll use to crank out the final scores. For example, is it better to have an engine that gives modestly relevant results almost all the time, or an engine that gives really good answers sometimes, better on average than the other engine, but sometimes gives back complete garbage?

Some of these problems are so time consuming that other groups have taken a radically different approach. Instead of worrying about the "right" or "wrong" answers, they just have users try both engines, and see which one they like better! This type of **AB Testing** is much more common on the World Wide Web. Of course there are also many technical details with these tests. For example, should users specifically tell you which search results they like better, or should you just randomly change it behind the scenes and see which one produces more clicks?

Things get a bit more technical at this point. Below is an outline and links to more detailed pages. Remember this is a work in progress, and something you can contribute to.

Two General Types of Testing

- Absolute Truth / Matrix / Grid / TREC / Relevancy Assertions
  - The correct answers for each search are known ahead of time
  - Humans judges often decide these correct answers, stored as **Relevancy Assertions**
  - Can be labor intensive to setup
  - continued on the **Relevancy Assertion Testing** page
- AB Testing / User Preference
  - Tracks explicit or implicit preferences between engines A/B
  - Often dispenses with the notion of the "correct" answer
  - Can be easier to setup, but some fear the best answers will be missed by both engines
  - continued on the **AB Testing** page

Beyond Precision and Recall: How Engines are Judged

- Binary vs. Non-Binary Grading Systems
  - Early TREC had binary judgements, only Yes/No on whether each doc was related to a test search
  - More choices were later added
  - A system can use letter grades (A, B, C, D and F) or numeric grades
  - Another style asks testers to sort documents in their preferred order
- Classic Measurements: Precision and Recall
  - Recall: "Did I find all the documents I expected to get back? What percent?"
  - Precision: "Did the system bring back other documents that weren't relevant? What percent were on target?"
- Newer Ideas:
  - Rank: The order of documents that were returned
    - Generally a 1 in 20 match in the #1 spot is better than a 50% rate where all matches are on the second page.
  - Interactivity: What navigators or visualization were given to help the user iteratively drill down and find what they were looking for
    - Facets and sorting: Clickable filters and sort options
    - Unsupervised Clustering: Related terms or phrases, or related searches
    - Spelling and thesaurus suggestions
  - Subject Disambiguation, Sentiment, Conflicting Information, crowd hints
    - **kidney** *bean* or **kidney** *cell*
    - "**best** football team in the UK"
- *Mathematical assessments generally covered under implementations.*

Sources of Variance, AKA "Problems"

- Different Goals
  - Perfect/Human vs. Best vs. Acceptable vs. Better than X
  - Constrained vs. Unconstrained Resources (time, cpu, storage)
- Sample Size
  - Amount of Data
    - Fixed set or growing over time
  - Number of Testers (AB or Relevancy Judgments)
  - Number of Searches
- Vertical vs. Horizontal Content
  - One extreme: Specific demo may cover just one discipline, for example Medical Journals
  - Other extreme: Internet covers vastly disparate domains
- Vocabulary Variation / Mistmatch: Search vs. Content
- Users

        - Experienced vs. New Searcher
        - Subject Expert vs. Novice
        - Spelling, typing and computer proficiency
        - Reading Level, Native Language, IQ
        - Interface Medium (large visual display, small text display, audible, Braille, etc)
        - Amount of Effort to understand Search
        - Willingness to Iterate
        - Searching for specific answer vs. General Exploration
- Type of Searches
    - Length / 1 or 2 words
    - Full question
    - Sample text
    - Internet Boolean
    - Advanced Boolean / Syntax / Proximity
        - Wildcard, Regex, etc.
- Abbreviations
- Punctuation
    - Chemical
    - Source Code
    - Units of Measure
    - Literal vs. Search Operator
- Popular vs. Outlier / Researcher
- Potential for Shared Search Engine Biases
    - TF/IDF
    - Shared Thesaurus
    - Similar fuzzy matching (Snowball, Soundex, etc)

Multilingual Search Evaluations

- Conceptually easier if same text is translated into each language as source docs and test searches
    - AKA "Parallel Text"
    - Would need text that has BOTH been widely translated AND doesn't have overly restrictive copyrights
        - Government documents?
        - Older famous books?
        - Even in these cases, gaps need to be tracked and accounted/compensated for
    - Even parallel text Introduces an additional variance due to translation
        - Perhaps used automated tools to translate out to some other language
        - Very imprecise, BUT perhaps this would serve as the basis for "worst case" score?
    - For grid/TREC style testing you could either:
        - Re-use the relevancy assertions / "truth grid" from the native language
            - Rationale: documents about "water usage in California" are still about that subject, regardless of the language
            - Anti-rationale: translation tends to change the meaning of both docs and searches, and therefore doubly so the relevance
        - Ideally the relevancy assertions would be filled in again by native speakers
            - Nice idea, but labor intensive
            - Even within one language it'd be idea to have multiple testers, as discussed above
- What to do if the text is **different** in each language...
    - If still want "absolute truth", some other ideas in best to worst:
        - Try to use similar subject matter
        - Use content and searches from a similar profession or market
        - If using completely different content, try to quantify the differences and correct the results with statistics
    - Give up on grid/truth tests and switch to AB testing. Rationales:
        - Even in the best ideal test cases there could be underlying cultural differences that skew the test
        - Search may fundamentally perform differently in some languages, for example Asian languages that don't have visible word breaks. Or they may be used to certain search engine behaviors intrinsic to search in their language.
        - Different languages imply different markets. Even if a search engine were mathematically proven to be "just as accurate" in another language, that fact might be vastly overridden by other market forces, such as availability of other search engines

Non-Textual Search Evaluations

- Search using Multimedia tags (Ex: Flickr photo tags)
    - vs. Search that looks at the actual bits and bytes
    - User tags (explicit vs implicit)
- Evaluating True Binary Search
    - Types of data
        - Images
        - Audio
        - Video (visual and/or audible sound track)
        - General Binary Data
        - Telemetry, instruments, financial, environmental, genetic sequences
    - How the search is entered
        - Searcher still uses textual descriptions / queries
        - Searcher presents a sample image (or sound clip, etc)
        - Searcher sketches a sample diagram (or hums a sample tune)
        - Searcher iteratively refines a sample or results set (harder to judge via classical grid/TREC style rankings)
- Usage Scenarios (beyond basic search)
    - Clustering (unsupervised)
    - Classification (supervised)
    - Entity Recognition / Extraction

Code Implementation

- Comparing TREC to ORP
    - TREC Code
        - Written in C
        - License Restrictions
    - ORP Code
        - Java
        - Apache License (still *some* restrictions)
- Interface between ORP and Search Engines
    - Search Engines generally don't support native TREC/ORP formats
        - Each engine would need adapters
        - **OR** ORP could interface with native Search Engines' APIs
        - **OR** Use some intermediate format like OpenSearch or other Federated Search technique
    - maybe Lucene Benchmark Package
- Implementing Judgments
    - Grid Methods, borrow from TREC?
        - Classic Precision and Recall
        - MRR, MAP, BPref, NDCG
    - AB Methods
        - Predictive? Constant mixing, TODO: needs expansion

Data Considerations

- General Considerations
    - Character Encoding
    - Simple record files vs. XML
    - Interchange with Excel / OpenOffice / Numbers / Google Docs
    - Interaction w/ Databases...
    - Interaction w/ OpenSearch
    - Version Control
- Specific Entities to Store
    - Sample Documents
    - Searches (AKA TREC Topics)
    - Relevancy Judgments (AKA TREC qrels)
    - AB Preferences (click-throughs, explicit ratings, etc)
    - Controlled Index vs. Federated Search (TODO: explain)
        - Search Engine Results List Formats / APIs
    - Textual vs. Non-Textual Data
        - Corpus, Searches and Judgments
        - *See other section for non-text discussion*