Mid-term Review

Jiepu Jiang
Department of Computer Science, Virginia Tech
2018/10/23
Mid-term Exam

• In class
• When: 10/25 (Thursday), 3:30-4:45pm EST
• Where: the same classroom
• 20% of your total
• Coverage: Lecture 1-18
• Five questions (You don’t need a calculator)
  • One related to indexing, text preprocessing, index search
  • Two related to retrieval models
    • e.g., VSM, BM25, QL, KLD, query expansion, doc expansion
    • Do not include: learning-to-rank, neural nets for ranking
  • One related to test-collection based evaluation & metrics
  • One related to human factors
Mid-term Exam

• In class
• When: 10/25 (Thursday), 3:30-4:45pm EST
• Where: the same classroom
• 20% of your total
• Coverage: Lecture 1-18
• Five questions (You don’t need a calculator)
  • One related to indexing, text preprocessing, index search
  • Two related to retrieval models
    • e.g., VSM, BM25, QL, KLD, query expansion, doc expansion
    • Do not include: learning-to-rank, neural nets for ranking
  • One related to test-collection based evaluation & metrics
  • One related to human factors
Indexing

- The “logical” structure of an inverted index.
- Organized by terms, along with their statistics & posting lists.
- What is a posting list?
  - Frequency posting list: a list of <docid, TF>
  - Positional posting list: a list of <docid, position-in-doc>

<table>
<thead>
<tr>
<th>Term</th>
<th>Term Statistics</th>
<th>Posting List</th>
</tr>
</thead>
<tbody>
<tr>
<td>abandon</td>
<td>DF = 384, CTF = 963</td>
<td>(doc#3, 1) (doc#5, 2) …</td>
</tr>
<tr>
<td>abase</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>retrieve</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Searching Index: document-at-a-time

- Keep a cursor for each posting list (highlighted cells).
- Iteratively:
  - Compare the cursors’ current values
  - Calculate score for the document with the smallest ID
  - Move the smallest cursor(s) to the next position(s)
Searching Index: term-at-a-time

- Keep temporary scores of search results in memory (such as in a hash table).
- Process the posting lists one after another
- Update temporary scores while iterating each posting list.
Index Search

- Doc-at-a-time
  - Keep a cursor for each posting list
  - Compare the current docid of different lists
  - Score each document at a time

- Term-at-a-time
  - Process the posting lists one after another

- Time to process a query
  - # unique terms: how many posting lists to iterate through
  - The length of the posting lists: how many entries are there
  - e.g., longer queries take more time to process, queries with common words take longer to process
O'Neal averaged 15.2 points, 9.2 rebounds and 1.0 assists per game.
Tokenization and query matching

Original text: O’Neal
- o’neal
- oneal
- o neal

Original text: 1,600
- 1,600
- 1600
- 1 600

Possible user queries
- o’neal
- oneal
- o neal

Possible user queries
- 1600
- 1,600
Letter case

Most IR systems ignore letter case differences
• Including Google, Bing, Yahoo, …
• Transforms everything to lowercase

Sometimes letter case may be informative, e.g.,
• the SMART information retrieval system …
• the US health care system …
Stop words removal

Stop words

• words that are ignored during text processing (indexing & searching)
• usually not informative
• usually very frequent (removing them significantly reduces index size)
• usually does not affect much on search effectiveness
  • Exception: “to be, or not to be” (Hamlet, Act III, Scene I, by Shakespeare)
• If you are not sure and can afford the disk space, keep them.

Lucene’s standard English stop words (33 words)

• a, an, and, are, as, at, be, but, by, for, if, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, this, to, was, will, with
Stemming

Purpose

• To regularize words
  • Multiple words will be normalized into the same token
  • Vocabulary size will decrease
  • The length of the posting lists will increase
• Plural \(\rightarrow\) singular, verb (different tenses), adj & adv etc.
• Example: “cats” and “cat”; “retrieve”, “retrieval”, “retrieving”
• Stemming does not always work well

Algorithmic or dictionary-based

• Algorithmic: Porter, snowball
• Dictionary: Krovetz
Examples of stemming “errors”

Overstemming

<table>
<thead>
<tr>
<th>Original</th>
<th>Porter2</th>
<th>Krovetz</th>
</tr>
</thead>
<tbody>
<tr>
<td>organization</td>
<td>organ</td>
<td>organization</td>
</tr>
<tr>
<td>organ</td>
<td>organ</td>
<td>organ</td>
</tr>
<tr>
<td>heading</td>
<td>head</td>
<td>heading</td>
</tr>
<tr>
<td>head</td>
<td>head</td>
<td>head</td>
</tr>
</tbody>
</table>

Understemming

<table>
<thead>
<tr>
<th>Original</th>
<th>Porter2</th>
<th>Krovetz</th>
</tr>
</thead>
<tbody>
<tr>
<td>european</td>
<td>european</td>
<td>europe</td>
</tr>
<tr>
<td>europe</td>
<td>europ</td>
<td>europe</td>
</tr>
<tr>
<td>urgency</td>
<td>urgenc</td>
<td>urgent</td>
</tr>
<tr>
<td>urgent</td>
<td>urgent</td>
<td>urgent</td>
</tr>
</tbody>
</table>
Vector Space Model: A Simple Example

- Consider each indexed term as a unique dimension
- Represent each document as a vector in the $k$-dimensional space
- $\vec{D} = [w_1, w_2, \ldots, w_k]$; for example, $w_i$ can be the frequency of $t_i$ in $D$

$D_2$: cat dog cat

$\vec{D}_2 = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$

We ignore word sequence in this simple VSM example – also called bag-of-words model.
Cosine Similarity: Computation

\[ \vec{x} = [x_1 \ x_2 \ \ldots \ x_k] \]

\[ \vec{y} = [y_1 \ y_2 \ \ldots \ y_k] \]

Dot product

\[ \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} = \frac{\sum_{i=1}^{k} x_i y_i}{\sqrt{\sum_{i=1}^{k} x_i^2} \sqrt{\sum_{i=1}^{k} y_i^2}} \]

Euclidean length
Cosine similarity ignores vector length

\[ \vec{x} = [x_1, x_2, \ldots, x_k] \]

\[ \vec{y} = [y_1, y_2, \ldots, y_k] \]

\[ \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \cdot ||\vec{y}||} = \frac{\vec{x}}{||\vec{x}||} \cdot \frac{\vec{y}}{||\vec{y}||} = \sum_{i=1}^{k} \frac{x_i y_i}{||\vec{x}|| \cdot ||\vec{y}||} \]

Dot product of two unit vectors

unit vectors
Cosine Similarity: IR Computation

\[ \vec{q} = \begin{bmatrix} q_1 & q_2 & \ldots & q_k \end{bmatrix} \]

\[ \vec{d} = \begin{bmatrix} d_1 & d_2 & \ldots & d_k \end{bmatrix} \]

\[ \cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\sum_{i=1}^{k} q_i d_i}{\sqrt{\sum_{i=1}^{k} q_i^2} \sqrt{\sum_{i=1}^{k} d_i^2}} \leq \frac{\sum_{i=1}^{k} q_i d_i}{\sqrt{\sum_{i=1}^{k} q_i^2} \sqrt{\sum_{i=1}^{k} d_i^2}} \]

Independent of ranking

Can iterate through only query terms

It is faster if the index stored the Euclidean length (different from document length).
VSM: TF-IDF Weighting

- \( \vec{D} = [w_1, w_2, \ldots, w_k] \)
- \( w_i \): how important the term \( t_i \) is for representing \( D \)'s information
- How to compute \( w_i \)?
- A popular approach is to use a TF-IDF weighting
- TF: within-document term frequency
- IDF: inverse document frequency

\[
w_i = \text{TF}(t_i, D) \cdot \text{IDF}(t_i)
\]

document-dependent \quad \text{document-independent}
Choices of TF: Binary

- Only consider whether or not a term appears in a document
- Ignore repeated occurrences of the same term
- When to use?
  - Very short documents: repeated occurrence of the same term is rare and unstable (due to the small text sample).
  - e.g. twitter search, passage/sentence retrieval, ...

\[
TF_{\text{binary}}(t_i, D) = \begin{cases} 
1 & c(t_i, D) > 0 \\
0 & c(t_i, D) = 0
\end{cases}
\]

**Notation**

\(c(t, D)\): the number of times \(t\) appears in \(D\) (term frequency).
Choices of TF: Raw Frequency

• Frequent terms in a document are more important (for representing that document’s information).
• The importance of a term is proportional to its frequency in \( D \).
• When to use?
  • Usually not a good idea …
  • Longer document gets higher TF and will be preferred
  • Frequent terms may have too strong influence …

\[
TF_{\text{raw}}(t_i, D) = c(t_i, D)
\]

Notation

\( c(t, D) \): the number of times \( t \) appears in \( D \) (term frequency).
Choices of TF: Log Frequency

• Frequent terms in a document are more important (for representing that document’s information).
• But repeated occurrences of the term will be penalized
  • The first occurrence of a term is the most important.
  • Repeated occurrences are less and less important.
  • A greater log base (b) penalizes the value by a greater extent.

\[
TF_{raw}(t_i, D) = \begin{cases} 
1 + \log_b c(t_i, D) & c(t_i, D) > 0 \\
0 & c(t_i, D) = 0 
\end{cases}
\]

**Notation**

\(c(t, D)\): the number of times \(t\) appears in \(D\) (term frequency).
\[
y = x = \log_1 x
\]

Graph showing:
- \( y = \log_2 x \)
- \( y = \log x \)
- \( y = \log_{10} x \)

TF vs Raw Frequency
Choices of TF: Log Frequency

- Prefers matching more unique terms
  - $c(t_1,d) = 2$, $c(t_2,d) = 1$ is better than $c(t_1,d) = 3$, $c(t_2,d) = 0$

query: **cat** **dog** **cat**

$\vec{q} = \begin{bmatrix} 1.69 & 1 & 0 \end{bmatrix}$

similarity

$D_1$: **cat** **cat** **cat**

$D_2$: **cat** **cat** **dog**

$D_3$: **cat** **dog** **dog**

$t_1$: **cat**

$t_2$: **dog**

$\log 2 \approx 0.69$

$\log 3 \approx 1.10$
Choices of TF: Others

- But raw frequency, log frequency, and binary are the most popular ones.

<table>
<thead>
<tr>
<th>Term frequency</th>
<th>( t_f_{t,d} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>n (natural)</td>
<td>( t_f_{t,d} )</td>
</tr>
<tr>
<td>l (logarithm)</td>
<td>( 1 + \log(t_f_{t,d}) )</td>
</tr>
<tr>
<td>a (augmented)</td>
<td>( 0.5 + \frac{0.5 \times t_f_{t,d}}{\max_t(t_f_{t,d})} )</td>
</tr>
<tr>
<td>b (boolean)</td>
<td>( \begin{cases} 1 &amp; \text{if } t_f_{t,d} &gt; 0 \ 0 &amp; \text{otherwise} \end{cases} )</td>
</tr>
<tr>
<td>L (log ave)</td>
<td>( \frac{1 + \log(t_f_{t,d})}{1 + \log(\text{ave}<em>{t \in d}(t_f</em>{t,d}))} )</td>
</tr>
</tbody>
</table>

Table from the CDM
Choices of IDF: Uniform

- Every term is equally important
- Almost always a bad idea ...
- When to use?
  - I can’t remember any time it worked ...

\[ IDF_{\text{uniform}}(t_i) = 1 \]
Choices of IDF: KSJ

• The original IDF by Karen Spärck Jones (KSJ)
• The log base does not affect ranking (in most retrieval models)
• The total frequency of the term in the corpus does not directly influence IDF (although almost always highly correlated with $n_t$).

\[
IDF_{KSJ}(t) = \log \frac{N}{n_t}
\]

**Notation**

$N$: the total number of documents in the corpus.

$n_t$: the total number of documents contains $t$. 
Choices of IDF: BM25

- The IDF used in BM25 (Thursday)
- +0.5 used for smoothing zero values, no influence in a large corpus
- A greater discounting: very frequency terms have zero weight

\[
IDF_{BM25}(t) = \begin{cases} 
\log \frac{N-n_t + 0.5}{n_t + 0.5} & n_t < \frac{N}{2} \\
0 & n_t \geq \frac{N}{2}
\end{cases}
\]

**Notation**

- \(N\): the total number of documents in the corpus.
- \(n_t\): the total number of documents contains \(t\).
Multivariate Bernoulli Distribution

• Results of $n$ independent Bernoulli trials \{X_1, \ldots, X_n\}
  • $n$ different coins; toss each coin once
  • Each coin may have a different probability of heads/tails

$X_1$: tails  \hspace{1cm} X_2$: heads \hspace{1cm} X_3$: heads \hspace{1cm} X_4$: heads

• Different from a binomial distribution with size $n$
  • Toss the same coin $n$ times (each time independent of others)
A Multivariate Bernoulli Model of Document

- Consider a document $D$ as the outcome of the model
  - Let $\vec{D}$ be a document $D$’s binary term occurrence vector
  - What is the probability of $\vec{D}$ by this MB model?

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>$\vec{D}$</th>
<th>$P(X_i = 1)$</th>
<th>$P(X_i = 0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>1</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>search</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>information</td>
<td>1</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>data</td>
<td>0</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>computer</td>
<td>1</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>science</td>
<td>0</td>
<td>0.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

$$P(D) = P(\text{index}=1) \times P(\text{retrieval}=1) \times P(\text{search}=0) \times P(\text{information}=1) \times P(\text{data}=0) \times P(\text{computer}=1) \times P(\text{science}=0) \times$$

$$= 0.4 \times 0.3 \times 0.5 \times 0.9 \times 0.2 \times 0.9 \times 0.6$$
Naïve Bayes Classification using MB Models

| $X_i$   | $\overrightarrow{D}$ | $P(X_i = 1|IR)$ | $P(X_i = 1|DB)$ |
|---------|-----------------------|-----------------|-----------------|
| index   | 1                     | 0.7             | 0.8             |
| search  | 1                     | 0.9             | 0.9             |
| information | 0                | 0.8             | 0.6             |
| data    | 1                     | 0.5             | 0.9             |
| computer | 0                    | 0.4             | 0.6             |
| relevance | 1                     | 0.9             | 0.1             |
| SQL     | 0                     | 0.1             | 0.8             |

$P(IR) = 0.3$ $P(DB) = 0.7$

$$
\frac{P(IR|D)}{P(DB|D)} = \frac{P(IR)}{P(DB)} \cdot \prod_{X_i \in V} \frac{P(X_i = D_i | IR)}{P(X_i = D_i | DB)}
$$

$$
\frac{P(IR|D)}{P(DB|D)} = \frac{0.3 \cdot 0.7 \cdot 0.9 \cdot 1 - 0.8 \cdot 0.5 \cdot 1 - 0.4 \cdot 0.9 \cdot 1 - 0.1}{0.7 \cdot 0.8 \cdot 0.9 \cdot 1 - 0.6 \cdot 0.9 \cdot 1 - 0.6 \cdot 0.1 \cdot 1 - 0.8} \approx 6.33
$$

$D$ is 6.33 times more likely to be an IR article than a DB one.
Recap: Binary Independence Model

(Robertson & Spärck Jones, 1976)

- IR as interactive and query-dependent binary classification
- Similar to naïve Bayes classification using multivariant Bernoulli models
- Assumes the availability of some user judgments to calculate $R$ and $NR$

\[
\log \frac{P(R \mid D, Q)}{P(NR \mid D, Q)} \Rightarrow \sum_{t_i \in Q \cap D} \log \frac{p_i(1-q_i)}{(1-p_i)q_i}
\]

\[
p_i = P(X_i = 1 \mid R, Q)
\]

\[
q_i = P(X_i = 1 \mid NR, Q)
\]

(Croft & Harper, 1979)

- When user feedback is not available, just ignore $p_i$ and $1 - p_i$
- Because the majority of documents are not relevant, we can just approximate $NR$ using the whole corpus.

\[
\sum_{t_i \in Q \cap D} \log \frac{n_i(1-q_i)}{(1-p_i)q_i} \Rightarrow \sum_{t_i \in Q \cap D} \log \frac{1-q_i}{q_i} \Rightarrow \sum_{t_i \in Q \cap D} \log \frac{N-n_{t_i} + 0.5}{n_{t_i} + 0.5}
\]
Recap: Okapi BM25

BM25 ranks results by the sum of weights for each query term.

\[
\text{score}_{BM25}(d,q) = \sum_{q_i \in q} \text{weight}_{BM25}(d,q_i)
\]

The weight for each query term is a TF-IDF like function

\[
\text{weight}_{BM25}(d,q_i) = \frac{(k_1 + 1) \cdot tf_{(q_i,d)}}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf_{(q_i,d)}} \cdot \log \frac{N - n_i + 0.5}{n_i + 0.5}
\]

- \( tf_{(q_i,d)} \): the raw frequency of \( q_i \) in \( d \)
- \( n_i \): the document frequency of \( q_i \)
- \( N \): the total number of documents in the corpus
- \( dl \): the length of the document \( d \)
- \( avdl \): the average length of documents in the corpus
- \( k_i \) and \( b \) are two parameters
Recap: BM25 parameters

$$weight_{BM25}(d,q_i) = \frac{(k_1 + 1) \cdot tf_{(q_i,d)}}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf_{(q_i,d)}} \cdot \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

$k_1$ determines the upperbound of TF:

$$\lim_{tf \to +\infty} \frac{(k_1 + 1) \cdot tf}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf} = k_1 + 1$$

For an average-length document ($dl = avdl$),

$$TF = \frac{(k_1 + 1) \cdot tf}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf} = \frac{(k_1 + 1) \cdot tf}{k_1 + tf}$$

<table>
<thead>
<tr>
<th>tf (raw term freq)</th>
<th>TF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$+\infty$</td>
<td>$k_1 + 1$</td>
</tr>
</tbody>
</table>
Recap: BM25 parameters

\[
weight_{BM25}(d, q_i) = \frac{(k_1 + 1) \cdot tf_{(q_i,d)}}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf_{(q_i,d)}} \cdot \log \frac{N - n_i + 0.5}{n_i + 0.5}
\]

For a longer-than-average document, raw \( tf \) will be penalized,

\[
TF = \frac{(k_1 + 1) \cdot tf}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf} = \frac{(k_1 + 1) \cdot tf}{k_1 + tf + k_1b \cdot \frac{(dl - avdl)}{avdl}} < \frac{(k_1 + 1) \cdot tf}{k_1 + tf}
\]

For a shorter-than-average document, raw \( tf \) will be boosted,

\[
TF = \frac{(k_1 + 1) \cdot tf}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf} = \frac{(k_1 + 1) \cdot tf}{k_1 + tf + k_1b \cdot \frac{(dl - avdl)}{avdl}} > \frac{(k_1 + 1) \cdot tf}{k_1 + tf}
\]
\[
\frac{tf}{k_1 \left( (1-b) + b \cdot \frac{dl}{avdl} \right) + tf}
\]

\(b=0.75\) (the “default” value)

Term frequency in long documents get penalized!
\[ \frac{tf}{k_1 \left( (1-b) + b \cdot \frac{dl}{avdl} \right) + tf} \]

For long documents (dl=10*avdl)

For longer-than-average documents, TF increases faster for smaller b values.
\[ b \text{ determines how fast TF increases ...} \]

\[ \frac{tf}{k_1 \left( (1 - b) + b \cdot \frac{dl}{avdl} \right) + tf} \]

For short documents \((dl=0.1*avdl)\)

For shorter-than-average documents, TF increases faster for greater b values.
Multinomial Language Model

- A unigram language model $\theta$: a distribution of words
- The vocabulary has $k$ words: $\theta$ is just like a $k$-side dice
- $\theta$ has $k$ parameters:
  - For each of the $k$ words, $P(t|\theta)$
  - Note that $\sum_{t \in V} P(t | \theta) = 1$

| $t$          | $P(t | \theta)$ |
|--------------|-----------------|
| index        | 0.21            |
| retrieval    | 0.32            |
| search       | 0.18            |
| information  | 0.11            |
| data         | 0.06            |
| computer     | 0.04            |
| science      | 0.08            |

Model parameters

Sum up to 1
Multinomial Language Model

- We can consider the text as “generated” from the model.
- For example, a document $D$ has $n$ words.
- $D$ is the outcome of sampling $n$ times from $\theta$.
  - Each time independent of others.
- $D$ ($n=5$): “information retrieval search index retrieval”

$$P(D|\theta) = \prod_{i=1}^{n} P(t_i|\theta)$$

$$= 0.11 \times 0.32 \times 0.18 \times 0.21 \times 0.32$$

| $t$      | $P(t|\theta)$ |
|----------|---------------|
| index    | 0.21          |
| retrieval| 0.32          |
| search   | 0.18          |
| information | 0.11   |
| data     | 0.06          |
| computer | 0.04          |
| science  | 0.08          |
### Naïve Bayes Using Multinomial Models

| $t$          | $c(t,D)$ | $P(t|IR)$ | $P(t|DB)$ |
|--------------|----------|-----------|-----------|
| index        | 1        | 0.21      | 0.17      |
| retrieval    | 2        | 0.32      | 0.05      |
| search       | 1        | 0.18      | 0.22      |
| information  | 1        | 0.11      | 0.12      |
| data         | 0        | 0.06      | 0.33      |
| computer     | 0        | 0.04      | 0.08      |
| science      | 0        | 0.08      | 0.03      |

Prior Prob. $P(IR)=0.3$ $P(DB)=0.7$

$$\frac{P(IR|D)}{P(DB|D)} = \frac{P(IR)}{P(DB)} \cdot \prod_{t \in D} \frac{P(t|IR)^{c(t,D)}}{P(t|DB)^{c(t,D)}}$$

$$\frac{P(IR|D)}{P(DB|D)} = \frac{0.3 \cdot 0.21 \cdot \left(\frac{0.32}{0.05}\right)^2 \cdot 0.18 \cdot 0.11}{0.7 \cdot 0.17 \cdot \left(\frac{0.05}{0.22}\right) \cdot 0.12} \approx 16.26$$

D is 16.26 times more likely an IR article than a DB one.
The Query likelihood model (QL)

- Estimate a language model $\theta_D$ for the document $D$
- For a query $q$, rank documents by $P(q|\theta_D)$
- **QL:** The likelihood of the query $q$ generated by $\theta_D$
- In the case unigram LM, rank results by:

$$P(q | \theta_D) = \prod_{t \in q} P(t | \theta_D)$$

- Or equivalently, $\log P(q | \theta_D) = \sum_{t \in q} \log P(t | \theta_D)$

- Using log probability to avoid overflow …
  - 32bit Float: $\pm3.40282347\times10^{38}$
  - 64bit Double: $\pm1.79769313486231570\times10^{308}$
Recap: The Query likelihood model (QL)

- Each document is generated from a document LM $\theta_D$
- Estimate a language model $\theta_D$ for the document $D$
- Rank documents by $P(q | \theta_D)$
  - Example: QL ranks $D_1$ higher than $D_2$

| D_1’s model | \( P(t | \theta_{D_1}) \) | \( P(t | \theta_{D_2}) \) | \( P(q | \theta_{D_1}) \) |
|----------|-----------------|-----------------|-----------------|
| index    | 0.21            | 0.17            | 0.11 \times 0.32 |
| retrieval| 0.32            | 0.05            |                 |
| search   | 0.18            | 0.22            |                 |
| information | 0.11          | 0.12            |                 |
| data     | 0.06            | 0.33            |                 |
| computer | 0.04            | 0.08            |                 |
| science  | 0.08            | 0.03            |                 |

**D_2’s model**

\[
P(q | \theta_{D_1}) = \prod_{t \in q} P(t | \theta_{D_1}) = 0.11 \times 0.32
\]

\[
P(q | \theta_{D_2}) = \prod_{t \in q} P(t | \theta_{D_2}) = 0.12 \times 0.05
\]
Model Estimation

• The simplest way – Maximum Likelihood Estimation (MLE)

\[
\hat{P}_{MLE}(t | \theta_D) = \frac{c(t, D)}{|D|} = \frac{\text{term frequency}}{\text{document length}}
\]

• An example

\[
\hat{P}_{MLE}(\text{information} | \theta_D) = \frac{2}{12} \quad \hat{P}_{MLE}(\text{retrieval} | \theta_D) = \frac{1}{12}
\]

\[
\hat{P}_{MLE}(\text{is} | \theta_D) = \frac{1}{12} \quad \hat{P}_{MLE}(\text{an} | \theta_D) = \frac{1}{12} \quad \hat{P}_{MLE}(\text{for} | \theta_D) = \frac{1}{12}
\]

\[
\hat{P}_{MLE}(\text{important} | \theta_D) = \frac{1}{12} \quad \hat{P}_{MLE}(\text{technique} | \theta_D) = \frac{1}{12}
\]

D: Information retrieval is an important technique for solving the information overflow problem. | $|D| = 12$ |
In short, in most IR applications, MLE is just counting frequency and divided by the total (sample size) …

- MLE of a document model (multinomial)

\[ \hat{P}_{MLE}(t | \theta_D) = \frac{c(t, D)}{|D|} = \frac{\text{term frequency}}{\text{document length}} \]

- MLE of a corpus model (multinomial)

\[ \hat{P}_{MLE}(t | \theta_{\text{corpus}}) = \frac{\sum_D c(t, D)}{\sum_D |D|} = \frac{\text{corpus term frequency}}{\text{corpus length}} \]

- MLE of a multivariate Bernoulli model of the corpus

\[ \hat{P}_{MLE}(X_i = 1 | \theta_{\text{corpus}}) = \frac{\text{DF}(t_i)}{N} = e^{-\text{IDF}(t_i)} \]
MLE: Recall Several Issues

• Unseen words get zero probability
  • As long as one query term does not appear in $D$, the document gets zero probability.
  • Ranking by $P_{\text{MLE}}(q|D)$ is similar to Boolean AND ...
  • But no occurrence does not mean impossible.

• Limited sample size
  • MLE is reasonably good for a large sample size.
  • But we are estimating a document model, usually just a few hundred/thousand words long.
  • In some cases (will cover next lecture), we also need to estimate a query model. The sample size is even shorter.

• Solution: smoothing (will discuss a few slides later ...)
Jelinek-Mercer Smoothing

• Start simple, but reasonably good
  • Using $P(t | \text{Corpus})$ as the background model
  • Set $\lambda$ to be constant for all documents, independent of any document or query characteristics

• Tune to optimize retrieval performance
  • e.g., maximize mean values of P@10 or AP over a set of different queries in a dataset.
  • optimal value of $\lambda$ varies with different databases, query sets, etc.

• Correctly setting $\lambda$ is very important

$$P(t | \theta_D) = (1-\lambda) \cdot P_{MLE}(t | \theta_D) + \lambda \cdot P(t | \text{Corpus})$$
Jelinek-Mercer Smoothing

- Example: $\lambda = 0.5$, $|D| = 1281$

$$P_{\text{smoothed}} \left( \text{the} \mid \theta_D \right) = 0.5 \times \frac{106}{1281} + 0.5 \times 0.063904$$

<table>
<thead>
<tr>
<th>word</th>
<th>freq</th>
<th>$P_{\text{MLE}}(*) \mid D$</th>
<th>$P(*) \mid \text{Corpus}$</th>
<th>Smoothed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>106</td>
<td>0.082748</td>
<td>0.063904</td>
<td>0.073326</td>
</tr>
<tr>
<td>soviet</td>
<td>18</td>
<td>0.014052</td>
<td>0.000208</td>
<td>0.007130</td>
</tr>
<tr>
<td>chernobyl</td>
<td>10</td>
<td>0.007806</td>
<td>0.000012</td>
<td>0.003909</td>
</tr>
<tr>
<td>disclosure</td>
<td>1</td>
<td>0.000781</td>
<td>0.000053</td>
<td>0.000417</td>
</tr>
<tr>
<td>divert</td>
<td>1</td>
<td>0.000781</td>
<td>0.000014</td>
<td>0.000397</td>
</tr>
<tr>
<td>downplaye</td>
<td>1</td>
<td>0.000781</td>
<td>0.000001</td>
<td>0.000391</td>
</tr>
<tr>
<td>each</td>
<td>1</td>
<td>0.000781</td>
<td>0.000489</td>
<td>0.000635</td>
</tr>
<tr>
<td>early</td>
<td>1</td>
<td>0.000781</td>
<td>0.000486</td>
<td>0.000633</td>
</tr>
</tbody>
</table>
Dirichlet Smoothing

• Problem with Jelinek-Mercer
  • All documents have the same $\lambda$
  • Longer documents provide better estimates (because it provides a larger sample), and thus its own MLE is more reliable

• Make smoothing depend on sample size (adaptive)

• Here $|D|$ is the length of the sample and $\mu$ is a constant

$$P(t \mid \theta_D) = \frac{c(t, D) + \mu \cdot P(t \mid Corpus)}{|D| + \mu}$$

MLE weight: $1 - \lambda = \frac{|D|}{|D| + \mu}$  
Corpus weight: $\lambda = \frac{\mu}{|D| + \mu}$
Dirichlet Smoothing

• Example: $\mu = 500, \ |D| = 1281$

\[
P_{\text{smoothed}}(\text{the} | \theta_D) = \frac{106 + 500 \times 0.063904}{1281 + 500} = 0.077458
\]

| word       | freq | $P_{\text{MLE}}(* | D)$ | $P(* | \text{Corpus})$ | Smoothed   |
|------------|------|--------------------------|------------------------|------------|
| the        | 106  | 0.082748                 | 0.063904               | 0.077458   |
| soviet     | 18   | 0.014052                 | 0.000208               | 0.010165   |
| chernobyl  | 10   | 0.007806                 | 0.000012               | 0.005618   |
| disclosure | 1    | 0.000781                 | 0.000053               | 0.000576   |
| divert     | 1    | 0.000781                 | 0.000014               | 0.000565   |
| downplaye  | 1    | 0.000781                 | 0.000001               | 0.000562   |
| each       | 1    | 0.000781                 | 0.000489               | 0.000699   |
| early      | 1    | 0.000781                 | 0.000486               | 0.000698   |
Smoothing and IDF

- Similar to many retrieval models, we can write QL as:

\[
\text{score}(q, D) = \sum_{t \in q} w(t \mid D), \text{ where } w(t, D) = \log P(t \mid D)
\]

**Dirichlet smoothing:**

\[
P(t \mid D) = \frac{tf + \mu \cdot P(t \mid \text{Corpus})}{|D| + \mu}
\]

**JM smoothing:**

\[
P(t \mid D) = (1 - \lambda) \cdot \frac{tf}{|D|} + \lambda \cdot P(t \mid \text{Corpus})
\]

\[
\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} . \frac{\partial P(t \mid D)}{\partial tf}
\]

**Dirichlet:**

\[
\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} \cdot \frac{1}{|D| + \mu}
\]

**JM:**

\[
\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} \cdot \frac{1 - \lambda}{|D|}
\]

\(tf\) is discrete, but let’s just assume the functions are all continuous here.
Smoothing and IDF

- No matter in which smoothing method is employed, common words get much higher $P_{\text{smoothed}}(t \mid D)$
- The weight (score) of the common words by QL increases much slower than that for less common and rare words while raw $tf$ increases.

$$\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} \cdot \frac{\partial P(t \mid D)}{\partial tf}$$

Dirichlet: $$\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} \cdot \frac{1}{|D| + \mu}$$

JM: $$\frac{\partial w(t, D)}{\partial tf} = \frac{1}{P(t \mid D)} \cdot \frac{1 - \lambda}{|D|}$$

For MLE: $P(t, D) = \frac{tf}{|D|}$, $$\frac{\partial \log P(t, D)}{\partial tf} = \frac{1}{tf}$$, only depends on $tf$

$tf$ is discrete, but let’s just assume the functions are all continuous here.
Dirichlet smoothing, \( \mu = 1500 \).
Long document, \(|D| = 5000\); short document, \(|D| = 100\).
Common word, \(P(t|\text{corpus}) = 0.01\); less common word, \(P(t|\text{corpus}) = 0.0001\).

\[ \log P_{\text{smoothed}} \]

For common words, \(\log P_{\text{smoothed}}\) increases much slower.
Kullback–Leibler (KL) Divergence

• Let $P$ and $Q$ be two probability distributions defined on the same event space, the KL divergence from $Q$ to $P$ is defined as (when $P$ and $Q$ are discrete):

$$KLD(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

• KL-divergence
  • is a distance/similarity function for two probability distributions
  • is asymmetrical, $KLD(P \parallel Q) \neq KLD(Q \parallel P)$
  • is always non-negative
  • gets the minimum value (0) when $P=Q$ (exactly the same)
    • $KLD(P \parallel P) = 0$
The KL-divergence Model

- Recall a language model is a word distribution
- Estimate a document language model $\theta_D$ (similar to QL)
- Also, estimate a query language model $\theta_q$
- Rank documents by $-KLD(\theta_q \| \theta_D)$ (negative KL-divergence)
  - Recall KLD get the minimum value when two distributions are the same (the most similar)
  - Equivalent to ranking by negative cross entropy

$$-KLD(\theta_q \| \theta_D) = - \sum_t P(t \mid \theta_q) \log \frac{P(t \mid \theta_q)}{P(t \mid \theta_D)}$$

$$= \sum_t P(t \mid \theta_q) \log P(t \mid \theta_D) - \sum_t P(t \mid \theta_q) \log P(t \mid \theta_q)$$

A constant in ranking
The KL-divergence Model

• How to estimate a query model is the key!
• In case we use a MLE to estimate the query model, KLD is equivalent to QL

\[ -KLD(\theta_q \parallel \theta_D) = \sum_t P(t \mid \theta_q) \log P(t \mid \theta_D) - \sum_t P(t \mid \theta_q) \log P(t \mid \theta_q) \]

when \( \hat{P}_{MLE}(t \mid \theta_q) = \frac{c(t,q)}{|q|} \),

\[ \sum_t P(t \mid \theta_q) \log P(t \mid \theta_D) \]

\[ = \sum_t \frac{c(t,q)}{|q|} \log P(t \mid \theta_D) = \frac{\sum_t c(t,q) \log P(t \mid \theta_D)}{|q|} = \frac{1}{|q|} \cdot \text{QL} \]
How to improve retrieval?

• Clustering search results
  • Group top-ranked results into different topics
  • Showing only a few results for each topic
  • To avoid that the top-ranked results are biased towards only one particular topic ...

• Using clusters to improve document representation
  • Because a document is (relatively) short ...
  • Document representation is boosted by taken into account the clusters/topics it belongs to
Cluster-based Document Model

• Liu & Croft, SIGIR ’04
• Using k-means for clustering; unigram as features
• Represent a cluster as a language model

\[
P(t \mid \text{cluster}) = \lambda \cdot \frac{c(t, \text{cluster})}{\sum_{i} c(t_i, \text{cluster})} + (1 - \lambda) \cdot P(t \mid \text{corpus})
\]

• Smooth a document \( D \)'s MLE model using
  • The corpus model (the same as QL)
  • The model of the cluster \( D \) belongs to

\[
P(t \mid D) = \lambda_1 P_{\text{MLE}}(t \mid D) + \lambda_2 P(t \mid \text{cluster}) + (1 - \lambda_1 - \lambda_2) \cdot P(t \mid \text{corpus})
\]
LDA-based Document Model

• Wei & Croft, SIGIR ’05
• Similar to the cluster-based document model
• Smooth a document MLE model with
  • The corpus model
  • A mixture model of the document’s topics

\[
P(w \mid D) = \lambda \left( \frac{N_d}{N_d + \mu} P_{ML}(w \mid D) + \left( 1 - \frac{N_d}{N_d + \mu} \right) P_{ML}(w \mid \text{coll}) \right) + (1 - \lambda) P_{lda}(w \mid D)
\]

\[
P_{lda}(w \mid d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^{K} P(w \mid z, \hat{\phi}) P(z \mid \hat{\theta}, d)
\]
Pseudo-relevance Feedback

• Pseudo-relevance feedback (PRF); blind feedback; ...
  • Do an initial search using a regular approach, such as QL
  • Assume the top $k$ ranked results as relevant
  • Perform relevance feedback based on the top $k$ results
  • Normally by query expansion
  • Re-run the query

• A few practical issues
  • The assumption ...
  • Efficiency concerns: expand a short query (2-3 words) into a long one (e.g., ~50 words)
  • Practically effective for improving overall search effectiveness (in terms of the mean values of effectiveness metrics)
  • Our focus today
\textbf{Assumptions}

- A1: $P(D|R)$ is uniform
- A2: $P(t|D, R) = P(t|D)$ and $P(q_i|D, R) = P(q_i|D)$
- A3: $P(t, q|D, R) = P(t|D, R)P(q|D, R)$
- A4: $P(q|D, R) = \prod_{q_i \in q} P(q_i|D, R)$
RM1

RM1: $P(t \mid q, R) \propto P(t, q \mid R) \propto \sum_{D \in \{D_R\}} P(t \mid D) \prod_{q_i \in q} P(q_i \mid D)$

- **Computation**
  - Iterate over each feedback document (source) $D$
  - Assign a weight $P(q \mid D) = \prod_{q_i \in q} P(q_i \mid D)$ to $D$
    - In terms of PRF, we just retrieve top $k$ results by QL and weight each document by QL probability
    - Higher-ranked results get more weights
  - Expand a term $t$ from $D$ by the weight $P(t \mid D)P(q \mid D)$
  - Sum up terms’ weights in each feedback document $D$
  - Normalize the terms’ weights to probability

$$P(t \mid q, R) = \frac{\sum_{D \in \{D_R\}} \left( P(t \mid D) \prod_{q_i \in q} P(q_i \mid D) \right)}{\sum_{t_j} \sum_{D \in \{D_R\}} \left( P(t_j \mid D) \prod_{q_i \in q} P(q_i \mid D) \right)}$$
RM2

RM2: \( P(t \mid q, R) \propto \left( \sum_{D_j} P(t \mid D_j) \right) \cdot \prod_{q_i \in q} \sum_{D \in \{D_R\}} P(q_i \mid D) \frac{P(t \mid D)}{\sum_{D_j} P(t \mid D_j)} \)

- **Computation**
  - Iterate over each query term \( q_i \)
    - Iterate over each feedback document \( D \)
    - Assign a weight \( P(q_i \mid D) \) to \( D \)
    - Expand a term \( t \) from \( D \) by \( P(t \mid D)P(q_i \mid D) \): if both \( t \) and \( q_i \) occur frequently in \( D \), \( t \) gets a greater weight
  - Sum up the weight in each document
  - Multiply the expansion weight for each \( q_i \)
  - Normalize the terms’ weights to probability
Comparing different approaches

- Lv & Zhai, CIKM ’09

<table>
<thead>
<tr>
<th>S.w.</th>
<th>Metric</th>
<th>MLE</th>
<th>RM3</th>
<th>RM4</th>
<th>DMM</th>
<th>SMM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained on AP1 and Tested on AP2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.220</td>
<td>0.295</td>
<td>0.301</td>
<td>0.290</td>
<td>0.304</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.386</td>
<td>0.408</td>
<td>0.418</td>
<td>0.422</td>
<td>0.400</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>3074</td>
<td>3810</td>
<td>3892</td>
<td>3681</td>
<td>3933</td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.231</td>
<td>0.312</td>
<td>0.321</td>
<td>0.289</td>
<td>0.324</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.398</td>
<td>0.436</td>
<td>0.448</td>
<td>0.424</td>
<td>0.432</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>3154</td>
<td>3913</td>
<td>3908</td>
<td>3674</td>
<td>3921</td>
</tr>
<tr>
<td></td>
<td>Trained on TREC6 and Tested on TREC78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.217</td>
<td>0.249</td>
<td>0.242</td>
<td>0.235</td>
<td>0.251</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.437</td>
<td>0.438</td>
<td>0.426</td>
<td>0.443</td>
<td>0.443</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>5114</td>
<td>5805</td>
<td>5739</td>
<td>5476</td>
<td>5821</td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.217</td>
<td>0.251</td>
<td>0.243</td>
<td>0.235</td>
<td>0.252</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.434</td>
<td>0.454</td>
<td>0.446</td>
<td>0.433</td>
<td>0.441</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>5107</td>
<td>5799</td>
<td>5776</td>
<td>5500</td>
<td>5896</td>
</tr>
<tr>
<td></td>
<td>Well-Tuned on WT2G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/</td>
<td>AvgPr</td>
<td>0.293</td>
<td>0.338</td>
<td>0.319</td>
<td>0.327</td>
<td>0.330</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.450</td>
<td>0.500</td>
<td>0.470</td>
<td>0.494</td>
<td>0.496</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>1830</td>
<td>1822</td>
<td>1806</td>
<td>1843</td>
<td>1856</td>
</tr>
<tr>
<td>w/o</td>
<td>AvgPr</td>
<td>0.306</td>
<td>0.344</td>
<td>0.328</td>
<td>0.326</td>
<td>0.331</td>
</tr>
<tr>
<td></td>
<td>Pr@10</td>
<td>0.456</td>
<td>0.490</td>
<td>0.490</td>
<td>0.476</td>
<td>0.476</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>1870</td>
<td>1862</td>
<td>1879</td>
<td>1873</td>
<td>1889</td>
</tr>
</tbody>
</table>
Pseudo-relevance Feedback

• “Usually” believed to be a useful technique

• But somewhat controversial ...
  • Recall oriented; limited improvements in precision at the top
  • Making good queries bad; making bad queries worse
  • “Overall” improvements: average values of metrics?
  • But improving bad/difficult queries may be more important
  • Search efficiency concern
  • Difficult to control; unpredictable for the user

• Difficult to improve in noisy corpus (such as web corpus)
  • Using some clean corpus for query expansion, e.g., Wikipedia
Test Collection Based IR Evaluation

- How to build reliable IR test collections?
  - How many queries, pooling, relevance judgments criteria, etc.
- What are popular search evaluation metrics measuring?
  - P@k, RR, AP, DCG/nDCG, RBP, ERR, TBG, …
- How to draw meaningful conclusions from “scores”?  
  - Significance tests
Building a test collection: Challenges

Corpus

• A set of documents (e.g., news articles, web pages, tweets, medical records)
• Representativeness & coverage; static & dynamic

Queries (information needs)

• Are they real? e.g., sampling queries from real search logs (if you have data)
• How many queries are enough? (Voorhees & Buckley, 2002)

Relevance judgments

• Assessors (Voorhees, 1998)
• Pooling (Zobel, 1998)
• Relevance judgment criteria (Jiang et al., 2017)
Top-k Pooling

Purpose

• To select a subset of documents for relevance judgments
• Why only a subset? Because there are too many documents!
• Ideally, the subset should include a lot of relevant results …
  • Such that we can easily assume unjudged results are non-relevant

Top-k Pooling (TREC standard)

• Assume we have multiple different retrieval systems …
  • Not a big issue in the context of TREC evaluation 😊
• Using each system to retrieve top $k$ results
• Use the union set of top $k$ results from different systems as the pool
• Judge all the results in the pool; unjudged results are non-relevant
Criteria for Relevance Judgments

Binary or multi-level

• Earliest TREC tracks used binary judgments (1 or 0)
• Started to use multi-level judgments since around 2000

TREC web track criteria

• 4: the correct homepage for a navigational query
• 3: key site; authoritative and comprehensive; worth being a top hit
• 2: highly relevant, with a substantial amount of relevant information
• 1: at least minimal relevant information
• 0: not relevant
• Spam or junk (unlikely to be relevant to any query)
Effectiveness Evaluation Measures

Purpose

- To measure the quality of a ranked list of search results
- (Hopefully) a proxy of UX measures (without involving real users)

Some important ideas for designing evaluation metrics

- Relevant > non-relevant
- Multi-level judgments: highly relevant > marginally relevant
- Top-ranked results are more important
  - because users are more likely to view them than lower-ranked ones
- Comparing with perfect ranking lists (as defined by qrels)
- Modeling user behavior (for interacting with a ranked list of results)
Average Precision (AP)

• In many occasions, we prefer one measure.

\[ AP = \frac{1}{|R|} \cdot \sum_{i=1}^{n} Prec(i) \cdot relevance(i) \]

• \(|R|\): total number of relevant documents
• Prec(i): precision at top i documents
• relevance(i): 1 if relevant, otherwise 0
Average Precision (AP): example

Rank = 1, precision = 1
Rank = 3, precision = 2/3
Rank = 6, precision = 3/6
Rank = 9, precision = 4/9
Rank = 10, precision = 5/10

\[ AP = \frac{1}{5} \times (1 + \frac{2}{3} + \frac{3}{6} + \frac{4}{9} + \frac{5}{10}) \]

= the relevant documents

Ranking #1
Average Precision (AP): example

= the relevant documents

Ranking #2

AP=?
Average Precision (AP): example

Rank = 2, precision = 1/2
Rank = 5, precision = 2/5
Rank = 6, precision = 3/6
Rank = 7, precision = 4/7
A relevant result is not retrieved …
• we can consider the retrieved rank=+∞, precision = 0

AP = \frac{1}{5} \times \left( \frac{1}{2} + \frac{2}{5} + \frac{3}{6} + \frac{4}{7} + 0 \right)
Mean Average Precision (MAP)

One of the most frequently used measure today!

Measures a system’s perform on a set of queries.

Computation

• Calculate AP for each individual query $Q_i$
• Calculate the average (mean) of APs on the set of queries $Q$

$$MAP = \frac{1}{|Q|} \cdot \sum_{Q_i \in Q} AP(Q_i)$$
Reciprocal Rank (RR)

Motivation

• Only interested in the rank of the first answer
• Other answers are nice, but don’t matter
• Especially important in web search

\[
RR = \begin{cases} 
\frac{1}{r} & \text{If any results have been retrieved} \\
0 & \text{No relevant results retrieved} 
\end{cases}
\]

\(r\): the rank of the first relevant document in the ranked list.
Mean Reciprocal Rank (MRR)

MRR is the mean RR for a set of queries in the test collection.

Computation

• Calculate AP for each individual query $Q_i$
• Calculate the average (mean) of APs on the set of queries $Q$

\[
MRR = \frac{1}{|Q|} \cdot \sum_{Q_i \in Q} RR(Q_i)
\]
Discounted Cumulative Gain (DCG)

Background

• DCG was proposed around 2000 …
• Most previous metrics only consider binary relevance, e.g., AP and P@k
• P@k emphasizes on top-ranked results in a “rough” way
  • 0-1 cut at rank $k$

DCG: two important contributions

• Taking into account multi-level relevance judgments
• Penalizing lower ranked results in a “smooth” manner

DCG

DCG: notations

• $k$ -- the number of top results to look into
• $i$ -- the rank of a particular result
• $r_i$ -- the relevance grade for the $i$th result

\[
DCG@k = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2 (i + 1)}
\]

Gain function: making a difference between results with different relevant levels

DCG

DCG: notations

• $k$ -- the number of top results to look into
• $i$ -- the rank of a particular result
• $r_i$ -- the relevance grade for the $i$th result

$$DCG@k = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2 (i + 1)}$$

Discounting function: assigning more weights to results at top ranks

Normalized DCG (nDCG)

Purpose

• To scale DCG scores into [0, 1]
• Ideal ranked list: sorting results in qrels by judged relevance scores
• IDCG@k: the DCG@k score for an ideal ranked list
  • IDCG is the upper bound of DCG based on current qrels
• Note that IDCG may be influenced by qrels …

\[
\text{nDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}
\]

nDCG: an Example

System output

- \([D_1, D_2, D_3, D_4, D_5]\)

\[
\text{DCG}@5 = \frac{1}{\log_2 2} + \frac{3}{\log_2 3} + \frac{0}{\log_2 4} + \frac{0}{\log_2 5} + \frac{3}{\log_2 6}
\]

Ideal ranked list

- \([D_2, D_5, D_1, D_6, X]\)

\[
\text{IDCG}@5 = \frac{3}{\log_2 2} + \frac{3}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{0}{\log_2 6}
\]

\[
\text{nDCG}@5 = \frac{\text{DCG}@5}{\text{IDCG}@5}
\]

Relevance Judgments

<table>
<thead>
<tr>
<th>Document</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_1)</td>
<td>1</td>
</tr>
<tr>
<td>(D_2)</td>
<td>2</td>
</tr>
<tr>
<td>(D_3)</td>
<td>0</td>
</tr>
<tr>
<td>(D_4)</td>
<td>0</td>
</tr>
<tr>
<td>(D_5)</td>
<td>2</td>
</tr>
<tr>
<td>(D_6)</td>
<td>1</td>
</tr>
<tr>
<td>(D_7)</td>
<td>0</td>
</tr>
</tbody>
</table>
Evaluation Measures: A Summary

Multi-level relevance judgments
• Highly relevant results are more important
• Many measures use an exponential function $2^{relevance}$

Discounting function
• Penalize the importance of results at lower ranks
• Recent measures all explicitly model how users view search results

Position based model (DCG/nDCG, RBP)
• Discount only depends on the rank of the result

Cascade model (ERR, TBG)
• Discount depends on previous results
• ERR (prev. results’ relevance); TBG (prev. results’ relevance & length)
Paired t-test

Purpose
• Compare means of two related samples
• Two systems’ performance on the same set of queries

An example
• Usually for examine “before-after” effect
• Purpose: is a medication helpful for reducing blood pressure?
• Ask a group of $N$ patients to take the medication
• Two related samples:
  • Blood pressure level prior to the medication
  • Blood pressure level after the medication
  • The two samples are for the same group of patients
• Null hypothesis: no difference (the medication does not work)
Paired t-test: an example

Compare QL and RM3

- Test collection: Robust04 and 249 queries (N = 249)
- Mean P@10: QL: 0.440; RM3: 0.459
- H₀: QL and RM3 yield the same P@10
- H₁: QL and RM3 yield different P@10 (two-tail)
- \( X_D = X_1 - X_2 = 0.019 \)
- Base on the data, \( S_D = 0.115 \)
- Test statistics \( t = \frac{\bar{X}_D}{\frac{S_D}{\sqrt{N}}} = 2.642 \)
- Degree of freedom = \( N - 1 = 248 \)
- \( P( |t| = 2.642 \mid DF = 248 ) = 0.009 < 0.01 \)
- Reject H₀ & adopt H₁
- Usually use * for \( p < 0.05 \), ** for \( p < 0.01 \), and *** for \( p < 0.001 \)
- Reporting exact \( p \) values is often unnecessary
Different Types of CHI-IR Studies and Data

Purposes

- Observational, e.g., characterizing, formulate initial hypotheses
- Experimental, e.g., testing & verifying hypotheses, evaluation

<table>
<thead>
<tr>
<th></th>
<th>Observational</th>
<th>Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Studies</td>
<td>In-lab behavior observations</td>
<td>In-lab controlled tasks, comparison of systems</td>
</tr>
<tr>
<td><em>Controlled interpretation of behavior with detailed instrumentation</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field Studies</td>
<td>Ethnography, case studies, panels (e.g., Nielsen)</td>
<td>Clinical trials and field tests</td>
</tr>
<tr>
<td><em>In the wild, ability to probe for detail</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log Studies</td>
<td>Logs from a single system</td>
<td>A/B testing of alternative systems or algorithms</td>
</tr>
<tr>
<td><em>In the wild, little explicit feedback but lots of implicit signals</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Different types of user data in HCl research.

### Different Types of CHI-IR Studies and Data

#### Settings
- Lab studies: artificial situation created & controlled by researchers
- Field studies: look into a real situation; much less obtrusive
- Log studies: large-scale real user data; accessibility of logs 🤔

<table>
<thead>
<tr>
<th></th>
<th>Observational</th>
<th>Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lab Studies</strong></td>
<td>Controlled interpretation of behavior with detailed</td>
<td>In-lab behavior observations</td>
</tr>
<tr>
<td></td>
<td>instrumentation</td>
<td>In-lab controlled tasks, comparison of systems</td>
</tr>
<tr>
<td><strong>Field Studies</strong></td>
<td>In the wild, ability to probe for detail</td>
<td>Ethnography, case studies, panels (e.g., Nielsen)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clinical trials and field tests</td>
</tr>
<tr>
<td><strong>Log Studies</strong></td>
<td>In the wild, little explicit feedback but lots of</td>
<td>Logs from a single system</td>
</tr>
<tr>
<td></td>
<td>implicit signals</td>
<td>A/B testing of alternative systems or algorithms</td>
</tr>
</tbody>
</table>

Table 1. Different types of user data in HCI research.

Experiment Design: Basics

Several Steps

• Formulate a hypothesis: X influences Y (in a particular way)
  • May come from past experiments, data, literature, etc.
• Identify independent variables (X) and different levels
  • You should be able to control X in your experiments
  • Avoid having too many independent variables in one study
  • Focus on only a few IVs that are most related to your research question
• Identify dependent variables (Y)
  • Should be measurable
  • Should indicate the expected outcome of the study
  • Example: the purpose is to measure user’s search performance
  • Acceptable DVs: user feedback, number of clicks, time takes to finish, etc.
  • Unacceptable DVs: query length (may not indicate performance)
Factorial Design

Factors (Independent Variables)

• Frequently examined factors in interactive IR studies
  • e.g., system, task type, gender, users’ knowledge level, language fluency, ...
• Usually you are either interested in studying the factor, or the factor is very important such that you cannot ignore in the experiment
  • e.g., task type is usually a dominating factor for results; gender is okay

Condition

• A particular setting of the factors (IVs)
• Avoid having too many IVs and conditions
Experiment Design: Concerns

Internal Validity

• Are observed results actually caused by the independent variables?
• Ordering effects: the sequence of the conditions may influence results
  • Possible reasons: learning, fatigue
• Selection bias, e.g., the ways of recruiting subjects make them biased, the experiment conditions are not properly randomized, etc.
• Experimenter bias, e.g., you expect users to prefer a particular functionality and thus constantly remind them to use it during the experiment

External Validity

• Can observed results be generalized to the world outside the lab?
• Are the subjects representative of the target population?
• Are the lab experiment conditions & tasks realistic?
• ...
Within & Between Subjects Design

Within Subjects Design

- Each subject (participant) tests all experiment conditions
- Results are compared within each user (to eliminate the effects of individuals)
- But the results are likely to be influenced by order effects

Between Subjects Design

- Each subject (participant) tests only one condition.
- Do we observe differences because the two groups of users are different?

A within subjects design: each user works on all three systems (conditions)

<table>
<thead>
<tr>
<th>User</th>
<th>Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>A, B, C</td>
</tr>
<tr>
<td>User 2</td>
<td>B, C, A</td>
</tr>
<tr>
<td>User 3</td>
<td>C, A, B</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

A between subjects design: users work on different systems (conditions)

<table>
<thead>
<tr>
<th>User</th>
<th>Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>A</td>
</tr>
<tr>
<td>User 2</td>
<td>B</td>
</tr>
<tr>
<td>User 3</td>
<td>C</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Within vs. Between: When to use?

Within Subjects Design

• Pros: eliminates the differences between people; usually needs a smaller sample to test an effect (because use dependent sample stat methods)
• Cons: ordering effects (can counterbalance), e.g., learning and fatigue

Between Subjects Design

• (Usually) easy to conduct; requires substantially more samples (related to statistical tests; will come back later)

Many practical concerns …

• How many conditions do you have? Sometimes we have to use a between subjects design or a mixed one simply because we have too many conditions (such that it is impractical to ask a user to test all of them).

• Can we precisely control users and experiments? Many online & crowdsourcing experiments use between subjects design because they have no way to make sure subjects (by themselves) can precisely follow a within subjects design experiment protocol to complete the experiment.
Counterbalancing

Purpose

• To reduce the influence of ordering effects by varying the order of conditions systematically (not randomly)

Latin Square

• In combinatorics and in experimental design, a Latin square is an n x n array filled with n different symbols, each occurring exactly once in each row and exactly once in each column.

• A Latin square is said to be reduced (also, normalized or in standard form) if both its first row and its first column are in their natural order.

<table>
<thead>
<tr>
<th>User</th>
<th>Sequence</th>
<th>User</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>A -&gt; B -&gt; C</td>
<td>User 4</td>
<td>A -&gt; B -&gt; C</td>
</tr>
<tr>
<td>User 2</td>
<td>B -&gt; C -&gt; A</td>
<td>User 5</td>
<td>B -&gt; C -&gt; A</td>
</tr>
<tr>
<td>User 3</td>
<td>C -&gt; A -&gt; B</td>
<td>User 6</td>
<td>C -&gt; A -&gt; B</td>
</tr>
</tbody>
</table>
Institutional Review Board (IRB)

The very first thing to do!
• Almost any studies involving human subjects require IRB approval.
• Note that it takes time (usually at least a month)

Common IRB concerns on IR studies
• IR studies are usually low-risk, but are not non-risk
• Privacy issues, e.g.:
  • What data is collected?
  • Is it possible to identify the participant based on the collected data?
  • In which way you are going to maintain and share the data
• Topics & contents may cause uncomfortable feelings
  • e.g., search tasks related to very sad, controversial, or offensive topics, such as 911, abortion, gender or racial discrimination, etc.
• Risks of some apparatus
  • Some own-made wearable devices?
  • Popular devices are quite safe, e.g., EGG, fMRI
Correlation Analysis

Correlation between X and Y
• How does Y change with the changes of X?
• Usually used to form up hypothesis; difficult to draw conclusions

Scatter plot examples:

A positive correlation

A negative correlation
Correlation: direction and strength

- $r$ ranges from -1 to +1
- $r$ quantifies both the strength and direction of a linear relationship
- Strength (the absolute value): how closely the points follow a straight line
- Direction (the sign): positive or negative
Interpreting Pearson’s Correlation

• Pearson’s r is a linear correlation coefficient
  • When x changes, to which extent y changes by a proportional magnitude?
• Strong, moderate, weak correlation …
  • No gold standard for the threshold; depends on the research community

| |r| value | Interpretation |
|---|---|---|
| 0.8 ~ 1.0 | Very strong relationship |
| 0.6 ~ 0.8 | Strong relationship |
| 0.4 ~ 0.6 | Moderate relationship |
| 0.2 ~ 0.4 | Weak relationship |
| 0.0 ~ 0.2 | Weak or no relationship |

• Examples of reporting correlation
  • We observed a significant strong positive correlation between the number of clicks in a session and user satisfaction ratings ($r = 0.73$, $p < 0.01$).
Correlation ≠ Causality

• Correlation does not necessarily indicate a causality between the two variables.
• Two variables can be correlated because of a common cause.
• Correlation only looks into two variables; does not take into account other factors.
Regression Analysis: Differences

LR in behavioral analysis

- Quantify the relationship between IVs and DV
  - Looking into the value and sign of the coefficients \( (b) \), and \( p \) values
  - With other variables being controls (comparing to correlation)
- Can be predictive (but usually not the primary focus)
  - You can probably find better prediction models in ML …
- For example:
  - Does an IV have a significant positive effect on the DV?

LR in machine learning

- Prediction & performance
- For example:
  - How accurate is my prediction model?
Linear Regression: Model Fitness

$R^2$ (R-squared; Coefficient of determination)

- Quantifies how well you model explains the data
  - Compared with a baseline model that explains just using the mean of $y$
- (Practically) ranges from 0 to 1; below 0?
  - 1: (perfectly explains the data)
  - 0: (cannot explains the data)

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]

![Graphs showing R-squared values 0.81 and 0.24](image)
Linear Regression: Adjusted $R^2$

$R^2$
- Quantifies how well you model explains the data
- But $R^2$ will be inflated when the number of IVs increase

Adjusted $R^2$
- Takes into account the number of IVs (predictors)
- Each additional IV should make a large enough contribution in $R^2$
  - Otherwise adjusted $R^2$ get penalized
- $p$: the number of IVs (does not include the bias term);
- $n$: the number of data points (sample size)

$$\text{Adjusted } R^2 = 1 - \left(1 - R^2\right) \cdot \frac{n - 1}{n - p - 1} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \cdot \frac{n - 1}{n - p - 1}$$
Linear Regression: Coefficients (example)

- The influence of the four factors on usefulness
  - \( \text{Usef} = 0.42 \text{ nov} + 0.12 \text{ effort} + 0.15 \text{ under} + 0.32 \text{ relia} - 0.34 \)
  - Interpret the effect of Novelty on Usef
    - \( b > 0 \) and \( p << 0.001 \); our model suggests a significant positive effect of novelty on usefulness
    - \( b = 0.42 \); while other conditions being equal, our model suggests that one unit increase in user’s novelty rating increases the usefulness rating by 0.42

---

**Linear regression model (robust fit):**

\[
\text{usef} \sim 1 + \text{nov} + \text{effort} + \text{under} + \text{relia}
\]

**Estimated Coefficients:**

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
<th>tStat</th>
<th>pValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>-0.34171</td>
<td>0.32334</td>
<td>-1.0568</td>
<td>0.29094</td>
</tr>
<tr>
<td>nov</td>
<td>0.42583</td>
<td>0.035567</td>
<td>11.972</td>
<td>2.7355e-30</td>
</tr>
<tr>
<td>effort</td>
<td>0.12231</td>
<td>0.049757</td>
<td>2.4581</td>
<td>0.014199</td>
</tr>
<tr>
<td>under</td>
<td>0.15881</td>
<td>0.045018</td>
<td>3.5276</td>
<td>0.00044554</td>
</tr>
<tr>
<td>relia</td>
<td>0.32186</td>
<td>0.041967</td>
<td>7.6692</td>
<td>5.5306e-14</td>
</tr>
</tbody>
</table>
ANOVA

• Limitation of t-test
  • Only compare two groups’ means
  • Only one factor; no factorial design

• Example
  • Ask users to use three systems in two different types of tasks
    • Rate their satisfaction after each condition
  • Research questions
    • Does the choice of search system influence user satisfaction?
    • Does the type of task influence user satisfaction?
  • Difficult to draw conclusions using t-test

<table>
<thead>
<tr>
<th></th>
<th>Lucene</th>
<th>Galago</th>
<th>Indri</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task 1</strong></td>
<td>3.40</td>
<td>3.72</td>
<td>3.65</td>
</tr>
<tr>
<td><strong>Task 2</strong></td>
<td>3.56</td>
<td>3.48</td>
<td>3.82</td>
</tr>
</tbody>
</table>
One-way ANOVA

• Only one factor
  • $k$ groups, e.g., ($k = 3$) Lucene vs. Galago vs. Indri
  • We ask $N$ users to use and rate the three systems
  • We do not look into task difference so far
  • We observe the mean ratings as follows

<table>
<thead>
<tr>
<th></th>
<th>Lucene</th>
<th>Galago</th>
<th>Indri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.40</td>
<td>3.82</td>
<td>3.60</td>
</tr>
</tbody>
</table>

• Null hypothesis
  • The three systems have no difference (mean is the same for all)

• Alternative hypothesis
  • At least one system is different.
  • Do not assume particular relationship.
  • Research question: whether the choice of system matters.
One-way ANOVA

• The Idea
  • Examine where the variance of the observations comes from
  • Between group: the variance for data from different conditions
    • Quantify the differences caused by the factor and chance
  • Within group: the variance for data in the same condition
    • Quantify the differences caused by chance (e.g., individual difference)
  • The test statistics looks into the between and within-group variances

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between</td>
<td>91.476</td>
<td>2</td>
<td>45.733</td>
<td>4.467</td>
<td>.021</td>
</tr>
<tr>
<td>Within</td>
<td>276.400</td>
<td>27</td>
<td>10.237</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>367.867</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
One-way ANOVA

• Results & Interpretation
  • F-test gets \( p = 0.021 < 0.05 \) (the threshold)
  • Reject the null hypothesis
  • Conclude that at least one system is different from others
    • Indicates that the choices of systems do influence user satisfaction
  • But we don’t know how exactly the three systems differ
    • Run post-hoc tests, e.g., Turkey’s HSD

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between</td>
<td>91.476</td>
<td>2</td>
<td>45.733</td>
<td>4.467</td>
<td>.021</td>
</tr>
<tr>
<td>Within</td>
<td>276.400</td>
<td>27</td>
<td>10.237</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>367.867</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Two-way ANOVA

• **Interpretation and results**
  
  • Similar to one-way ANOVA
  
  • Run F-test, draw conclusion based on p-value
    
    • The choice of task does not affect user satisfaction
    
    • The choice of system does affect
    
    • Interaction: does affect

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F-Value</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>2</td>
<td>8.222</td>
<td>4.1111</td>
<td>1.71</td>
<td>0.2094</td>
</tr>
<tr>
<td>System</td>
<td>2</td>
<td>20.222</td>
<td>10.1111</td>
<td>4.20</td>
<td>0.0318</td>
</tr>
<tr>
<td>Task * System</td>
<td>4</td>
<td>46.222</td>
<td>11.5556</td>
<td>4.80</td>
<td>0.0082</td>
</tr>
<tr>
<td>Error</td>
<td>18</td>
<td>43.333</td>
<td>2.4074</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>26</td>
<td>118.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note that ...

- Note that many papers may not report the statistics in the same format you learned from your statistics courses …
  - Probably because of page limits in conference papers 😊

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F-Value</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>2</td>
<td>8.222</td>
<td>4.1111</td>
<td>1.71</td>
<td>0.2094</td>
</tr>
<tr>
<td>System</td>
<td>2</td>
<td>20.222</td>
<td>10.1111</td>
<td>4.20</td>
<td>0.0318</td>
</tr>
<tr>
<td>Task * System</td>
<td>4</td>
<td>46.222</td>
<td>11.5556</td>
<td>4.80</td>
<td>0.0082</td>
</tr>
<tr>
<td>Error</td>
<td>18</td>
<td>43.333</td>
<td>2.4074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>26</td>
<td>118.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Kelly et al. (2009)

- Results: mean & standard deviation of ratings for each condition

Table 4. Mean (standard deviation) for Exit Questionnaire items according to condition (TS=term suggestions; QS=query suggestions; SGS=system-generated suggestions; UGS=user-generated suggestions) (higher values in bold).

<table>
<thead>
<tr>
<th>Questionnaire Item</th>
<th>TS</th>
<th>QS</th>
<th>SGS</th>
<th>UGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The suggestions made things I wanted to accomplish easier to do.</td>
<td>3.24</td>
<td>3.58</td>
<td>3.20</td>
<td>3.66</td>
</tr>
<tr>
<td>2. The suggestions helped me modify my queries.</td>
<td>2.75</td>
<td>2.67</td>
<td>3.17</td>
<td>2.16</td>
</tr>
<tr>
<td>3. The suggestions helped me think of new approaches to searching.</td>
<td>2.49</td>
<td>3.24</td>
<td>2.42</td>
<td>3.40</td>
</tr>
<tr>
<td>4. The suggestions helped me better understand the topics.</td>
<td>2.65</td>
<td>3.29</td>
<td>2.72</td>
<td>3.28</td>
</tr>
<tr>
<td>5. The suggestions were easy to use.</td>
<td>3.02</td>
<td>2.75</td>
<td>3.23</td>
<td>2.46</td>
</tr>
<tr>
<td>6. It was easy to understand how the suggestions related to the search topics.</td>
<td>3.40</td>
<td>2.75</td>
<td>3.08</td>
<td>3.06</td>
</tr>
<tr>
<td>7. The quality of the suggestions was good.</td>
<td>3.51</td>
<td>3.27</td>
<td>3.63</td>
<td>3.10</td>
</tr>
<tr>
<td>8. It was easy to find relevant documents with the system.</td>
<td>2.91</td>
<td>3.51</td>
<td>3.13</td>
<td>3.30</td>
</tr>
<tr>
<td>9. It was easy to understand why some documents were retrieved in response to my queries.</td>
<td>2.44</td>
<td>2.76</td>
<td>2.55</td>
<td>2.66</td>
</tr>
<tr>
<td>10. Overall, the system was effective in helping me complete the search tasks.</td>
<td>2.91</td>
<td>3.31</td>
<td>3.15</td>
<td>3.06</td>
</tr>
<tr>
<td>11. Overall, I was satisfied with my performance.</td>
<td>3.13</td>
<td>3.44</td>
<td>3.43</td>
<td>3.10</td>
</tr>
</tbody>
</table>

Example: Kelly et al. (2009)

- Results: effects of the two factors

<table>
<thead>
<tr>
<th>Item</th>
<th>Type (Terms or Queries)</th>
<th>Source (System or Users)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F(1,108) = 2.29, p=.133, \text{ns}$</td>
<td>$F(1,108) = 3.88, p&lt;.05, \text{U&gt;S}$</td>
</tr>
<tr>
<td>2</td>
<td>$F(1,108) = .115, p=.735, \text{ns}$</td>
<td>$F(1,108) = 21.72, p&lt;.01, \text{S&gt;U}$</td>
</tr>
<tr>
<td>3</td>
<td>$F(1,108) = 10.75, p&lt;.01, \text{Q&gt;T}$</td>
<td>$F(1,108) = 18.48, p&lt;.01, \text{U&gt;S}$</td>
</tr>
<tr>
<td>4</td>
<td>$F(1,108) = 7.34, p&lt;.01, \text{Q&gt;T}$</td>
<td>$F(1,108) = 5.63, p&lt;.05, \text{U&gt;S}$</td>
</tr>
<tr>
<td>5</td>
<td>$F(1,108) = 2.04, p=.156, \text{ns}$</td>
<td>$F(1,108) = 15.59, p&lt;.01, \text{S&gt;U}$</td>
</tr>
<tr>
<td>6</td>
<td>$F(1,108) = 9.45, p&lt;.01, \text{T&gt;Q}$</td>
<td>$F(1,108) = .011, p=.917, \text{ns}$</td>
</tr>
<tr>
<td>7</td>
<td>$F(1,108) = 1.31, p=.254, \text{ns}$</td>
<td>$F(1,108) = 6.48, p&lt;.05, \text{S&gt;U}$</td>
</tr>
<tr>
<td>8</td>
<td>$F(1,108) = 10.18, p&lt;.01, \text{Q&gt;T}$</td>
<td>$F(1,108) = .705, p=.403, \text{ns}$</td>
</tr>
<tr>
<td>9</td>
<td>$F(1,108) = 3.38, p=.069, \text{ns}$</td>
<td>$F(1,108) = .361, p=.549, \text{ns}$</td>
</tr>
<tr>
<td>10</td>
<td>$F(1,108) = 3.84, p&lt;.05, \text{Q&gt;T}$</td>
<td>$F(1,108) = .168, p=.682, \text{ns}$</td>
</tr>
<tr>
<td>11</td>
<td>$F(1,108) = 3.17, p=.078, \text{ns}$</td>
<td>$F(1,108) = 3.05, p=.084, \text{ns}$</td>
</tr>
</tbody>
</table>