INFSCI 2140
Information Storage and Retrieval
Lecture 5: Retrieval Evaluation

Peter Brusilovsky
http://www2.sis.pitt.edu/~peterb/2140-061

The issue of evaluation: TREC

- Text RETrieval Conferences organized by NIST
  - http://trec.nist.gov/
- Annual TREC IR “competitions”
  - Expanding set of “tracks”
  - Standard document sets
  - Standard queries and “topics”
Evaluation: Macro view

Life Cycle of an Information System

Evaluation: Micro view
Effectiveness

- The effectiveness of a retrieval system is related to the user satisfaction – i.e. is related to the ectosystem

![Diagram showing the relationship between information need, query, and results.]

How Good is the Model?

- Was the query language powerful enough to represent the need?
- Were we able to use query syntax to express what we need – Operators – Weights
- Were the words from the limited vocabulary expressive enough?
Relevant, Pertinent, Useful…?

- **Relevance**
  - how well the documents respond to the query

- **Pertinence:**
  - how well the documents respond to the information need

- **Usefulness (vs. relevance)**
  - Useful but not relevant
  - Relevant but useless

How can we measure it?

- **Binary measure (yes/no)**
- **N-ary measure:**
  - 3 very relevant
  - 2 relevant
  - 1 barely relevant
  - 0 not relevant

- $N = ?$: consistency vs. expressiveness
Precision and Recall (Diagram)

Precision and Recall (Table)

<table>
<thead>
<tr>
<th>Relevant</th>
<th>Not retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>Not retrieved</td>
</tr>
<tr>
<td>w</td>
<td>x</td>
</tr>
<tr>
<td>y</td>
<td>z</td>
</tr>
</tbody>
</table>

- Relevant = $n_1 = w + x$
- Retrieved = $n_2 = w + y$

- Precision: $P = \frac{w}{\text{Retrieved}}$
- Recall: $R = \frac{w}{\text{Relevant}}$
**Precision and Recall**

Precision = \( \frac{w}{n_2} = \frac{w}{w + y} \)

Recall = \( \frac{w}{n_1} = \frac{w}{w + x} \)

- Number of retrieved documents that are relevant
- Number of retrieved documents
- Number of relevant documents

**How are they related?**

- Suppose that the system is running in response to a query and Recall and Precision are measured as increasing number of documents are retrieved.
  - At the beginning imagine that only one document is retrieved and that it is relevant:
    - Precision = 1
    - \( \text{Recall} = \frac{1}{n_1} \)
    - Very low
How are they related?

- On the other extreme suppose that every document in the database is retrieved:

  \[
  \text{Precision} = \frac{n_2}{N} \\
  \text{Recall} = 1
  \]

  \begin{itemize}
  \item Very low
  \item All relevant documents are retrieved
  \item Total number of documents in the collection
  \end{itemize}

How are they related?

- Precision falls and recall rises as the number of documents retrieved in response to a query is increased.
- The number of returned documents can be considered as a search parameter.
- Changing it we can build a precision/recall graphs.
### Precision-Recall Graph

<table>
<thead>
<tr>
<th>Rel./notRel</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>3</td>
<td>0.66666667</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>0.75</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.66666667</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>0.57142857</td>
<td>0.8</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>9</td>
<td>0.44444444</td>
<td>0.8</td>
</tr>
<tr>
<td>10</td>
<td>0.4</td>
<td>0.8</td>
</tr>
<tr>
<td>11</td>
<td>0.36363636</td>
<td>0.8</td>
</tr>
<tr>
<td>12</td>
<td>0.33333333</td>
<td>0.8</td>
</tr>
<tr>
<td>13</td>
<td>0.38461538</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0.35714286</td>
<td>1</td>
</tr>
</tbody>
</table>

- Imagine that a query is submitted to the system.
- 14 documents are retrieved
- 5 of them are relevant
- These 5 are also the total number of relevant documents in the collection

### Precision and Recall Graphs
Precision Graph

- Precision falls when more and more documents are retrieved.
- Note sawtooth shape!

Recall Graph

- Recall when more and more documents are retrieved.
- Note terraced shape!
Precision-Recall Graph

- Sequences of points \((p, r)\)
- Similar to \(y = 1 / x\):
  - Inversely proportional!
- Sawtooth shape
- Use smoothed graphs
- How we can compare different IR systems using precision-recall graphs?
Precision-Recall Graph

- The system a has the best performances, but what about system b and c, which one is the best?

Fallout

- The proportion of not relevant document that are retrieved (it should be low for a good IR system)
- Fallout measures how well the system filters out not-relevant documents

\[
F = \frac{y}{N - n_1}
\]

- Number of not relevant documents that are retrieved
- Total number of not relevant documents
Generality

- Proportion of relevant documents in the collection. It is more related to the query rather than to the retrieval process.

\[ G = \frac{n_1}{N} \]

Number of relevant documents in the collection

Total number documents

Exercise 1

Imagine that an IR system retrieved 10 documents in answer to a query, but only the document number 1, 3, 5, 7 are relevant.

Calculate Precision, Recall and Fallout considering that there are other 6 relevant documents that were not retrieved and that the total number of documents in the collection is 100 (included the 10 retrieved).
Problems of recall & precision

- Hard to find recall
- Neither shows effectiveness
  - Comparing the graphs
  - F-measure
  - Relative performance as another single measure
- Recall & precision may not be important for the user

Problems with Recall

- Precision can be determined exactly
  \[
  \text{Precision} = \frac{\text{# of relevant docs retrieved}}{\text{# of retrieved docs}}
  \]
- Recall cannot be determined exactly because it requires the knowledge of all of relevant documents in the collection. Recall can only be estimated
  \[
  \text{Recall} = \frac{\text{# of relevant docs retrieved}}{\text{# of relevant docs}}
  \]
The Need for a Single Measure

- To compare two IR systems it would be nice to use just one number, and precision and recall are
  - Related to each other
  - Give an incomplete picture of the system
- F-Measure (not fallout!)
  - $F = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$
  - Combines recall and precision in a single efficiency measure (*harmonic mean* of precision and recall)

Relative Performance

- $R = \frac{P}{1 - P}$
  - Relevant to non-relevant retrieved
- $F = \frac{G}{1 - G}$
  - Relevant to non-relevant in the collection
- $R/F$ - relative performance
Relative Performance

- Relative performance should be greater than one if we want that the system does better in locating relevant documents than it does rejecting not-relevant ones

\[
\frac{R}{F} = \frac{\frac{P}{1 - P}}{\frac{G}{1 - G}} > 1
\]

Precision and Recall: User View

- It is not clear how important they are for the users:
  - Precision is usually more important than recall, because users appreciate outputs that do not contain not-relevant documents
  - This, of course, depends on the kind of user: high recall is important for an attorney that needs to determine all the legal precedents to a case.
What does the user want?
Restaurant case

- The user wants to find a restaurant serving Sashimi. She can use 2 IR systems. How we can say which one is better?

User - oriented measures

- Coverage ratio:
  \[
  \text{known\_relevant\_retrieved} / \text{known\_relevant}
  \]

- Novelty ratio:
  - \(\text{new\_relevant} / \text{Relevant}\)

- Relative recall
  - \(\text{relevant\_retrieved} / \text{wants\_to\_examine}\)

- Recall Effort:
  - \(\text{wants\_to\_examine} / \text{had\_to\_examine}\)
Coverage and Novelty

- **Coverage Ratio**: proportion of relevant documents known to the user that are actually retrieved
  - A high coverage ratio would give to the user some confidence that the system is locating all the relevant documents

- **Novelty Ratio**: proportion of relevant retrieved documents that were unknown to the user
  - A high novelty ratio suggests that the system is effective in locating documents previously unknown to the user

For example if the user knows that there are 16 relevant documents (but they are not all the relevant documents) and the system retrieve 10 relevant documents included 4 of those that the user knows we have:

- Coverage ratio = \[
\frac{4}{16}
\]
- Novelty ratio = \[
\frac{6}{10}
\]

User may expect 40 relevant documents in total
Relative Recall and Effort

- **Relative Recall**: The ratio of relevant retrieved documents examined by the user to the number of documents the user would have liked to examine
  - If the system has retrieved 5 relevant documents among 20 - how large is the relative recall?

- **Relative Effort**: The ratio of number of relevant documents desired to the number of documents examined by the user to find the number of relevant documents desired
  - this ratio go to 1 if the relevant docs are the first examined, to early 0 if the user would need to examine hundreds of documents to find the desired few.

What happen when we increase the number of documents retrieved?

- **At low retrieval volumes** when we increase the number of documents retrieved, the number of relevant documents increase more rapidly than the number of not relevant documents
What happens when we increase the number of documents retrieved?

- At **high retrieval volumes** when we increase the number of documents retrieved, the situation is reversed.

From Query to System Performance

- Precision and Recall change with the retrieval value.
- Averaging the values obtained might provide an adequate measure of the effectiveness of the system.
- To evaluate system performance, we compute average precision and recall.
Three Points Average

- Fix recall and count precision!
- For a given query three points average precision is computed by **averaging the precision** of the retrieval system at three recall levels, typically:
  - 0.25 0.5 0.75
  - or
  - 0.2 0.5 0.8
- Same can be done for recall

Other Averages

- For a given query **eleven points average precision** is computed by averaging the precision of the retrieval system at eleven recall levels
  - 0.0 0.1 0.2 ... 0.9 1.0
- If finding exact recall points is hard, it is done at different levels of document retrieval
  - 10, 20, 30, 40, 50... relevant retrieved documents
Expected Search Length

■ Definition
  - a way to estimate the number of documents that a user have to read in order to find the desired number of relevant documents.
  - M to examine to find N relevant

■ Calculation

■ Graphing

■ Average search length

Taking the Order into Account

■ Results of search is not a set, but a sequence
■ Recall and Precision fail to take into account the ordering of the retrieval results
■ Two documents that contains the same information can be judged by the system in a different way
  - the first in the list is considered relevant
  - the second one, maybe separated from the first by many other documents, is considered much less relevant
Frustration

- Two systems can give a very different perception if they just organize the same documents in a different way:

| All the relevant documents in the first positions | Relevant documents scattered in the list at the end of the list |

Normalized Recall

- To take into account this effect the normalized recall was introduced.
- Imagine that we know all the relevant documents
  - an ideal system will present all the relevant documents before the not relevant ones.
Suppose that the relevant ones are 1, 2, 4, 5, 13 in a list of 14 documents. The graph obtained is:

The area between the two graphs (the black one) is a measure of the effectiveness of the system. This measure is always reduced to a value between 0 and 1: 1 for the ideal system and 0 for the system that presents all the relevant documents at the end.
Sliding Ratio

- Sliding ratio is a measure that takes into account the weight (the relevance value) of the documents retrieved and do not needs the knowledge of all the relevant documents.
- Assume that we retrieve N=5 documents that are ranked by the system. Then assume that the user assign a relevance value to these documents.

<table>
<thead>
<tr>
<th>n</th>
<th>doc</th>
<th>relevance, (w_i)</th>
<th>(w_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(d_1)</td>
<td>7.0</td>
<td>7.0</td>
</tr>
<tr>
<td>2</td>
<td>(d_2)</td>
<td>5.0</td>
<td>12.0</td>
</tr>
<tr>
<td>3</td>
<td>(d_3)</td>
<td>0.0</td>
<td>12.0</td>
</tr>
<tr>
<td>4</td>
<td>(d_4)</td>
<td>2.5</td>
<td>14.5</td>
</tr>
<tr>
<td>5</td>
<td>(d_5)</td>
<td>8.2</td>
<td>22.7</td>
</tr>
</tbody>
</table>

Sum of the weights so far
The perfect system will rank the documents in the same way of the user.

### Sliding Ratio

<table>
<thead>
<tr>
<th>doc</th>
<th>relevance, ($w_i$)</th>
<th>$W_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_5$</td>
<td>8.2</td>
<td>8.2</td>
</tr>
<tr>
<td>$d_4$</td>
<td>5.0</td>
<td>20.2</td>
</tr>
<tr>
<td>$d_3$</td>
<td>2.5</td>
<td>22.7</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0.0</td>
<td>22.7</td>
</tr>
</tbody>
</table>

Documents are rearranged

---

### Sliding Ratio

<table>
<thead>
<tr>
<th>$n$</th>
<th>doc</th>
<th>relevance, ($w_i$)</th>
<th>$w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$d_1$</td>
<td>7.0</td>
<td>7.0</td>
</tr>
<tr>
<td>2</td>
<td>$d_2$</td>
<td>5.0</td>
<td>12.0</td>
</tr>
<tr>
<td>3</td>
<td>$d_3$</td>
<td>0.0</td>
<td>12.0</td>
</tr>
<tr>
<td>4</td>
<td>$d_4$</td>
<td>2.5</td>
<td>14.5</td>
</tr>
<tr>
<td>5</td>
<td>$d_5$</td>
<td>8.2</td>
<td>22.7</td>
</tr>
</tbody>
</table>

Real system

### Sliding Ratio

<table>
<thead>
<tr>
<th>$doc$</th>
<th>relevance, ($w_i$)</th>
<th>$W_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_5$</td>
<td>8.2</td>
<td>8.2</td>
</tr>
<tr>
<td>$d_4$</td>
<td>5.0</td>
<td>20.2</td>
</tr>
<tr>
<td>$d_3$</td>
<td>2.5</td>
<td>22.7</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0.0</td>
<td>22.7</td>
</tr>
</tbody>
</table>

Ideal system
Sliding Ratio

The Sliding Ratio is the ratio of the last two columns

\[
SR = \frac{w_i}{W_i}
\]

<table>
<thead>
<tr>
<th>n</th>
<th>doc</th>
<th>relevance, (w_i)</th>
<th>(w_i)</th>
<th>doc</th>
<th>relevance, (w_i)</th>
<th>(W_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(d_1)</td>
<td>7.0</td>
<td>7.0</td>
<td>(d_1)</td>
<td>8.2</td>
<td>8.2</td>
</tr>
<tr>
<td>2</td>
<td>(d_2)</td>
<td>5.0</td>
<td>12.0</td>
<td>(d_1)</td>
<td>7.0</td>
<td>15.2</td>
</tr>
<tr>
<td>3</td>
<td>(d_3)</td>
<td>0.0</td>
<td>12.0</td>
<td>(d_2)</td>
<td>5.0</td>
<td>20.2</td>
</tr>
<tr>
<td>4</td>
<td>(d_4)</td>
<td>2.5</td>
<td>14.5</td>
<td>(d_4)</td>
<td>2.5</td>
<td>22.7</td>
</tr>
<tr>
<td>5</td>
<td>(d_5)</td>
<td>8.2</td>
<td>22.7</td>
<td>(d_3)</td>
<td>0.0</td>
<td>22.7</td>
</tr>
</tbody>
</table>

Real system

Ideal system

Sliding Ratio

If the number of retrieved documents \(N\) is large enough then SR is a reasonably accurate picture of the retrieval system performances
Homework 1

Imagine that an IR system retrieved 20 document in answer to a query, but only documents number 1, 3, 8, 9, 13, 15, and 20 are relevant.

Calculate Precision, Recall, Fallout and the ratio Recall/Fallout considering that there are other 5 relevant documents that were not retrieved and that the total number of documents in the collection is 100 (included the 20 retrieved).

Explore this problem using graphing applet

Homework 2

<table>
<thead>
<tr>
<th>Doc. Number</th>
<th>Rel=1</th>
<th>notRel=0</th>
<th>Relevance Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td></td>
<td>0.9</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td></td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td></td>
<td>0.2</td>
</tr>
</tbody>
</table>

Imagine that a pool of users assign a relevance weights to the relevant documents. Calculate the column of the sliding ratio.