Evaluation

Why System Evaluation?

2

- There are many retrieval models/ algorithms/ systems, which one is the best?
- What is the best component for:
  - Ranking function (dot-product, cosine, …)
  - Term selection (stopword removal, stemming…)
  - Term weighting (TF, TF-IDF,…)
- How far down the ranked list will a user need to look to find some/all relevant documents?

Difficulties in Evaluating IR Systems

- Effectiveness is related to the ***relevancy*** of retrieved items.
- Relevancy is not typically binary but continuous.
- Even if relevancy is binary, it can be a difficult judgment to make.
- Relevancy, from a human standpoint, is:
  - Subjective: Depends upon a specific user’s judgment.
  - Situational: Relates to user’s current needs.
  - Cognitive: Depends on human perception and behavior.
  - Dynamic: Changes over time.

Human Labeled Corpora (Gold Standard)

- Start with a corpus of documents.
- Collect a set of queries for this corpus.
- Have one or more human experts exhaustively label the relevant documents for each query.
- Typically assumes binary relevance judgments.
- Requires considerable human effort for large document/query corpora.

Should we instead use the accuracy measure for evaluation?

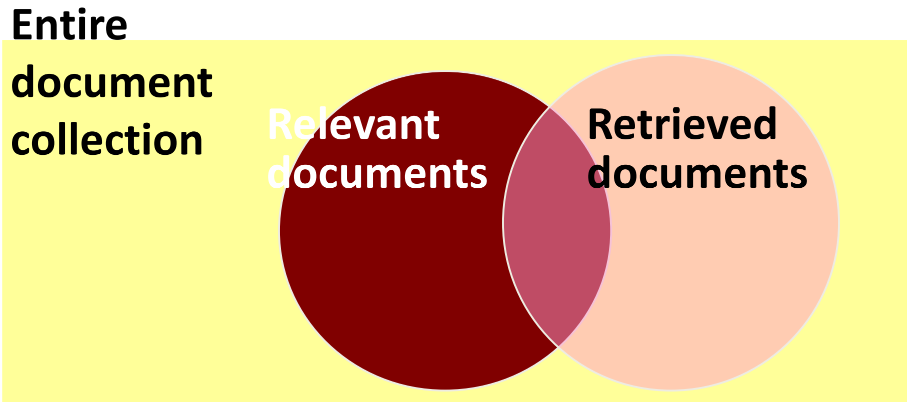
- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The **accuracy** of an engine: the fraction of these classifications that are correct
  - (tp + tn) / (tp + fp + fn + tn)
  - (t = true, f = false, p = positive, n = negative)
- **Accuracy** is a commonly used evaluation measure in machine learning classification work
- Why is this not a very useful evaluation measure in IR?

Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget….

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

Precision and Recall



	retrieved	not retrieved
irrelevant	retrieved & irrelevant	Not retrieved & irrelevant
relevant	retrieved & relevant	not retrieved but relevant

$$recall = \frac{Number\ of\ relevantdocumentsretrieved}{Total\ number\ of\ relevantdocuments}$$

$$precision = \frac{Number\ of\ relevantdocuments\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

Precision/Recall

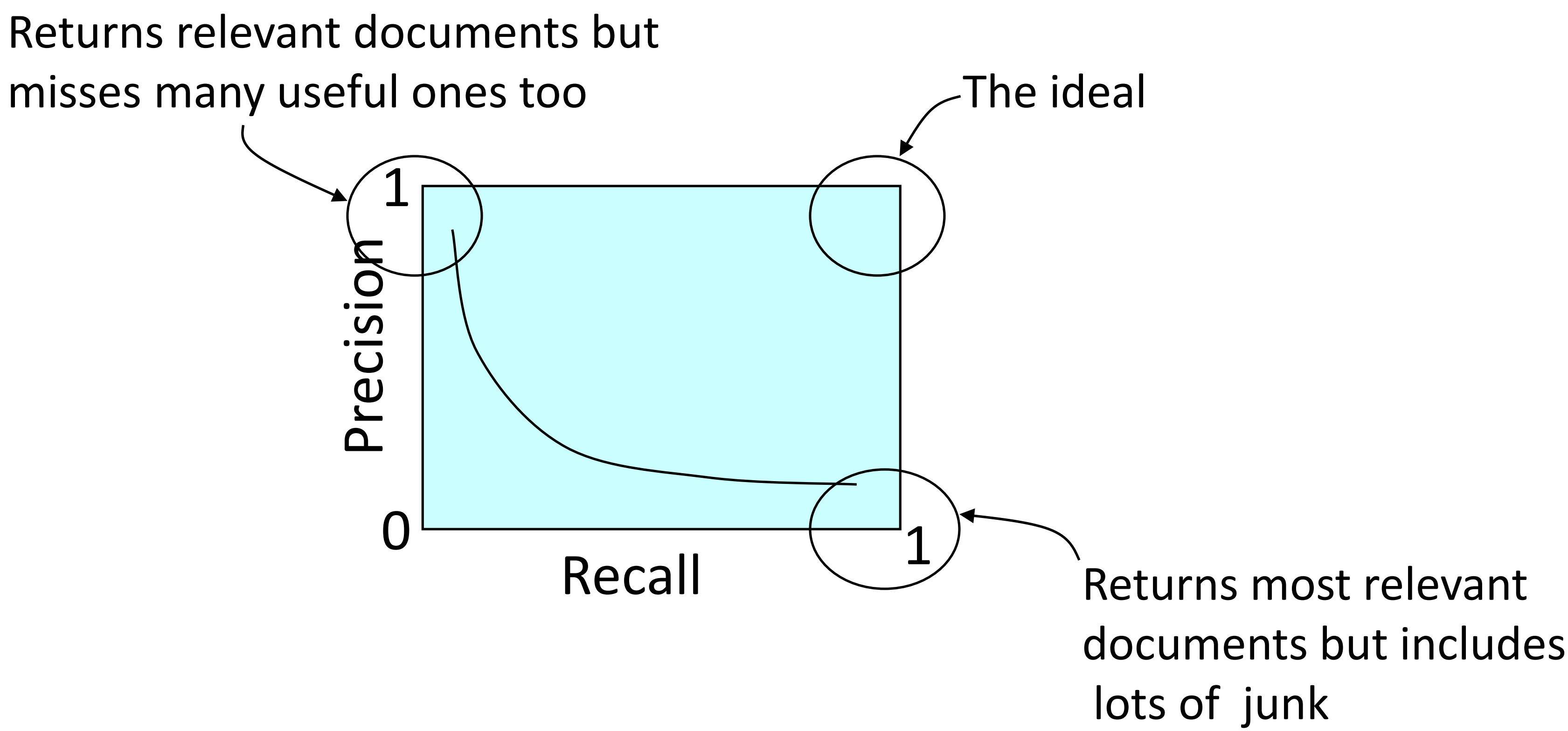
- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved

- In a good system, precision decreases as either the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

Precision and Recall

- Precision
  - The ability to retrieve top-ranked documents that are mostly relevant.
- Recall
  - The ability of the search to find ***all*** of the relevant items in the corpus.

Trade-off between Recall and Precision



Precision @*k*

- Perhaps most appropriate for most of web search: all people want are good matches on the first one or two results pages
- This leads to measuring precision at fixed low levels of retrieved results, such as 10 or 30 documents. This is referred to as “Precision at *k”, for example “Precision at 10”.*
- It has the advantage of not requiring any estimate of the size of the set of relevant documents

- Disadvantage:
  - It does not average well, since the total number of relevant documents for a query has a strong influence on precision at *k.*

Computing Recall/Precision Points

- For a given query, produce the ranked list of retrievals.
- Adjusting a threshold on this ranked list produces different sets of retrieved documents, and therefore different recall/precision measures.
- Mark each document in the ranked list that is relevant according to the gold standard.
- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.

Mean Average Precision (MAP)

13

- Mean average precision (MAP)
  - Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved
  - MAP for query collection is arithmetic average.
    - Macro-averaging: each query counts equally

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\mathrm{Precision}(R_{jk})$$

Q is total number of queries
m_j is total number of relevant documents for query j
R_{jk} is rank of k^{th} relevant document in ranked retrieval set of documents for query j

Average Precision : Example 1

n	Doc #	Relevant	Recall	Precision
1	588	X	0.1	1
2	589	X	0.2	1
3	576		0.2	0.67
4	534	X	0.3	0.75
5	577		0.3	0.6
6	103	X	0.4	0.667
7	234		0.4	0.57
8	543		0.4	0.5
9	134		0.4	0.44
10	654		0.4	0.4
11	356		0.4	0.36
12	635		0.4	0.33
13	333	X	0.5	0.38
14	643	X	0.6	0.42

Average Precision: (1 + 1 + 0.75 + 0.667 + 0.38 + 0.42)/10 = 0.42

Total Relevant documents for this query = R = 10

Average Precision: Example 2

n	Doc #	Relevant	Recall	Precision
1	588		0	0
2	589	X	0.1	0.5
3	576	X	0.2	0.67
4	534		0.2	0.5
5	577		0.2	0.4
6	103		0.2	0.33
7	234	X	0.3	0.42
8	543		0.3	0.38
9	134		0.3	0.33
10	654		0.3	0.3
11	356		0.3	0.27
12	635		0.3	0.25
13	333	X	0.4	0.31
14	643		0.4	0.28

Average Precision : (0.5 + 0.667 + 0.42 + 0.31)/10 = 0.19

Total Relevant documents for this query = R = 10

Mean Average Precision (MAP)

- **Average Precision**: Average of the precision values at the points at which each relevant document is retrieved.
  - Example 1: (1 + 1 + 0.75 + 0.667 + 0.38 + 0.42+0+0+0+0)/10 = 0.42
  - Example 2: (0.5 + 0.667 + 0.42 + 0.31+0+0+0+0+0+0)/10 = 0.19
  - (You can look at list of documents for these examples on slides 15 and 16)

- **Mean Average Precision**: Average of the average precision value for a set of queries.

Comparison of MAP and Precision at K

- MAP
  - System centric
  - If query has only one relevant document, and a very good retrieval system retrieves it at first rank , MAP will be 1
  - If a query has large number of relevant documents, and a poor retrieval system retrieves documents in random order then MAP will be not very high
- Precision at K
  - User Centric
  - If a query has only one relevant document , and a very good retrieval system retrieves it at first rank, Precision at 10 will be 0.1
  - If a query has large number of relevant documents, and a poor retrieval system retrieves documents in random order then P at 10 can very high

Average Precision (AP)

- It is more sensitive to changes at top ranks as compared to changes at bottom ranks
- For example, suppose total relevant documents = 1, both systems retrieve 1 relevant document

Rank	System 1	System 2
1	R	NR
2	NR	NR
3	NR	NR
4	NR	R
5	**NR**	**NR**
AP	**1**	**0.25**

Difference in AP values = 0.75

Rank	System 1	System 2
501	R	NR
502	NR	NR
503	NR	NR
504	NR	R
505	**NR**	**NR**
AP	**0.00199**	**0.00198**

Difference in AP values = 0.00000118

Navigational Queries

- When There’s only 1 Relevant Document
  - known-item search
  - navigational queries
  - looking for a fact
- Search Length = Rank of the answer
  - measures a user’s effort

Mean Reciprocal Rank (MRR)

- Consider rank position, K, of first relevant doc
- Reciprocal Rank score = $\frac{1}{K}$
- MRR is the mean Reciprocal Rank across multiple queries

Mean Reciprocal Rank (MRR)

- Easily interpretable for navigational queries
- If relevant document appears at first rank then RR will be 1/1 = 1.
- MRR 0f 0.5 means on average you will see relevant document at 2nd rank and MRR of 0.1 means on average you will see relevant document at 10th rank
- It measures amount of effort user has to spend looking for relevant document

Discounted Cumulative Gain

- Popular measure for evaluating web search and related tasks
- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant document
  - the lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

Discounted Cumulative Gain

- Uses *graded relevance as a measure of the usefulness, or gain, from examining a document*
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted, at lower ranks*
- Typical discount is 1/*log (rank)*
  - *With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3*

Discounted Cumulative Gain

- *DCG is the total gain accumulated at a particular rank p:*

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

- Alternative formulation:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

  – used by some web search companies
  – emphasis on retrieving highly relevant documents

DCG Example

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0

- discounted gain:

  3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0

- DCG:

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Motivation for Normalizing DCG

- Ranking for Q1 = [3,0,2,0,1,0,0,0,0]
- Ranking for Q2 = [3,1,2,2,1,2,1,0,2,2]

Motivation for Normalizing DCG

- Relevant documents for Query 1 = [3,2,1,0,0,0,0,0,0]
- Relevant documents for Query 2 = [3,3,3,3,3,3,3,2,2,2,2,2,1,1]

- Ranking for Q1 = [3,0,2,0,1,0,0,0,0]
- Ranking for Q2 = [3,1,2,2,1,2,1,0,2,2]

Motivation for Normalizing DCG

- Relevant documents for Query 1 = [3,2,1,0,0,0,0,0,0]
- Relevant documents for Query 2 = [3,3,3,3,3,3,3,2,2,2,2,2,1,1]

- Ranking for Q1 = [3,0,2,0,1,0,0,0,0]
- Ranking for Q2 = [3,1,2,2,1,2,1,0,2,2]

- DCG of Ranking for Q1 = 4.7
- DCG of Ranking for Q2 = 6.68

- NDCG of Ranking for Q1 = 0.83
- NDCG of Ranking for Q2 = 0.62

Normalized DCG

- DCG numbers are averaged across a set of queries at specific rank values
  - e.g., DCG at rank 5 is 6.89 and at rank 10 is 9.61
- DCG values are often normalized by comparing the DCG at each rank with the DCG value for the perfect ranking
  - *makes averaging easier for queries with different numbers of relevant documents*

NDCG Example

- Perfect ranking:

  3, 3, 3, 2, 2, 2, 1, 0, 0, 0

- ideal DCG values:

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10

- DCG of actual list: 3, 2, 3, 0, 0, 1, 2, 2, 3, 0

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

- NDCG values (divide actual by ideal)

  1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88

  NDCG ≤1 at any rank position

Slide Credits

- Christopher Manning