

Lecture 6: Evaluation

Information Retrieval

Computer Science Tripos Part II

Ronan Cummins¹

Natural Language and Information Processing (NLIP) Group



UNIVERSITY OF
CAMBRIDGE

ronan.cummins@cl.cam.ac.uk

2017

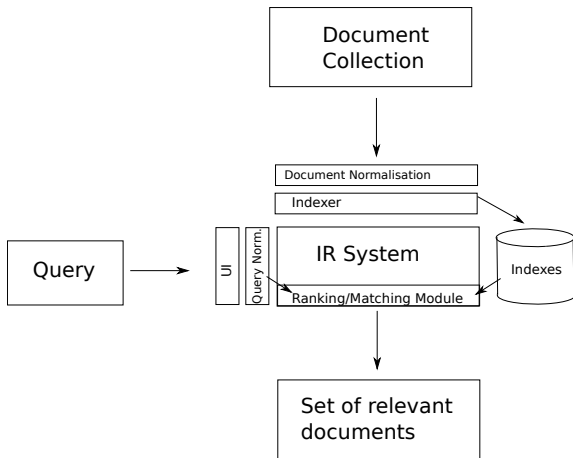
¹Adapted from Simone Teufel's original slides

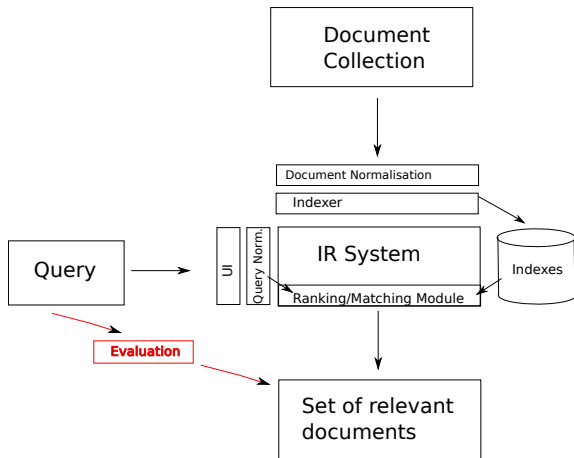
- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation
- 4 Ranked evaluation
- 5 Benchmarks
- 6 Other types of evaluation

- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation
- 4 Ranked evaluation
- 5 Benchmarks
- 6 Other types of evaluation

Summary: Ranked retrieval

- In VSM one represents documents and queries as weighted tf-idf vectors
- Compute the cosine similarity between the vectors to rank
- Language models rank based on the probability of a document model generating the query





Today: how good are the returned documents?

- 1 Recap/Catchup
- 2 Introduction**
- 3 Unranked evaluation
- 4 Ranked evaluation
- 5 Benchmarks
- 6 Other types of evaluation

- How fast does it index?
 - e.g., number of bytes per hour
- How fast does it search?
 - e.g., latency as a function of queries per second
- What is the cost per query?
 - in dollars

Measures for a search engine

- All of the preceding criteria are **measurable**: we can quantify speed / size / money
- However, the key measure for a search engine is **user happiness**.
- What is user happiness?
- Factors include:
 - Speed of response
 - Size of index
 - Uncluttered UI
 - We can measure
 - Rate of return to this search engine
 - Whether something was bought
 - Whether ads were clicked
 - Most important: **relevance**
 - (actually, maybe even more important: it's free)
- Note that none of these is sufficient: blindingly fast, but useless answers won't make a user happy.

Most common definition of user happiness: Relevance

- User happiness is equated with the relevance of search results to the query.
- But how do you measure relevance?
- Standard methodology in information retrieval consists of three elements.
 - A benchmark document collection
 - A benchmark suite of queries
 - An assessment of the relevance of each query-document pair

Relevance: query vs. information need

- Relevance to **what?** The query?

Information need i

"I am looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine."

- translated into:

Query q

[red wine white wine heart attack]

- So what about the following document:

Document d'

At the heart of his speech was an attack on the wine industry lobby for downplaying the role of red and white wine in drunk driving.

- d' is an excellent match for query q ...
- d' is **not** relevant to the information need i .

- User happiness can only be measured by relevance to an information need, not by relevance to queries.
- Sloppy terminology here and elsewhere in the literature: we talk about query–document relevance judgments even though we mean information–need–document relevance judgments.

Overview

- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation**
- 4 Ranked evaluation
- 5 Benchmarks
- 6 Other types of evaluation

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

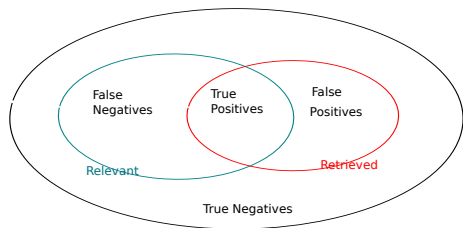
- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

THE TRUTH

WHAT THE
SYSTEM
THINKS

	Relevant	Nonrelevant
Retrieved	true positives (TP)	false positives (FP)
Not retrieved	false negatives (FN)	true negatives (TN)



$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

- You can increase recall by returning more docs.
- Recall is a non-decreasing function of the number of docs retrieved.
- A system that returns all docs has 100% recall!
- The converse is also true (usually): It's easy to get high precision for very low recall.

A combined measure: F

- F allows us to trade off precision against recall.

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1 - \alpha}{\alpha}$$

- $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, \infty]$
- Most frequently used: **balanced F** with $\beta = 1$ or $\alpha = 0.5$
 - This is the **harmonic mean** of P and R : $\frac{1}{F} = \frac{1}{2} \left(\frac{1}{P} + \frac{1}{R} \right)$

Example for precision, recall, F1

	relevant	not relevant	
retrieved	20	40	60
not retrieved	60	1,000,000	1,000,060
	80	1,000,040	1,000,120

- $P = 20 / (20 + 40) = 1/3$
- $R = 20 / (20 + 60) = 1/4$
- $F_1 = 2 \frac{1}{\frac{1}{3} + \frac{1}{4}} = 2/7$

- Why do we use complex measures like precision, recall, and F ?
- Why not something simple like accuracy?
- Accuracy is the fraction of decisions (relevant/nonrelevant) that are correct.
- In terms of the contingency table above,
accuracy = $(TP + TN)/(TP + FP + FN + TN)$.

Thought experiment

- Compute precision, recall and F_1 for this result set:

	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

- The snoogle search engine below always returns 0 results (“0 matching results found”), regardless of the query.

The logo for Snoogle.com, featuring the word "snoogle.com" in a stylized, multi-colored font (blue, red, and yellow) with a slight shadow effect.

Search for:

0 matching results found.

- Snoogle demonstrates that accuracy is not a useful measure in IR.

Why accuracy is a useless measure in IR

- Simple trick to maximize accuracy in IR: always say no and return nothing
- You then get 99.99% accuracy on most queries.
- Searchers on the web (and in IR in general) **want to find something** and have a certain tolerance for junk.
- It's better to return some bad hits as long as you return something.
- → We use precision, recall, and F for evaluation, not accuracy.

Recall-criticality and precision-criticality

- Inverse relationship between precision and recall forces general systems to go for compromise between them
- But some tasks particularly need good precision whereas others need good recall:

	Precision-critical task	Recall-critical task
Time	matters	matters less
Tolerance to cases of overlooked information	a lot	none
Information Redundancy	There may be many equally good answers	Information is typically found in only one document
Examples	web search	legal search, patent search

Difficulties in using precision, recall and F

- We should always average over a large set of queries.
 - There is no such thing as a “typical” or “representative” query.
- We need relevance judgments for information-need-document pairs – but they are expensive to produce.
- For alternatives to using precision/recall and having to produce relevance judgments – see end of this lecture.

Overview

- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation
- 4 Ranked evaluation**
- 5 Benchmarks
- 6 Other types of evaluation

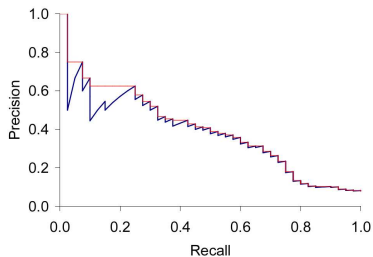
Moving from unranked to ranked evaluation

- Precision/recall/F are measures for **unranked sets**.
- We can easily turn set measures into measures of **ranked lists**.
- Just compute the set measure for each “prefix”: the top 1, top 2, top 3, top 4 etc results
- This is called Precision/Recall at Rank
- Rank statistics give some indication of how quickly user will find relevant documents from ranked list

Rank	Doc
1	d ₁₂
2	d ₁₂₃
3	d ₄
4	d ₅₇
5	d ₁₅₇
6	d ₂₂₂
7	d ₂₄
8	d ₂₆
9	d ₇₇
10	d ₉₀

- Blue documents are relevant
- $P@n$: $P@3=0.33$, $P@5=0.2$, $P@8=0.25$
- $R@n$: $R@3=0.33$, $R@5=0.33$, $R@8=0.66$

A precision-recall curve



- Each point corresponds to a result for the top k ranked hits ($k = 1, 2, 3, 4, \dots$)
- **Interpolation (in red): Take maximum of all future points**
- Rationale for interpolation: The user is willing to look at more stuff if both precision and recall get better.

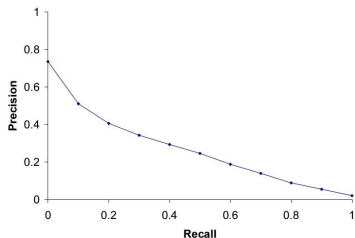
Another idea: Precision at Recall r

Rank	S1	S2
1	X	
2		X
3	X	
4		
5		X
6	X	X
7		X
8		X
9	X	
10	X	

→

	S1	S2
p @ r 0.2	1.0	0.5
p @ r 0.4	0.67	0.4
p @ r 0.6	0.5	0.5
p @ r 0.8	0.44	0.57
p @ r 1.0	0.5	0.63

Averaged 11-point precision/recall graph



- Compute interpolated precision at recall levels 0.0, 0.1, 0.2, ...
- Do this for each of the queries in the evaluation benchmark
- Average over queries
- The curve is typical of performance levels at TREC (more later).

Averaged 11-point precision more formally

$$P_{11-pt} = \frac{1}{11} \sum_{j=0}^{10} \frac{1}{N} \sum_{i=1}^N \tilde{P}_i(r_j)$$

with $\tilde{P}_i(r_j)$ the precision at the j th recall point in the i th query (out of N)

- Define 11 standard recall points $r_j = \frac{j}{10}$: $r_0 = 0$, $r_1 = 0.1$... $r_{10} = 1$
- To get $\tilde{P}_i(r_j)$, we can use $P_i(R = r_j)$ directly if a new relevant document is retrieved exactly at r_j
- Interpolation for cases where there is no exact measurement at r_j :

$$\tilde{P}_i(r_j) = \begin{cases} \max(r_j \leq r < r_{j+1}) P_i(R = r) & \text{if } P_i(R = r) \text{ exists} \\ \tilde{P}_i(r_{j+1}) & \text{otherwise} \end{cases}$$

- Note that $P_i(R = 1)$ can always be measured.
- Worked avg-11-pt prec example for supervisions at end of slides.

Mean Average Precision (MAP)

- Also called “average precision at seen relevant documents”
- Determine precision at each point when a new relevant document gets retrieved
- Use $P=0$ for each relevant document that was not retrieved
- Determine average for each query, then average over queries

$$MAP = \frac{1}{N} \sum_{j=1}^N \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(doc_i)$$

with:

- Q_j number of relevant documents for query j
- N number of queries
- $P(doc_i)$ precision at i th relevant document

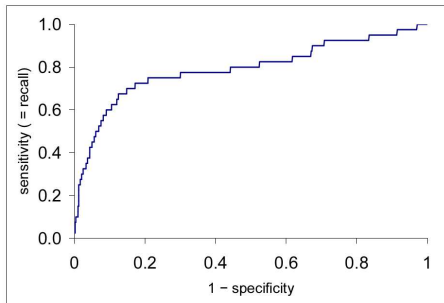
Mean Average Precision: example

$$(MAP = \frac{0.564+0.623}{2} = 0.594)$$

Query 1		
Rank		$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		
10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.25
AVG:		0.564

Query 2		
Rank		$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	0.2
AVG:		0.623

ROC curve (Receiver Operating Characteristic)



- x-axis: FPR (false positive rate): $FP / \text{total actual negatives}$;
- y-axis: TPR (true positive rate): $TP / \text{total actual positives}$, (also called sensitivity) \equiv recall
- $FPR = \text{fall-out} = 1 - \text{specificity}$ (TNR; true negative rate)
- But we are only interested in the small area in the lower left corner (blown up by prec-recall graph)

Variance of measures like precision/recall

- For a test collection, it is usual that a system does badly on some information needs (e.g., $P = 0.2$ at $R = 0.1$) and really well on others (e.g., $P = 0.95$ at $R = 0.1$).
- Indeed, it is usually the case that the **variance of the same system across queries** is much **greater than the variance of different systems on the same query**.
- That is, there are easy information needs and hard ones.

- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation
- 4 Ranked evaluation
- 5 Benchmarks**
- 6 Other types of evaluation

What we need for a benchmark

- A collection of documents
 - Documents must be representative of the documents we expect to see in reality.
- A collection of information needs
 - . . . which we will often incorrectly refer to as queries
 - Information needs must be representative of the information needs we expect to see in reality.
- Human relevance assessments
 - We need to hire/pay “judges” or assessors to do this.
 - Expensive, time-consuming
 - Judges must be representative of the users we expect to see in reality.

First standard relevance benchmark: Cranfield

- Pioneering: first testbed allowing precise quantitative measures of information retrieval effectiveness
- Late 1950s, UK
- 1398 abstracts of aerodynamics journal articles, a set of 225 queries, exhaustive relevance judgments of all query-document-pairs
- Too small, too untypical for serious IR evaluation today

Second-generation relevance benchmark: TREC

- TREC = Text Retrieval Conference (TREC)
- Organized by the U.S. National Institute of Standards and Technology (NIST)
- TREC is actually a set of several different relevance benchmarks.
- Best known: TREC Ad Hoc, used for first 8 TREC evaluations between 1992 and 1999
- 1.89 million documents, mainly newswire articles, 450 information needs
- No exhaustive relevance judgments – too expensive
- Rather, NIST assessors' relevance judgments are available only for the documents that were among the top k returned for some system which was entered in the TREC evaluation for which the information need was developed.

<num> Number: 508

<title> hair loss is a symptom of what diseases

<desc> Description:

Find diseases for which hair loss is a symptom.

<narr> Narrative:

A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, “thinning hair” and “hair loss” are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.

TREC Relevance Judgements



Humans decide which document–query pairs are relevant.

Example of more recent benchmark: ClueWeb09

- 1 billion web pages
- 25 terabytes (compressed: 5 terabyte)
- Collected January/February 2009
- 10 languages
- Unique URLs: 4,780,950,903 (325 GB uncompressed, 105 GB compressed)
- Total Outlinks: 7,944,351,835 (71 GB uncompressed, 24 GB compressed)

Interjudge agreement at TREC

information need	number of docs judged	disagreements
51	211	6
62	400	157
67	400	68
95	400	110
127	400	106

Impact of interjudge disagreement

- Judges disagree a lot. Does that mean that the results of information retrieval experiments are meaningless?
- No.
- Large impact on absolute performance numbers
- Virtually no impact on ranking of systems
- Supposes we want to know if algorithm A is better than algorithm B
- An information retrieval experiment will give us a reliable answer to this question . . .
- . . . even if there is a lot of disagreement between judges.

Overview

- 1 Recap/Catchup
- 2 Introduction
- 3 Unranked evaluation
- 4 Ranked evaluation
- 5 Benchmarks
- 6 Other types of evaluation

Evaluation at large search engines

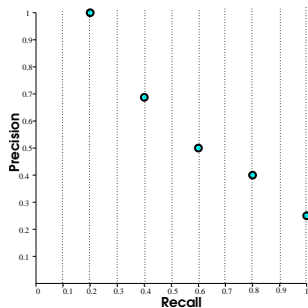
- Recall is difficult to measure on the web
- Search engines often use precision at top k , e.g., $k = 10 \dots$
- \dots or use measures that reward you more for getting rank 1 right than for getting rank 10 right.
- Search engines also use non-relevance-based measures.
 - **Example 1: clickthrough** on first result
 - Not very reliable if you look at a single clickthrough (you may realize after clicking that the summary was misleading and the document is nonrelevant) \dots
 - \dots but pretty reliable in the aggregate.
 - **Example 2: A/B testing**

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most

- Focused on evaluation for ad-hoc retrieval
 - Precision, Recall, F-measure
 - More complex measures for ranked retrieval
 - other issues arise when evaluating different tracks, e.g. QA, although typically still use P/R-based measures
- Evaluation for **interactive** tasks is more involved
- Significance testing is an issue
 - could a good result have occurred by chance?
 - is the result robust across different document sets?
 - slowly becoming more common
 - underlying population distributions unknown, so apply non-parametric tests such as the sign test

- MRS, Chapter 8

Worked Example avg-11-pt prec: Query 1, measured data points



- Blue for Query 1
- Bold Circles measured

Query 1			
Rank		R	P
1	X	0.2	1.00
2			
3	X	0.4	0.67
4			
5			
6	X	0.6	0.50
7			
8			
9			
10	X	0.8	0.40
11			
12			
13			
14			
15			
16			
17			
18			
19			
20	X	1.0	0.25

$$\tilde{P}_1(r_2) = 1.00$$

$$\tilde{P}_1(r_4) = 0.67$$

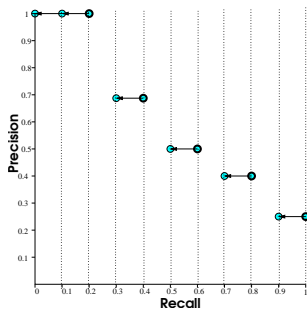
$$\tilde{P}_1(r_6) = 0.50$$

$$\tilde{P}_1(r_8) = 0.40$$

$$\tilde{P}_1(r_{10}) = 0.25$$

- Five r_j s ($r_2, r_4, r_6, r_8, r_{10}$) coincide directly with datapoint

Worked Example avg-11-pt prec: Query 1, interpolation



- Bold circles measured
- thin circles interpolated

Query 1			
Rank		R	P
1	X	.20	1.00
2			
3	X	.40	.67
4			
5			
6	X	.60	.50
7			
8			
9			
10	X	.80	.40
11			
12			
13			
14			
15			
16			
17			
18			
19			
20	X	1.00	.25

$$\tilde{P}_1(r_0) = 1.00$$

$$\tilde{P}_1(r_1) = 1.00$$

$$\tilde{P}_1(r_2) = 1.00$$

$$\tilde{P}_1(r_3) = .67$$

$$\tilde{P}_1(r_4) = .67$$

$$\tilde{P}_1(r_5) = .50$$

$$\tilde{P}_1(r_6) = .50$$

$$\tilde{P}_1(r_7) = .40$$

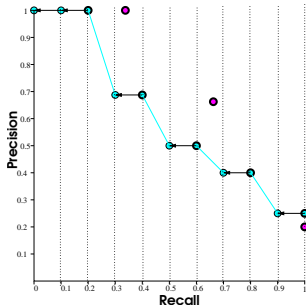
$$\tilde{P}_1(r_8) = .40$$

$$\tilde{P}_1(r_9) = .25$$

$$\tilde{P}_1(r_{10}) = .25$$

- The six other r_j s (r_0 , r_1 , r_3 , r_5 , r_7 , r_9) are interpolated.

Worked Example avg-11-pt prec: Query 2, measured data points



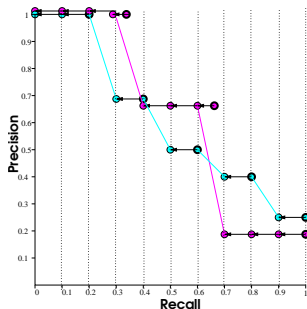
- Blue: Query 1; Red: Query 2
- Bold circles measured; thin circles interpol.

Query 2			
Rank	Relev.	R	P
1	X	.33	1.00
2			
3	X	.67	.67
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15	X	1.0	.2

$$\tilde{P}_2(r_{10}) = .20$$

- Only r_{10} coincides with a measured data point

Worked Example avg-11-pt prec: Query 2, interpolation



- Blue: Query 1; Red: Query 2
- Bold circles measured; thin circles interpol.

Query 2			
Rank	Relev.	R	P
1	X	.33	1.00
2			
3	X	.67	.67
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15	X	1.0	.2

$$\tilde{P}_2(r_0) = 1.00$$

$$\tilde{P}_2(r_1) = 1.00$$

$$\tilde{P}_2(r_2) = 1.00$$

$$\tilde{P}_2(r_3) = 1.00$$

$$\tilde{P}_2(r_4) = .67$$

$$\tilde{P}_2(r_5) = .67$$

$$\tilde{P}_2(r_6) = .67$$

$$\tilde{P}_2(r_7) = .20$$

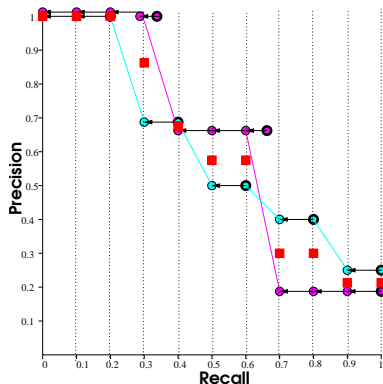
$$\tilde{P}_2(r_8) = .20$$

$$\tilde{P}_2(r_9) = .20$$

$$\tilde{P}_2(r_{10}) = .20$$

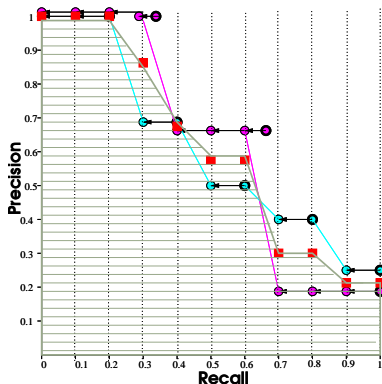
- 10 of the r_j s are interpolated

Worked Example avg-11-pt prec: averaging



- Now average at each p_j
- over N (number of queries)
- \rightarrow 11 averages

Worked Example avg-11-pt prec: area/result



- End result:
- 11 point average precision
- Approximation of area under prec. recall curve