Introduction to

Information Retrieval

Evaluation

How do you tell if users are happy?

- Search returns products relevant to users
  - How do you assess this at scale?
- Search results get clicked a lot
  - Misleading titles/summaries can cause users to click
- Repeat visitors/buyers
  - Do users leave soon after searching?
  - Do they come back within a week/month/… ?

Measuring relevance

- Three elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. An assessment of either Relevant or Nonrelevant for each query and each document

Relevance judgments

- Binary (relevant vs. non-relevant) in the simplest case
- More relevance levels also used(0, 1, 2, 3 ...)

Early public test Collections (20th C)

TABLE 4.3 Common Test Corpora

Collection	NDocs	NQrys	Size (MB)	Term/Doc	Q-D RelAss
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

Recent datasets: 100s of million web pages (GOV, ClueWeb, …)

Now we have the basics of a benchmark

- Let's review some evaluation measures
  - *Precision*
  - *Recall*
  - DCG
  - …

Evaluating an IR system

- Note: **user need** is translated into a **query**
- Relevance is assessed relative to the **user need,** *not* the **query**
- E.g., Information need: *My swimming pool bottom is becoming black and needs to be cleaned.*
- Query: ***pool cleaner***
- Assess whether the doc addresses the underlying need, not whether it has these words

Unranked retrieval evaluation: Precision and Recall – recap from IIR 8/video

- **<u>Binary assessments</u>**

Precision: fraction of retrieved docs that are relevant = P(relevant|retrieved)

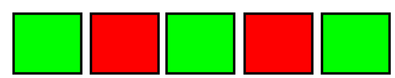
Recall: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

Precision@K

- Set a rank threshold K
- Compute % relevant in top K
- Ignores documents ranked lower than K
- Ex:
  - Prec@3 of 2/3
  - Prec@4 of 2/4
  - Prec@5 of 3/5

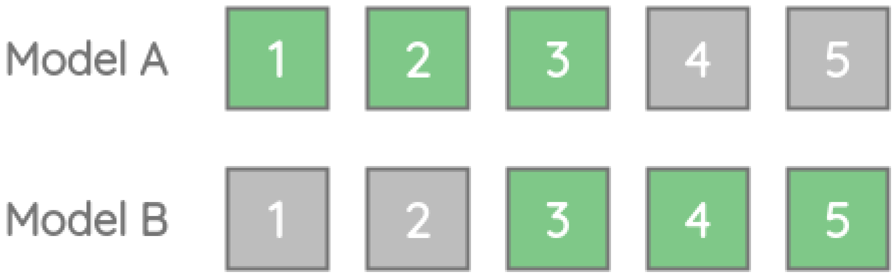


Precision@K

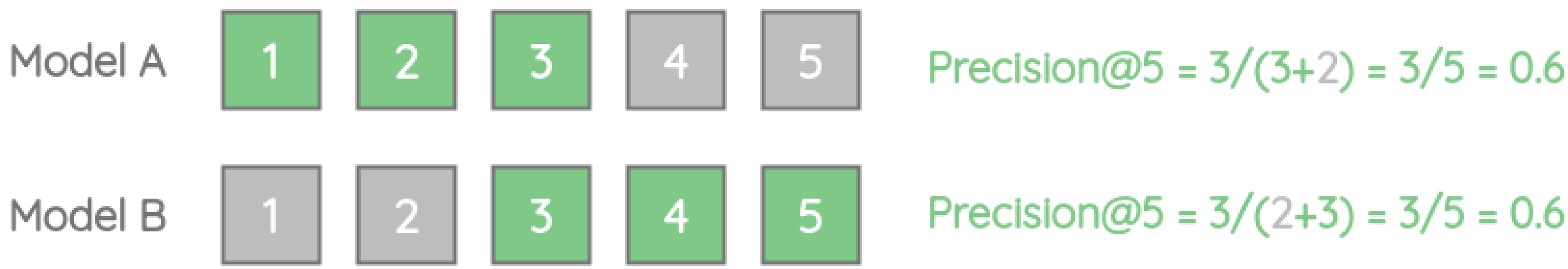
$$Precision@k = \frac{true\ positives@k}{(true\ positives@k) + (false\ positives@k)}$$

k	1	2	3	4	5
Precision@k	$\frac{1}{1} = 1$	$\frac{1}{2} = 0.5$	$\frac{2}{3} = 0.67$	$\frac{2}{4} = 0.5$	$\frac{3}{5} = 0.6$

Calculate Precision@5



Calculate Precision@5



Model A: first three items relevant,
Model B: last three items relevant.
Precision@5 same for both of these models even though model A is better.

Doesn't consider the position of the relevant items !!

Recall@K

$$Recall@k = \frac{true\ positives@k}{(true\ positives@k) + (false\ negatives@k)}$$

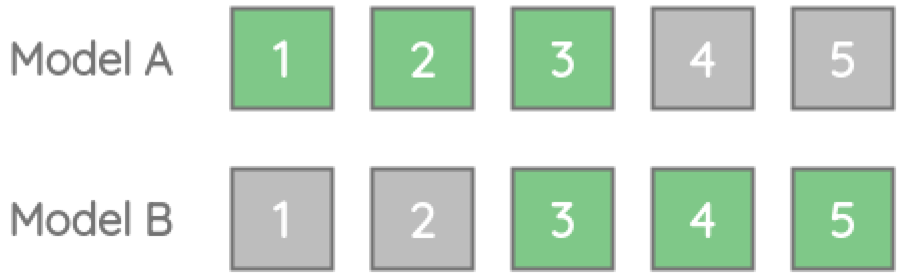
Recall@1 = 1/3 = 0.33

Recall@3 = 2/(2+1) = 2/3 = 0.67

Recall@K

k	1	2	3	4	5
Recall@k	$\frac{1}{(1+2)} = \frac{1}{3} = 0.33$	$\frac{1}{(1+2)} = \frac{1}{3} = 0.33$	$\frac{2}{(2+1)} = \frac{2}{3} = 0.67$	$\frac{2}{(2+1)} = \frac{2}{3} = 0.67$	$\frac{3}{(3+0)} = \frac{3}{3} = 1$

Calculate Recall@5



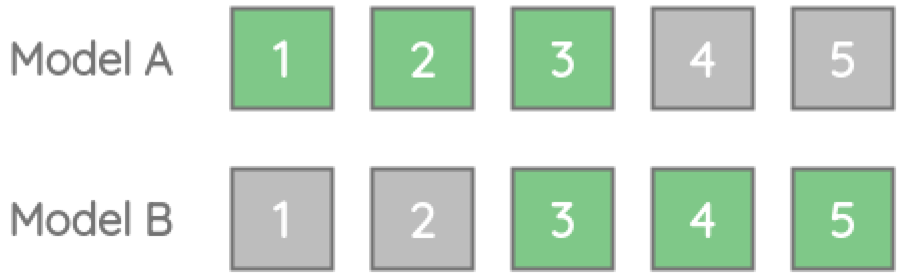
F1 Score@K

F1 Score: Harmonic mean of precision and recall

$$F1@k = \frac{2 * (Precision@k) * (Recall@k)}{(Precision@k) + (Recall@k)}$$

k	1	2	3	4	5
Precision@k	1	1/2	2/3	1/2	3/5
Recall@k	1/3	1/3	2/3	2/3	1
F1@k	$\frac{2*1*(1/3)}{(1+1/3)} = 0.5$	$\frac{2*(1/2)*(1/3)}{(1/2+1/3)} = 0.4$	$\frac{2*(2/3)*(2/3)}{(2/3+2/3)} = 0.666$	$\frac{2*(1/2)*(2/3)}{(1/2+2/3)} = 0.571$	$\frac{2*(3/5)*1}{(3/5+1)} = 0.749$

Calculate F1 Score@5



Order aware metrics

Mean Reciprocal Rank

useful when we want system to return best relevant item and want that item to be at higher position.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

$|Q|$ denotes total number of queries

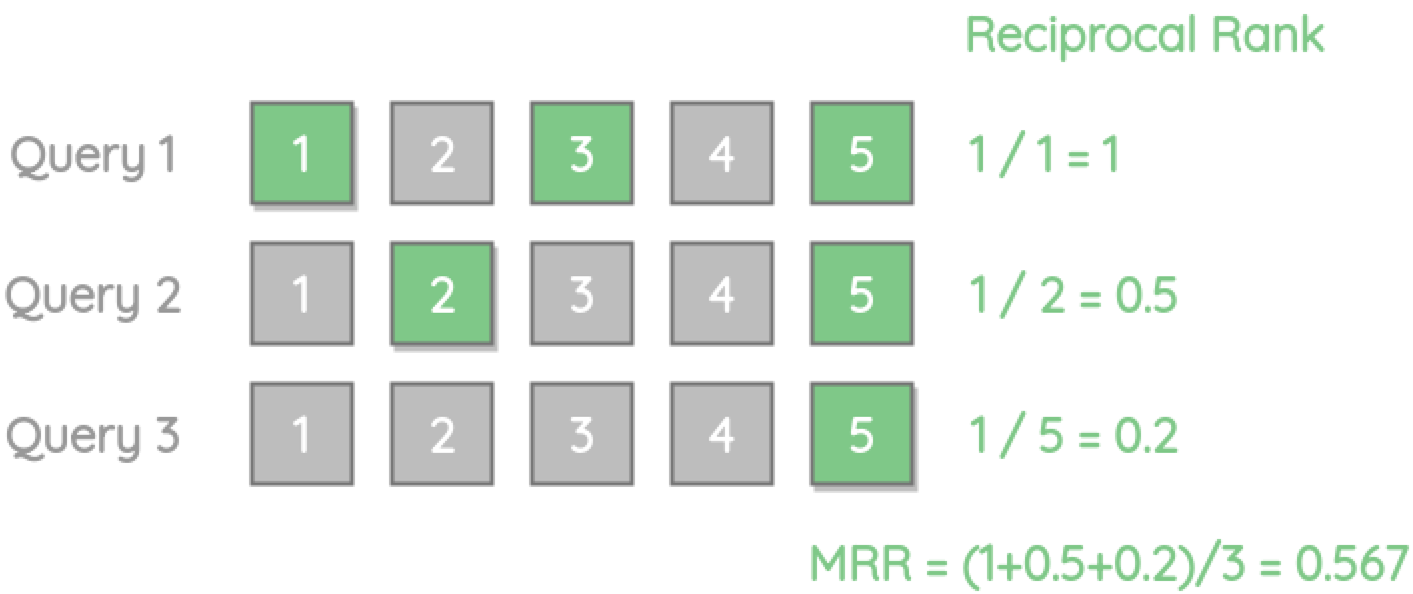
$rank_i$ denotes the rank of the first relevant result

Mean Reciprocal Rank

Calculate reciprocal of rank and then average across queries

Mean Reciprocal Rank

Calculate reciprocal of rank and then average across queries



MRR doesn't care about position of remaining relevant results.
If use-case requires returning multiple relevant results in the best possible way, it may not be a good metric

Average precision (AP)

Evaluates whether all of the ground-truth relevant items selected by model are ranked higher or not.

Unlike MRR, it considers all the relevant items.

$$AP = \frac{\sum_{k=1}^{n} (P(k) * rel(k))}{number\ of\ relevant\ items}$$

$rel(k)$ indicator function which is 1 when item at rank K is relevant.

$P(k)$ Precision@k metric

Average precision (AP)

	1	2	3	4	5
Precision@K	1	1/2	2/3	2/4	3/5

$$AP = \frac{(1 + 2/3 + 3/5)}{3} = 0.7555$$

Calculate Average precision (AP)

	1	2	3	4	5
Precision@K	1	1	1	3/4	3/5

Calculate Average precision (AP)

	1	2	3	4	5
Precision@K	1	1	1	3/4	3/5

$$AP = \frac{(1+1+1)}{3} = 1$$

Mean Average precision (MAP)

Evaluate average precision across multiple queries

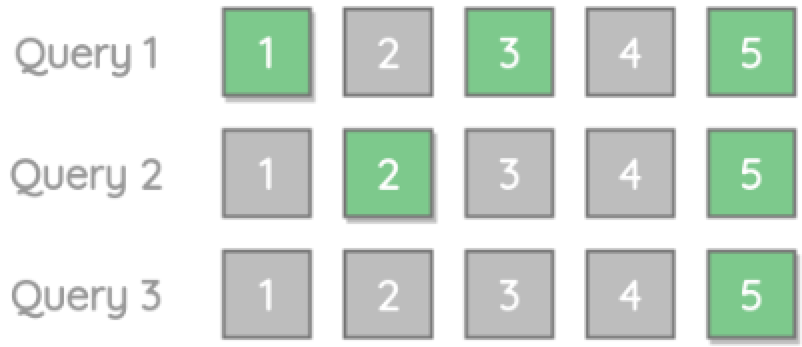
Given by: mean of average precision of different queries

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q)$$

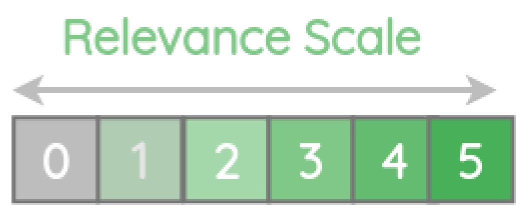
Q total number of queries

$AP(q)$ average precision for query q

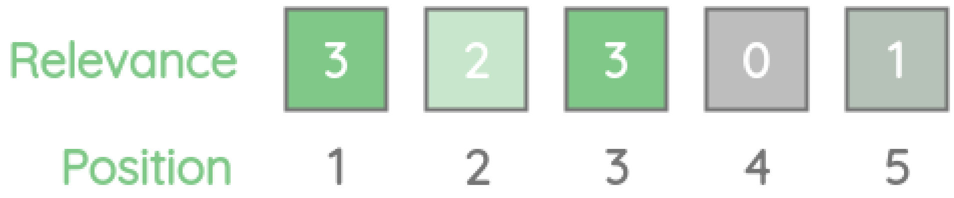
Calculate MAP



Relevance Grading



0 denotes least relevant and 5 denotes the most relevant



Cumulative Gain (CG@k)

Sum up relevance scores for top-K items

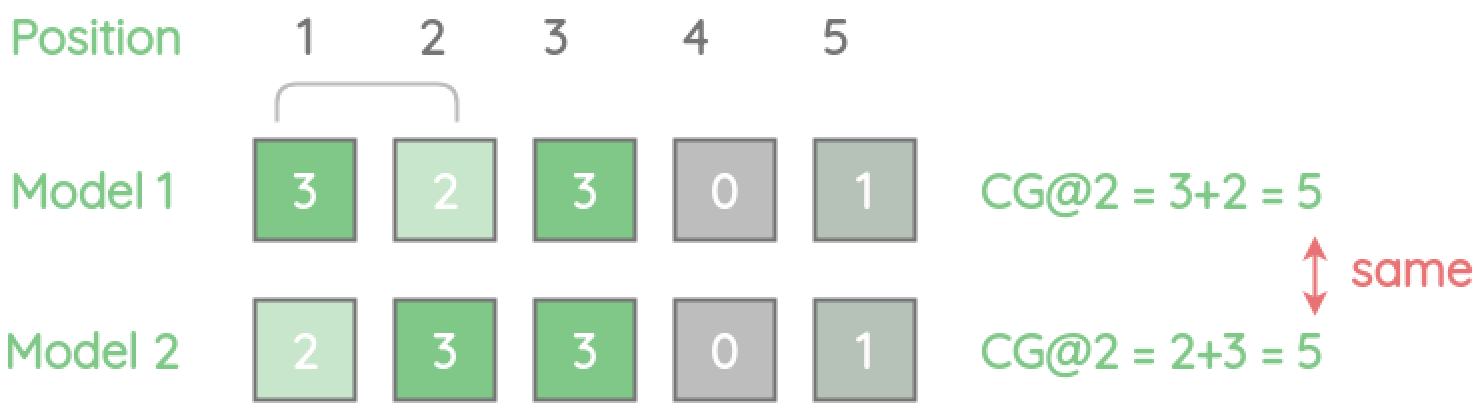
$$CG@k = \sum_{1}^{k} rel_i$$

CG@2:

Calculate Cumulative Gain (CG@k)

Position(k)	1	2	3	4	5
Cumulative Gain@k	3	3+2=5	3+2+3=8	3+2+3+0=8	3+2+3+0+1=9

Cumulative Gain (CG@k)



Item with relevance score of 3 at position 1 is better than same item relevance score 3 at position 2 !

Discounted Cumulative Gain (DCG@k)

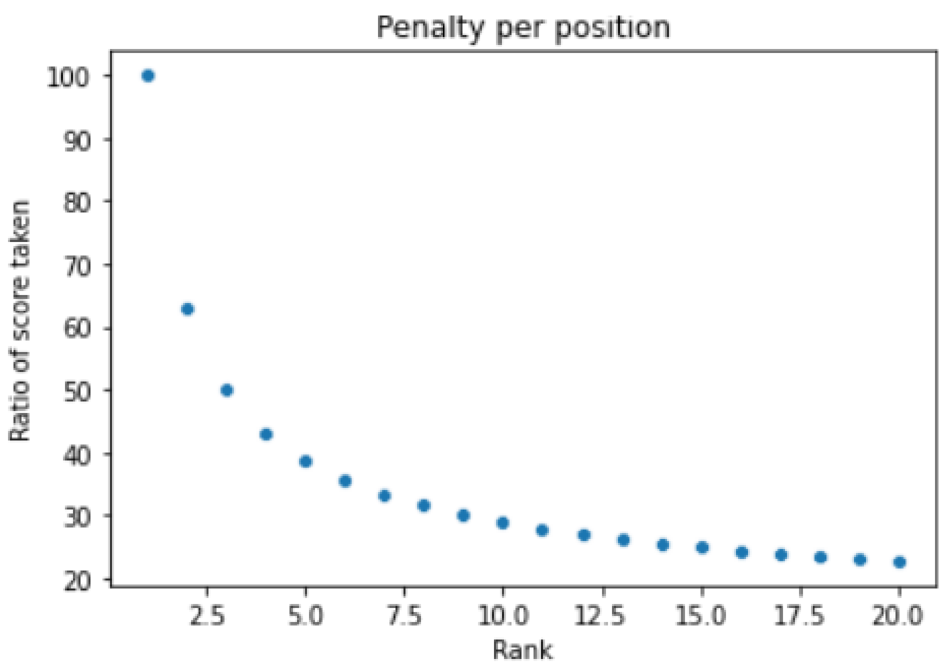
Penalize scores by their position.

Introduces log-based penalty function to reduce relevance score at each position.

i	$log_2(i+1)$
1	$log_2(1+1) = log_2(2) = 1$
2	$log_2(2+1) = log_2(3) = 1.5849625007211563$
3	$log_2(3+1) = log_2(4) = 2$
4	$log_2(4+1) = log_2(5) = 2.321928094887362$
5	$log_2(5+1) = log_2(6) = 2.584962500721156$

Discounted Cumulative Gain (DCG@k)

i	$log_2(i+1)$
1	$log_2(1+1) = log_2(2) = 1$
2	$log_2(2+1) = log_2(3) = 1.5849625007211563$
3	$log_2(3+1) = log_2(4) = 2$
4	$log_2(4+1) = log_2(5) = 2.321928094887362$
5	$log_2(5+1) = log_2(6) = 2.584962500721156$



Discounted Cumulative Gain (DCG@k)

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

$Position(i)$	$Relevance(rel_i)$	$log_2(i+1)$	$\frac{rel_i}{log_2(i+1)}$
1	3	$log_2(1+1) = log_2(2) = 1$	3 / 1 = 3
2	2	$log_2(2+1) = log_2(3) = 1.5849625007211563$	2 / 1.5849 = 1.2618
3	3	$log_2(3+1) = log_2(4) = 2$	3 / 2 = 1.5
4	0	$log_2(4+1) = log_2(5) = 2.321928094887362$	0 / 2.3219 = 0
5	1	$log_2(5+1) = log_2(6) = 2.584962500721156$	1 / 2.5849 = 0.3868

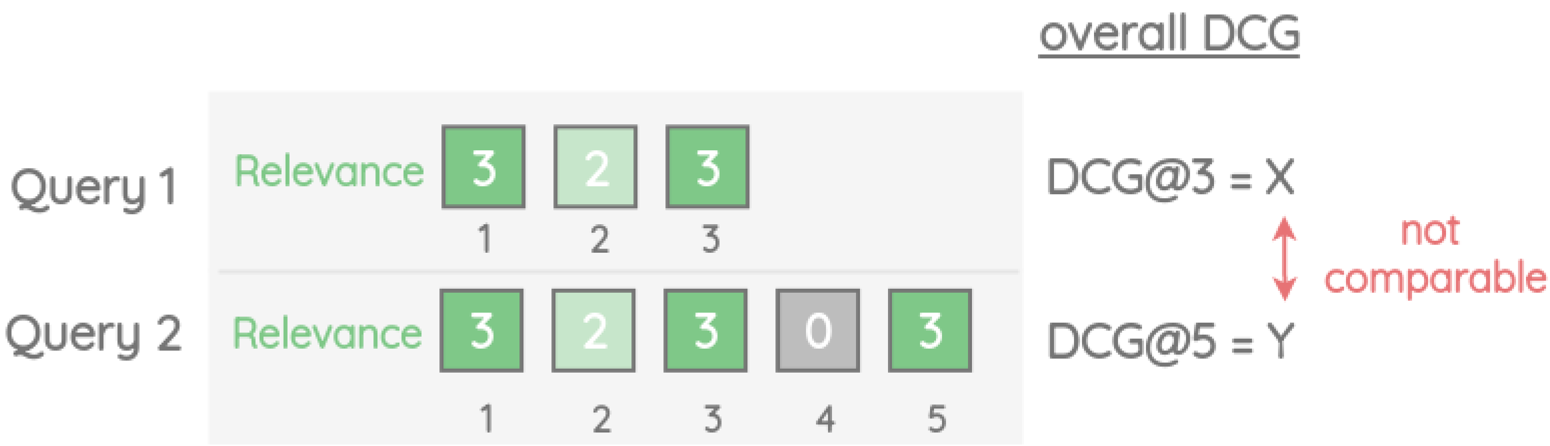
Discounted Cumulative Gain (DCG@k)

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

k	DCG@k
DCG@1	3
DCG@2	$3 + 1.2618 = 4.2618$
DCG@3	$3 + 1.2618 + 1.5 = 5.7618$
DCG@4	$3 + 1.2618 + 1.5 + 0 = 5.7618$
DCG@5	$3 + 1.2618 + 1.5 + 0 + 0.3868 = 6.1486$

Discounted Cumulative Gain (DCG@k)

Different size queries



Normalized Discounted Cumulative Gain (NDCG@k)

Normalize DCG values using ideal order of the relevant items to allow comparison across queries

Normalized Discounted Cumulative Gain (NDCG@k)

Ideal Order of Items

Relevance	3	3	2	1	0
Position	1	2	3	4	5

$Position(i)$	$Relevance(rel_i)$	$log_2(i+1)$	$\frac{rel_i}{log_2(i+1)}$	IDCG@k
1	3	$log_2(2) = 1$	3 / 1 = 3	3
2	3	$log_2(3) = 1.5849$	3 / 1.5849 = 1.8927	3+1.8927=4.8927
3	2	$log_2(4) = 2$	2 / 2 = 1	3+1.8927+1=5.8927
4	1	$log_2(5) = 2.3219$	1 / 2.3219 = 0.4306	3+1.8927+1+0.4306=6.3233
5	0	$log_2(6) = 2.5849$	0 / 2.5849 = 0	3+1.8927+1+0.4306+0=6.3233

Normalized Discounted Cumulative Gain (NDCG@k)

$$NDCG@k = \frac{DCG@k}{IDCG@k}$$

k	DCG@k	IDCG@k	NDCG@k
1	3	3	3 / 3 = 1
2	4.2618	4.8927	4.2618 / 4.8927 = 0.8710
3	5.7618	5.8927	5.7618 / 5.8927 = 0.9777
4	5.7618	6.3233	5.7618 / 6.3233 = 0.9112
5	6.1486	6.3233	6.1486 / 6.3233 = 0.9723

Scores range between 0 and 1.
A perfect ranking would get a score of 1.
Can also compare NDCG@k scores of different queries since it's a normalized score