Information Retrieval (CS60092)
Class Notes - 14 February (Part-2)

Suhas Jain - 19CS30048

1 Unranked Retrieval Evaluation

Information Systems can be measured with two metrics: precision and recall. When a user decides to search for information on a topic, the total database and the results to be obtained can be divided into 4 categories:

- Relevant and Retrieved or True Positive (tp)
- Relevant and Not Retrieved or False Negative (fn)
- Non-Relevant and Retrieved or False Positive (fp)
- Non-Relevant and Not Retrieved or True Negative (tn)

1.1 Precision

Precision is defined as the ratio of the number of relevant and retrieved documents(number of items retrieved that are actually useful to the user and match his search need) to the number of total retrieved documents from the query. Precision measures one aspect of information retrieval overhead for a user associated with a particular search.

$$\mathbf{Precision} = \frac{\mathbf{tp}}{\mathbf{tp} + \mathbf{fp}}$$

1.2 Recall

Recall is defined as ratio of the number of retrieved and relevant documents(the number of items retrieved that are relevant to the user and match his needs) to the number of possible relevant documents(number of relevant documents in the database). Recall measures to what extent a system processing a particular query is able to retrieve the relevant items the user is interested in seeing.

$$\mathbf{Recall} = \frac{\mathbf{tp}}{\mathbf{tp} + \mathbf{fn}}$$

1.3 Why is accuracy a bad measure?

$$\mathbf{Accuracy} = \frac{\mathbf{tp} + \mathbf{tn}}{\mathbf{tp} + \mathbf{fp} + \mathbf{tn} + \mathbf{fn}}$$

In case there is a big imbalance and one of the terms (let's say tn) is much larger than the others than we can see in the formula of accuracy that even if the other term (tp) is very low or even zero the accuracy remains very high. So, even when IR system is not retrieving the relevant documents (which is not ideal) our accuracy remains high because the system is not retrieving the large number of non-relevant documents also. Therefore we need better measures like precision and recall to better compare and understand our systems in case of such class imbalances which are very common.

2 Ranked Retrieval Evaluation

More often than not in IR systems we get documents in a ranked manner and not just a classification of what is relevant and what is not so we need better metrics to evaluate such results.

2.1 Binary Relevance

It is a method where a document will just have two levels of relevance that is zero or one.

2.1.1 Precision@k

If you were doing research on the a particular topic and you typed only the topic name into a search engine, you might find that out of the first 10 articles that come back, only 4 are related to him. Since out of the 10 articles that were returned, only 4 were relevant, you can calculate Precision@10 with:

$$\textbf{Precision@10} = \frac{\textbf{Relevant Items}}{\textbf{Viewed Items}} = \frac{n}{k} = \frac{4}{10}$$

In other words k is just the number of articles that you looked at, and Precision@k is the percentage of those articles that are relevant to you. As we are already familiar with precision, this is the exact same calculation, only we limit which items we include to only the first k after ordering them in some sensible way.

2.1.2 Mean Average Precision (MAP)

To understand mean average precision first let us understand average precision. For a particular query (q_i) let us say we get k documents and out of these k documents n documents were relevant. Let us call the ranks of these n relevant documents as: $r_1, r_2, ..., r_n$. To calculate the average precision for a query we will calculate $precision@r_1, precision@r_2, ..., precision@r_n$ and take average of these values. This will be called the average precision for q_i.

$$\textbf{Average Precision}(\mathbf{q_i}) = \frac{\textbf{precision@}\mathbf{r_1} + \textbf{precision@}\mathbf{r_2} + ... + \textbf{precision@}\mathbf{r_n}}{\mathbf{n}}$$

Now we calculate this average precision for multiple queries and take a mean of those values to get the MAP.

2.1.3 Mean Reciprocal Rank (MRR)

Let us assume a scenario where there is only one relevant result like a case where user is searching for a fact. In such cases the time the user takes to get to the first and only relevant document is proportional to the rank of the document. We can say that higher the rank of the document is the better our system is performing. For a particular query (q_i) let us say that the relevant document appears at position k_i then the reciprocal rank of this result is $\frac{1}{k_i}$. To calculate the MRR we calculate this reciprocal rank for various queries and take an average. So for n queries the MRR would be:

$$\mathbf{MRR} = \mathbf{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i}}$$

2.2 Multiple Levels of Relevance

Here a document can have multiple graded levels of relevance so we need a different measuere that is where NDCG comes in.

2.2.1 Normalised Discounted Cumulative Gain (NDCG)

NDCG is measure of ranking quality that is often used to measure effectiveness of web search engine algorithms or related applications
If we are to understand the NDCG metric accordingly we must first understand CG (Cumulative Gain) and DCG (Discounted Cumulative Gain), as well as understanding the two assumptions that we make when we use DCG and its related measures:

- Highly relevant documents are more useful when appearing earlier in the search engine results list.
- Highly relevant documents are more useful than marginally relevant documents, which are more useful than non-relevant documents.

Cumulative Gain: If every recommendation has a graded relevance score associated with it, CG is the sum of graded relevance values of all results in a search result list.

$$\mathbf{CG_p} = \mathbf{\sum_{i=1}^{p} rel_i}$$

The Cumulative Gain at a particular rank position p, where the rel_i is the graded relevance of the result at position i.

Discounted Cumulative Gain: The problem with CG is that it does not take into consideration the rank of the result set when determining the usefulness of a result set. To overcome this we introduce DCG. DCG penalizes highly relevant documents that appear lower in the search by reducing the graded relevance value logarithimically proportional to the position of the result.

$$\mathbf{DCG_p} = \mathbf{\sum_{i=1}^{p}\frac{2^{rel_i}-1}{\log_2(i+1)}}$$

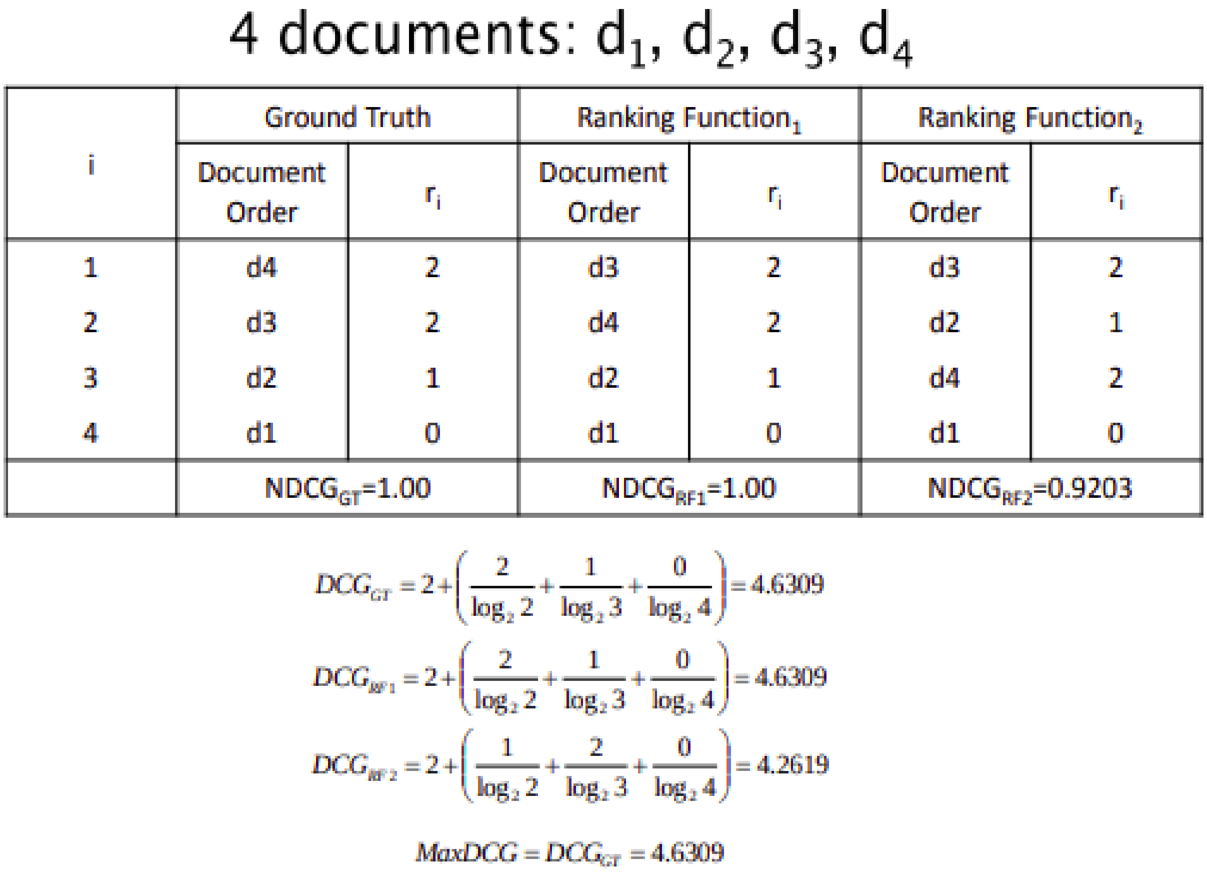
An issue arises with DCG when we want to compare the search engines performance from one query to the next because search results list can vary in length depending on the query that has been provided. Hence, by normalizing the cumulative gain at each position for a chosen value of p across queries we arrive at NDCG. We perform this by sorting all the relevant documents in the corpus by their relative relevance producing the max possible DCG through position p (a.k.a Ideal Discounted Cumulative Gain)

$$\mathbf{NDCG_p} = \mathbf{\frac{DCG_p}{IDCG_p}}$$

$$\mathbf{IDCG_p} = \sum_{\mathbf{i=1}}^{\mathbf{|REL_p|}} \frac{\mathbf{2^{rel_i} - 1}}{\mathbf{\log_2(i+1)}}$$

The ratios will always be in the range of [0, 1] with 1 being a perfect score — meaning that the DCG is the same as the IDCG. Therefore, the NDCG values can be averaged for all queries to obtain a measure of the average performance of a recommender systems ranking algorithm.

EXAMPLE:



4 documents: d_1, d_2, d_3, d_4

i	Ground Truth		Ranking Function$_1$		Ranking Function$_2$	
	Document Order	r_i	Document Order	r_i	Document Order	r_i
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
	NDCG$_{GT}$=1.00		NDCG$_{RF1}$=1.00		NDCG$_{RF2}$=0.9203	

$$DCG_{GT} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left(\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

$$MaxDCG = DCG_{GT} = 4.6309$$

Figure 1: Division into y-monotone polygons

LIMITATIONS:

- The NDCG does not penalize for bad documents in the results.
- Does not penalize missing documents in the results
- May not be suitable to measure performance of queiries that may often have several equally good results