Retrievability in IR

Half-Yearly MS Project Report

Submitted in Partial Fulfilment of the Requirements
for the evaluation of

PH5101: MS Project

by

Aman Sinha
5th Year BS-MS, 18MS065

Under Supervision of

Dr. Dwaipayan Roy
Department of Computational and Data Sciences

DPS Coordinator:

Prof. Rangeet Bhattacharyya
Department of Physical Sciences

to

Department of Physical Sciences
Indian Institute of Science Education and Research (IISER) Kolkata
Mohanpur - 741246, INDIA

December 2022

Abstract

Our civilisation has recorded all the information including scientific discoveries and knowledge across a variety of format, with digital storage being the most prevalent in our modern world. In order to access and make use of this vast pool of knowledge and information, we rely upon effective and efficient tools of searching which comes under the purview of the field of Information Retrieval (IR). Web Search engines, like Google, Bing, DuckDuckGo, influence us everyday with their selection of websites in the search results; therefore, with this comes the responsibility to be unbiased in retrieval of websites or documents. Retrievability is one such measure which quantitatively captures the ability of a document or website to be retrieved by a retrieval model irrespective of the search query.

This masters thesis studies the retrievability bias induced by three standard retrieval models. Then, compares the retrievability of judged documents with non-judged documents, to investigate the retrievability bias in document pooling strategy while creating relevance judgement. Next, the impact of query expansion on retrievability of documents is explored. In the next semester, correlation between retrievability scores and PageRank will be investigated and use of retrievability scores to boost retrieval performance will also be explored.

Contents

Retrievability Experiments on TREC 678 corpus

The main objective of the first semester's experimentation for the masters thesis is to investigate the retrievability bias for standard retrieval models using an improved query set, then look into retrievability disparity between documents in Relevance Judgement and documents otherwise, and finally explore the impact of query expansion on retrievability of collection documents.

For the reproducibility and comparison with the original results of Azzopardi and Vinay (2008) [1], three different retrieval models are chosen for the retrievability experiment which were part of their study as well. To check the consistency of result for the retrieval models across collections, a different corpus is selected. Selection of the corpus is partly based on availability of topic file and relevance judgement, and similarity of the corpus with the corpus used by Azzopardi and Vinay (2008) [1].

The retrievability values of documents is used to investigated if the documents selected for the relevance judgement is more retrievable than rest of the documents. This can reveal bias in the relevance judgement. Finally, a selection of query expansion method is made to do retrievability analysis on, to find out whether query expansion increase or decrease retrieval bias.

This chapter covers the main experimental work performed in first semester of this thesis. All the methods, experiments and results are presented in this chapter itself.

1.1 Experimental Setup

1.1.1 Hardware and Operating System

For the experiments as well as document preprocessing, indexing, evaluations and any other experiment related computation, two different systems with following main specifications are used:

Desktop PC

- Intel Core i7-12700 12th Gen @ 2.1GHz
- 16 GB main memory
- Ubuntu 22.04.1 LTS (Jammy Jellyfish) - 5.15.0-56-generic Kernel

IISER Kolkata Dirac Supercomputer

- Intel Xeon Gold 6148 CPU @ 2.40GHz
- 128 GB main memory
- Rocks 7.0 Manzanita (CentOS 7.4)

All additional software and packages used are identical:

- Python 3.10
- PyLucene 8.8.1
- trec eval 9.0.7
- Pyserini 0.12.0

The packages listed above are not exhaustive and several other python packages are used, which will be mentioned wherever they will be used. Details about the use of these packages and any particular settings will be discussed along with the experiment details in their respective subsections.

1.1.2 Dataset

TREC style collection is especially suitable because they feature Topics and their relevance judgement Qrels, which is required to do a contrast study between retrievability r(d) values of judged documents and rest of the documents.

The collection selected is the document collection used in TREC 2004 Robust Track, often referred to as **TREC 678** corpus. The document collection for the Robust track is the set of documents on both TREC Disks 4 and 5 minus the the Congressional Record on disk 4. [1]

Source	# Docs	Size (MB)
Financial Times	210,158	564
Federal Register 94	55,630	395
FBIS, disk 5	130,471	470
LA Times	131,896	475
Total Collection:	528,155	1904

The size of the document collection is close to 2 GB with 528,155 text documents. Vocabulary size of this corpus is close to 1.5 million.

For preparing the corpus for retrievals, it needs to be indexed first. Before that, document IDs (DOCID) are identified using the XML tags and contents of a document is rest of the text in the document

[1] https://trec.nist.gov/data/robust/04.guidelines.html

except the XML tags. Now having the contents of the documents and their doc-ids, Lucene (PyLucene[2]) is used with the analyzer set as "EnglishAnalyzer" (which performs basic text preprocessing and porter stemming) to index the corpus. Resulting index of the corpus created by lucene is of the size 1.9 GB (notice that the corpus size was also about 1.9 GB) and can be easily accessed using Lucene's IndexReader and IndexSearcher classes.

1.1.3 Retrieval Models

TFIDF, Okapi BM25, Language Model with Dirichlet Smoothing (LMDir) with $\mu = 1000$ are selected for retrievability study of standard retrieval models, as these retrieval models were also used by Azzopardi and Vinay in their 2008 research article on Retrievability [1].

TFIDF is an old and classic IR model with no hyper-parameters. It is often included in studies alongside with other models for standard comparison.

BM25 builds upon TFIDF way of scoring with better term-frequency and document length normalization and is considered a strong baseline. BM25 has two paramters: k_1 and b. Optimal paramters is selected from evaluations against 250 topics of 678-robust: $k_1 = 0.7$, $b = 0.35$.

LMDir Language Modelling using Dirichlet Smoothing is one of commonly used IR models from language model retrieval functions. LMDir has a smoothing parameter μ that is selected to be $\mu = 1000$ in order to match with the prior study [1].

All three retrieval algorithms are pre-implemented in Lucene. Therefore, respective Lucene functions from *Similarities* class are used to perform the retrievals for these retrieval models.

[2] PyLucene is a python wrapper around the Java Lucene (https://lucene.apache.org/pylucene/). Lucene is a Java library which provides indexing and search features, as well as other related advanced functionalities (https://lucene.apache.org/)

1.1.4 Model Effectiveness

Performance of each retrieval model on TREC 678 corpus with respective TREC topics 6,7,8 and robust (250 queries) is evaluated using *trec eval*. In the table 1.1, a few key performance metrics for the 3 algorithms are reported.

TREC 678	MAP	MRR	P@5	P@20
TFIDF	0.1561	0.5257	0.3598	0.2488
BM25	0.2596	0.6766	0.4988	0.3679
LMDir	0.2526	0.6774	0.4747	0.3600

Table 1.1: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Precision at 5 and 20 documents (P@5, P@20) with respect to their TREC queries

TFIDF performs significantly poorer than BM25 and LMDir. BM25 and LMDir provides equivalent performance for TREC 678 corpus with respect to the same testing query set. Later sections in the chapter show that despite similar effectiveness, BM25 and LMDir differ in terms of their retrieval bias.

1.2 Large Scale Retrieval Simulation

Estimation of $r(\mathbf{d})$ values for each document require a number of approximations to be made. Same as the original study [1], generalized utility/cost fucntion $f(.)$ is taken as the simple binary function which just indicates the presence or absence of a document in top ranks with cutoff c and the query weight o_q to be equal and constant by setting $o_q = 1$. For each retrieval model, $r(\mathbf{d})$ values are computed over all the documents in the collection for 5 different rank cutoffs: $c = 10, 20, 30, 50, 100$.

The next choice is the set of queries to be used. Since TREC collections are test collections, availability of a user query log is not possible. Therefore, a set of artificial queries need to be generated by some means. Azzopardi and Vinay (2008) [1], for creating their query set, used uni-grams and bi-grams sampled from the documents in the collection. All unigram queries were terms in the vocabulary which occurred at least 5 times. All two term queries were bigrams which occurred at least 20 times (if number of bigrams after frequency-thresholding more than 2 million, bigrams are sorted by their frequencies and the list is truncated at 2 million). Union of these two exclusive subsets of one term and two term queries was their query set.

Using the above method, queries were generated for TREC 678 collection but were found to be consisting a lot of undesirable terms, which are unlikely to be issued by a user to an IR system, creating a noisy query set. Therefore, for constructing a set of more realistic[3] queries, a modified method is employed to filter out undesirable queries as much as possible and will be discussed in the next subsection in detail.

1.2.1 Modified Query Generation Method

Types of queries (one term and two term queries), occurrence thresholds, and truncation prescription is kept same as done by Azzopardi and Vinay (2008) [1] without exploring the impact of variation in these choices to query set and retrievability analysis results.

Unigram Query Generation Method

Following are the steps:

[3] Here, a realistic query is such a query which looks close to what a real person might enter in a search engine. For example, a query ”said to” is considered less realitic than a query ”United Nations” . A more concrete notion could be derived from comparison with query logs, but is not done here.

1. All the tokens from lucene index of TREC 678 corpus is taken and all non-alphabetical tokens are removed, giving only alphabetical words.

2. All words are then converted to lowercase and stopwords[4] are removed.

3. This is the main step responsible for filtering unlikely queries. Part-of-Speech tagging[5] is done on words and then only Nouns (tag 'NOUN') and uncategorized (tag 'X' for others) are selected and rest of the words with other tags are removed. The reasoning behind such selection is that nouns tend to represent majority of realistic queries and words such as of, which, its, would often do not add meaning to the query or the information need of the user. Below is a table of tags which are removed and their examples[6] to help put things in perspective.

Tag	**Meaning**	**English Examples**
ADJ	adjective	*new, good, high, special, big, local*
ADP	adposition	*on, of, at, with, by, into, under*
ADV	adverb	*really, already, still, early, now*
CONJ	conjunction	*and, or, but, if, while, although*
DET	determiner, article	*the, a, some, most, every, no, which*
NUM	numeral	*twenty-four, fourth, 1991, 14:24*
PRT	particle	*at, on, out, over per, that, up, with*
PRON	pronoun	*he, their, her, its, my, I, us*
VERB	verb	*is, say, told, given, playing, would*

4. Frequency of each unique word is counted and words with frequency less than 5 are removed.

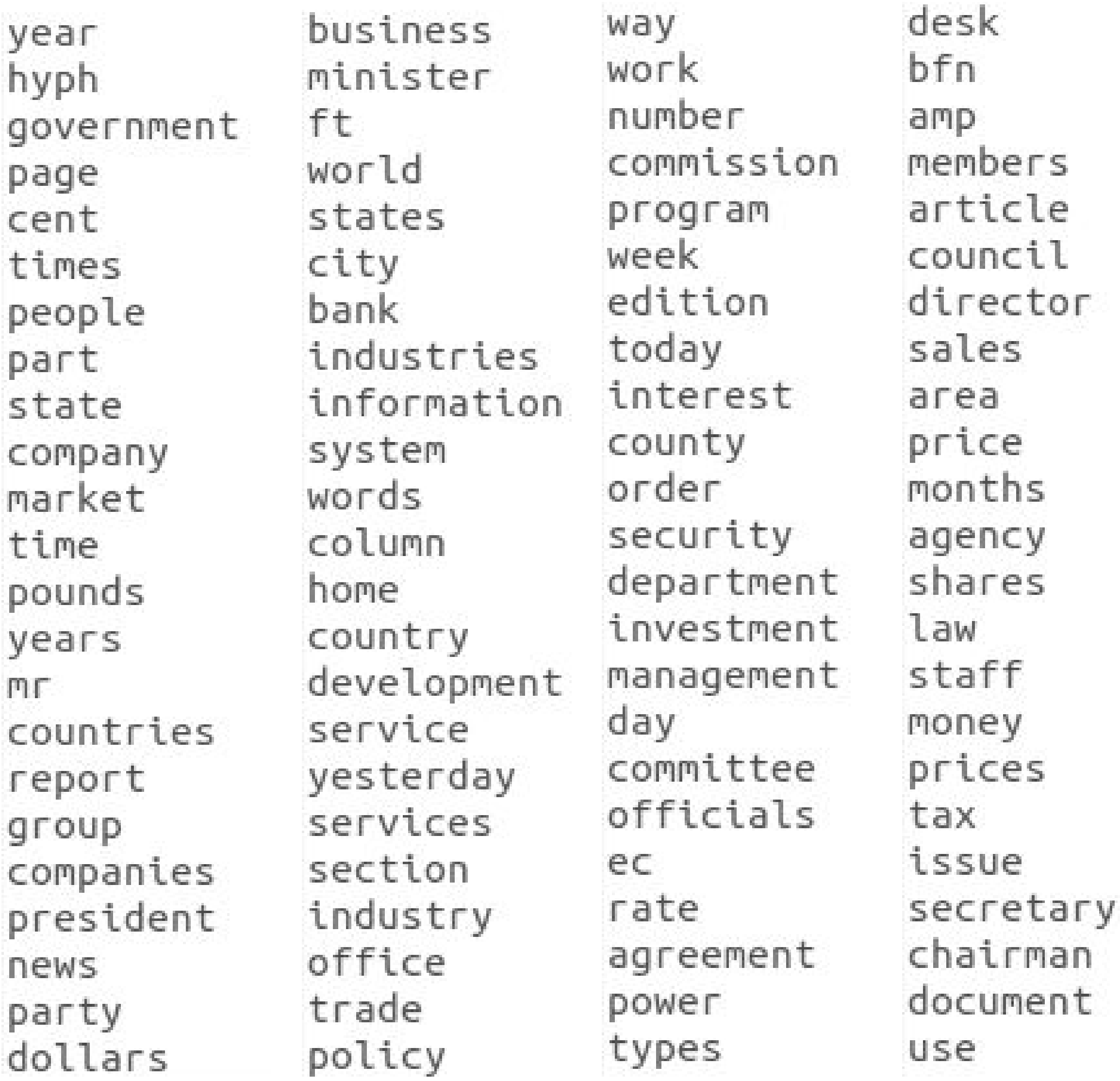
5. Words with only one character i.e. all alphabets are removed from list.

6. Now, if the number of words left in the set are more than 2 million,

[4] NLTK English Stopword set

[5] nltk.tag.pos_tag for universal tagset

[6] Taken from the NLTK book website: https://www.nltk.org/book/ch05.html



year
hyph
government
page
cent
times
people
part
state
company
market
time
pounds
years
mr
countries
report
group
companies
president
news
party
dollars
business
minister
ft
world
states
city
bank
industries
information
system
words
column
home
country
development
service
yesterday
services
section
industry
office
trade
policy
way
work
number
commission
program
week
edition
today
interest
county
order
security
department
investment
management
day
committee
officials
ec
rate
agreement
power
types
desk
bfn
amp
members
article
council
director
sales
area
price
months
agency
shares
law
staff
money
prices
tax
issue
secretary
chairman
document
use

Figure 1.1: Examples from the generated unigram query set.

words are sorted by their frequencies in descending order and truncated the list at 2 million.

All the above steps are followed for TREC 678 collection and the constructed set of words is considered to be the unigram queries, which will be posed to the IR system as one term queries. Some examples from the set of unigram queries is presented in Figure 1.1.

Bigram Query Generation Method

Following are the steps:

1. Content of each document are first blank-line tokenized[7] (to avoid two blank-line separated sentences, with the first sentence ending with no punctuation, getting considered one sentence) and then Punkt sentence tokenization[8] is done to get the sentences from each document.

2. Each sentence from all the documents are then word tokenized. Same as unigram query generation method, non-alphabetical tokens and stopwords are removed. All the remaining words are lowercased.

3. Pairs of consecutive words from each sentences from all the documents in the collection are extracted as bigrams.

4. Again, as previously done for unigram query generation, Part-of-Speech tagging is done for both the words in bigrams and then the bigrams having both words tagged only as either 'NOUN' or 'X' is retained and rest of the bigrams are discarded.[9]

5. Bigrams whose one of the word is just single character (an alphabet) is removed.

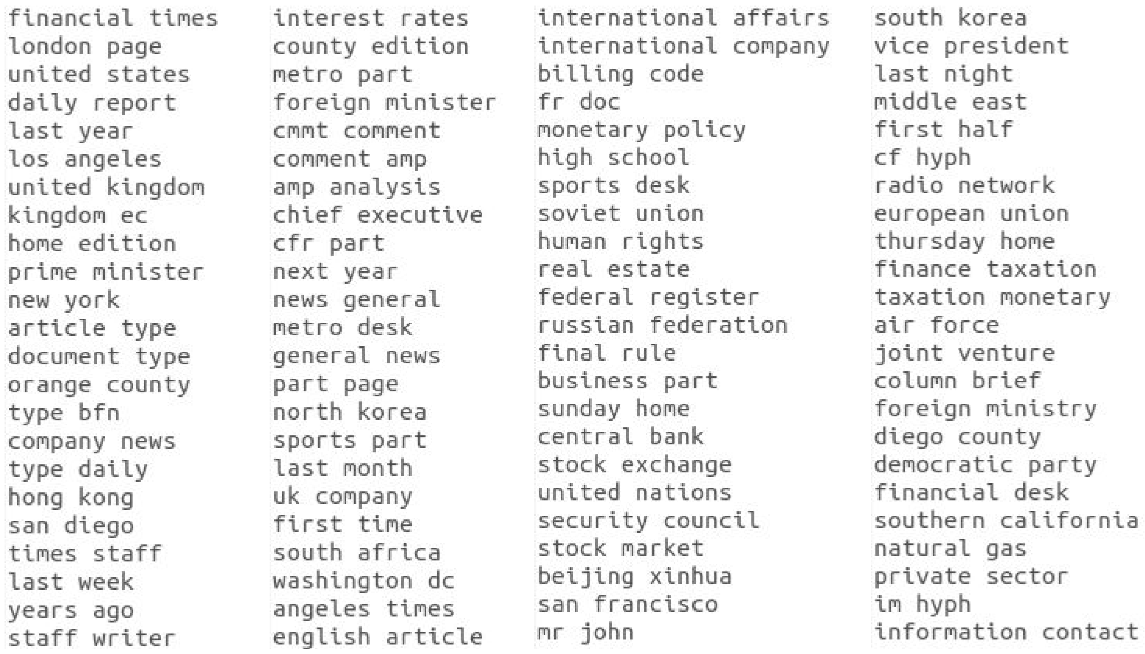
6. Occurrence count of each bigram is done and bigrams with frequency less than 20 is removed.

7. Again, same as in the last step of unigram query generation method, if the number of bigrams left in the set are more than 2 million, bigrams are sorted by their frquencies in descending order and list is truncated at 2 million bigram queries.

[7] nltk.tokenize.regexp.blankline_tokenize

[8] nltk.tokenize.sent_tokenize

[9] When this step was omitted, bigrams like "said to", "company of" (which is equivalent to 'company' and is part of unigram queries anyway), "come here" appeared which are considered to be undesirable in this thesis.



```
financial times
london page
united states
daily report
last year
los angeles
united kingdom
kingdom ec
home edition
prime minister
new york
article type
document type
orange county
type bfn
company news
type daily
hong kong
san diego
times staff
last week
years ago
staff writer
interest rates
county edition
metro part
foreign minister
cmmt comment
comment amp
amp analysis
chief executive
cfr part
next year
news general
metro desk
general news
part page
north korea
sports part
last month
uk company
first time
south africa
washington dc
angeles times
english article
international affairs
international company
billing code
fr doc
monetary policy
high school
sports desk
soviet union
human rights
real estate
federal register
russian federation
final rule
business part
sunday home
central bank
stock exchange
united nations
security council
stock market
beijing xinhua
san francisco
mr john
south korea
vice president
last night
middle east
first half
cf hyph
radio network
european union
thursday home
finance taxation
taxation monetary
air force
joint venture
column brief
foreign ministry
diego county
democratic party
financial desk
southern california
natural gas
private sector
im hyph
information contact
```

Figure 1.2: Examples from the generated bigram query set.

Following the above steps, bigram query set is generated from TREC 678 collection, which will be posed to the IR system as two term queries. Some examples of bigrams in this set is given in Figure 1.2.

Query Set

Unigram queries and bigram queries together form the query set that is used for retrievals. Below is the number of queries in the query set and in each subset:

	No. of queries
Unigram queries	137,029
Bigram queries	447,183
Query Set (Unigram + Bigram)	584,212

1.2.2 Retrievals and document retrievability

A large scale retrieval simulation is conducted on the Desktop PC mentioned in Section 1.1.1 by posing all the queries from the above query set and retrieving top 100 documents for each retrieval model. From the retrieval results, frequency of each document for all the search results in top c = 10, 20, 30, 50 & 100 rank is counted which gives the estimate of $r(\mathbf{d})$ for each document $\mathbf{d}$ in the collection for each retrieval model. Low c threshold would represent web search by users more accurately, whereas, high c threshold would represent expert users in prior art search or precedent retrieval jobs.

1.3 Trends in document retrievability

First observation from examining the $r(\mathbf{d})$ values corresponding to any retrieval model, is that some documents have retrievability score disproportionately high, and on the contrast, many documents are not retrieved even once.

Following the Retrievability Analysis Framework described in Section **??**, Lorenz curve and Gini-coefficient is used to study the inequality in retrievability scores in this huge collection of documents. Lorenz curve visualize the inequality whereas Gini coefficient summarises the inequality in one single metric.

In the following subsections, Lorenz curve and Gini coefficient of $r(\mathbf{d})$ for each three retrieval models (TFIDF, BM25, LMDir) and each rank cutoff values c (10, 20, 30, 50 & 100) for TREC 678 collection is presented.

1.3.1 Lorenz Curves

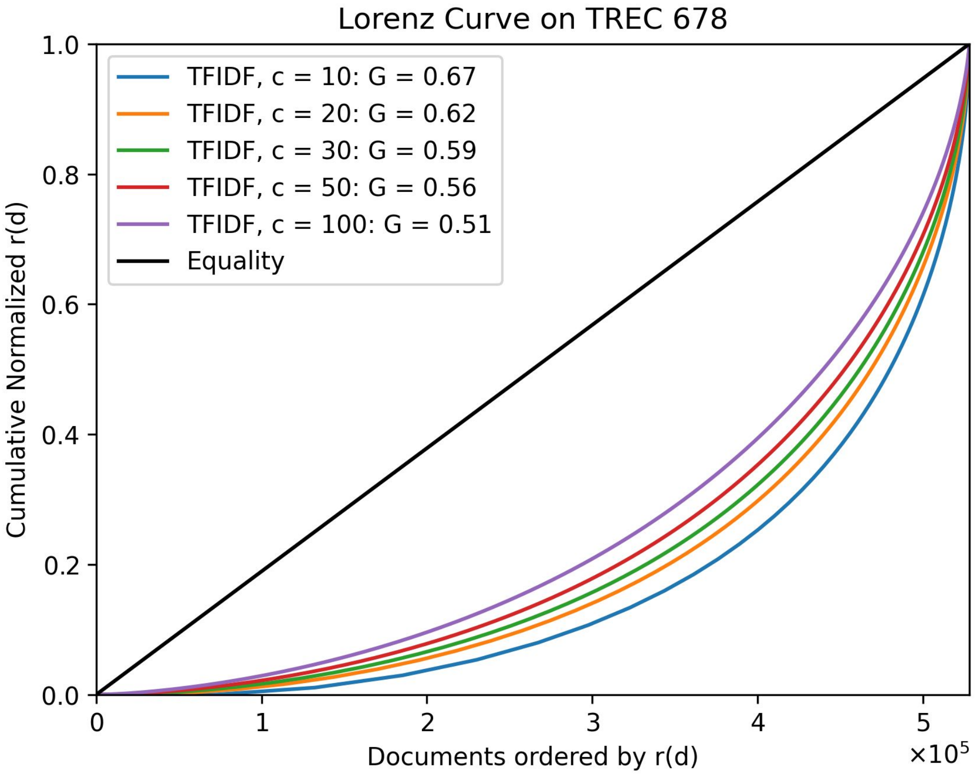


Figure 1.3: Lorenz Curve for TFIDF model.

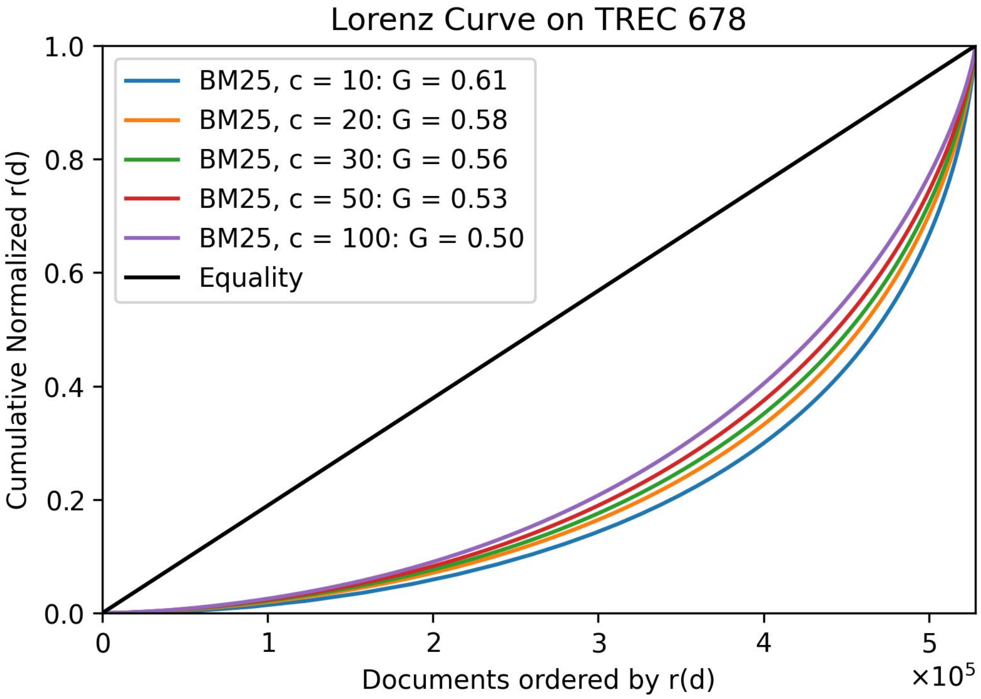


Figure 1.4: Lorenz Curve for BM25 model.

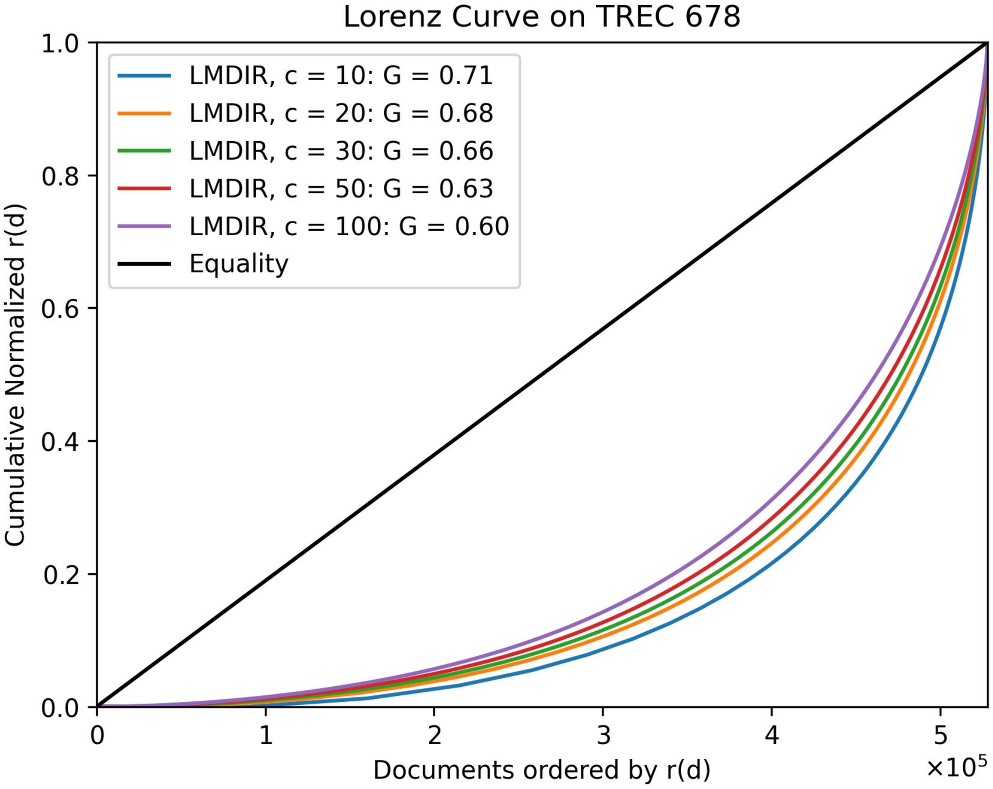


Figure 1.5: Lorenz Curve for LMDir model.

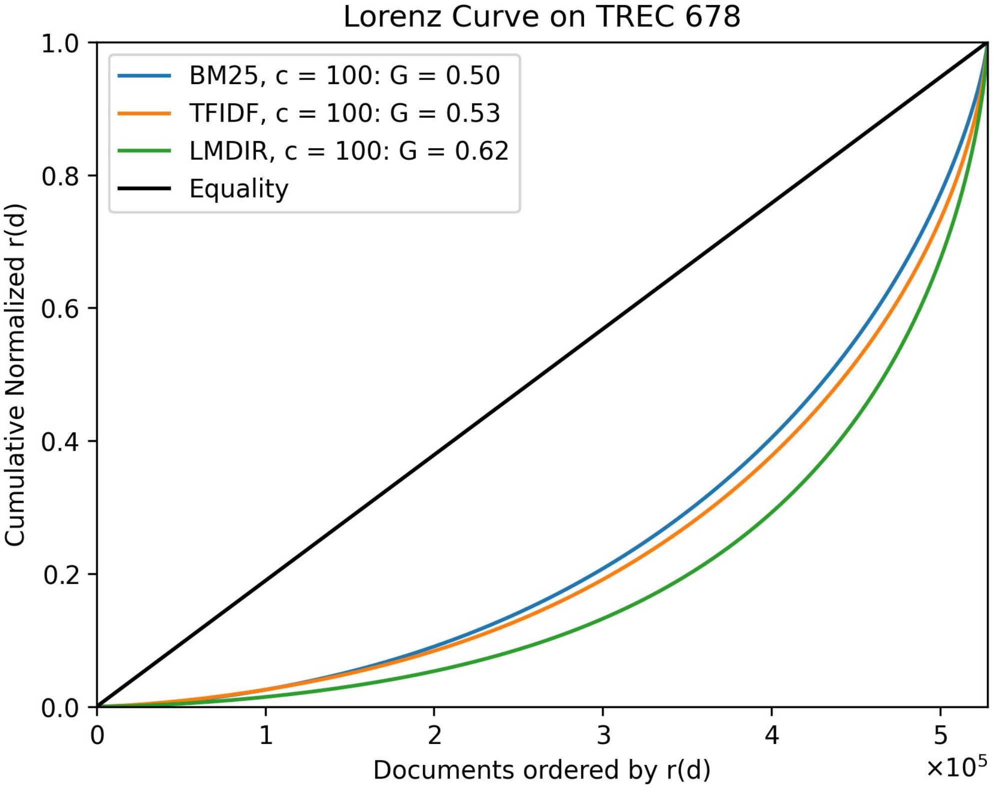
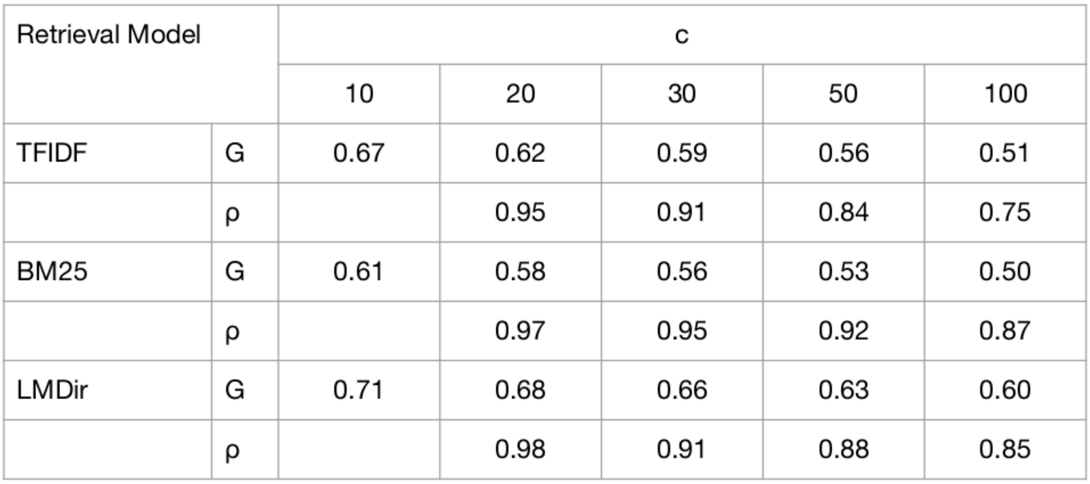


Figure 1.6: Lorenz Curve for TFIDF, BM25 and LMDir models for c = 100.

1.3.2 Gini Coefficient



Retrieval Model		c				
		10	20	30	50	100
TFIDF	G	0.67	0.62	0.59	0.56	0.51
	ρ		0.95	0.91	0.84	0.75
BM25	G	0.61	0.58	0.56	0.53	0.50
	ρ		0.97	0.95	0.92	0.87
LMDir	G	0.71	0.68	0.66	0.63	0.60
	ρ		0.98	0.91	0.88	0.85

Figure 1.7: Gini coefficient for all the retrieval models with different values of c for TREC 678 collection. Pearson's correlation coefficient ρ is calculated between $r(d)$ values of $c = 10$ and all other values of c. The relationship between all the pairs of $r(d)$ values is found to be statistically significant, thus verifying the stability of retrievability measure with respect to the choice of c.

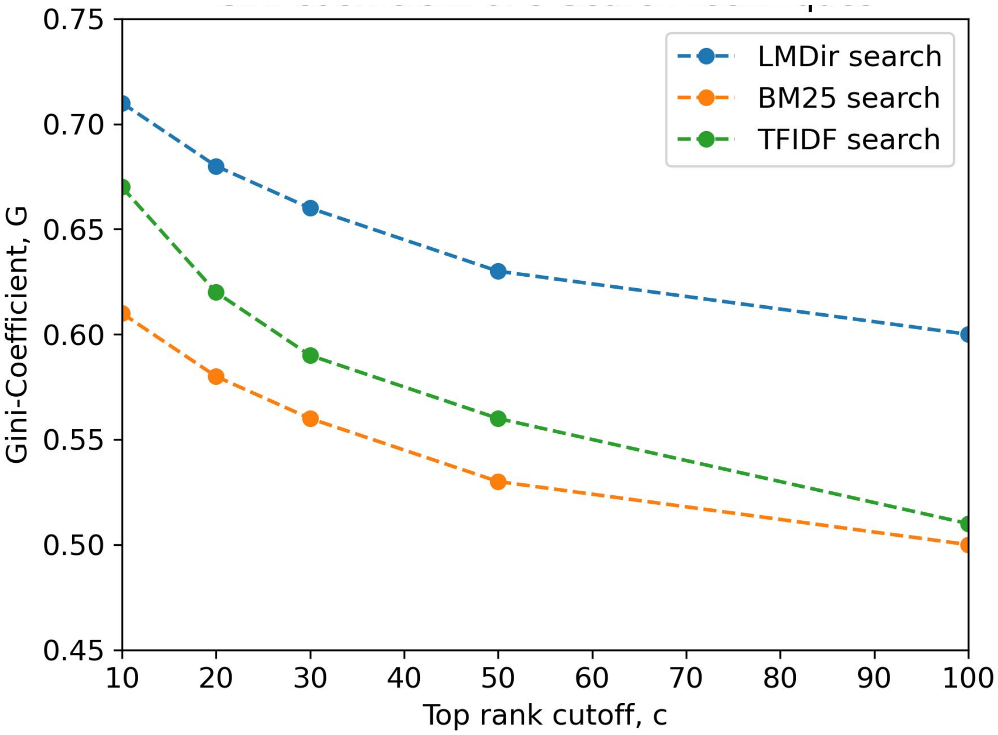


Figure 1.8: Plot of G vs c from the above table.

1.3.3 Observations

For all the retrieval models, Gini coefficient values slowly decrease as the value of c increase. This suggests that the exposure of user to the search result bias decrease as they look further down the rank list of search result. When $c = N$, where N is the no. of document in the collection, Gini coefficient becomes zero (absolutely no bias), because all the documents are retrieved all the time leading to equal $r(\mathbf{d})$ scores equal to number of queries. Whereas, if a user is only looking at a few top documents returned by a retrieval model, then the user is exposed to greater bias irrespective of the retrieval model.

BM25 is observed to induce least bias among the three retrieval models, whereas LMDir induced highest bias. Previously in Section 1.1, BM25 and LMDir was found to have equivalent performance; but now from looking at the Gini coefficient values of both models, BM25 and LMDir are not at all similar in terms of the inequality in retrievability of documents each of the model is inducing.

Another observation is that, as the c is increasing, Gini coefficient of TFIDF is catching up with the lower G value of BM25. For $c = 100$, Gini coefficient of TFIDF and BM25 is almost same, with BM25 still having a slightly lower value.

1.4 Bias in Relevance Judgement

The question is asked that whether the relevance judgement is also rigged with retrievability bias. That is to say, if the documents present in the relevance judgement have higher retrievability scores as compared to the documents which have not been included in the relevance judgement.

To address this question, relevance judgement of TREC 678 topics 678-robust (250 queries) is used. A set of document IDs present in

	Judged documents	Non-judged documents
Count	174787	353368
Mean r(d)	**131.03**	**80.15**
Std r(d)	139.21	86.21
Min r(d)	0	0
Max r(d)	3220	1534
25%	39	24
50%	93	52
75%	177	106

Table 1.2: Descriptive Statistics for $r(d)$ values (for BM25 with $c = 100$) of Judged and Non-judged documents.

the relevance judgement is formed, and a set of rest of the document IDs as non-judged documents. Descriptive statistics of retrievability values $r(d)$ for documents in the relevance judgement and rest of the non-judged documents is calculated (see Table 1.2) and then inferences are drawn from them.

From the Table 1.2, several observations can be made. First, even though the number of judged documents are about half of the number of non-judged documents, mean $r(\mathbf{d})$ of judged documents are significantly higher than non-judged documents. This suggests that there are some documents in relevance judgement which have very high retrievability scores which can be responsible for increasing the mean. Max $r(\mathbf{d})$ of judged documents is the overall highest $r(\mathbf{d})$, whereas Max $r(\mathbf{d})$ of non-judged documents is almost half of Max $r(\mathbf{d})$ of judged documents. The minimum $r(\mathbf{d})$ of both categories are zero, suggesting that judged documents do not necessarily contain only those documents which are retrieved by BM25. From the percentile information, it can be said that, although distribution of $r(\mathbf{d})$ of both judged and non-judged documents start from zero, $r(\mathbf{d})$ values' distribution is right shifted, with the shift increasing with increasing $r(\mathbf{d})$.

The relevance judgement is tending to be biased towards more retrievable documents, irrespective of the queries. This is a cause of concern because retrieval model evaluations rely upon relevance judgements in the Cranfield paradigm. The favoritism of relevance judgement towards documents with higher retrievability scores means that retrieval models preferring highly retrievable documents will get higher evaluation scores, due to which not only a retrieval model with higher bias will be deemed better but also the performance metrics calculated using such relevance judgement will be inaccurate; subsequently, if such a retrieval model is used to pool documents for creating any new relevance judgement, then the bias in relevance judgements will add up over time. Therefore, this observation of retrievability bias in constituent documents of relevance judgement has implications for the document pooling strategies devised or used to create relevance judgements.

1.5 Retrievability after RM3 Query Expansion

Investigation of the impact of query expansion on retrievability of documents and overall retrievability bias is carried out for **RM3** query expansion technique.

Initial retrieval is done using BM25 (with k_1 and b parameters same as before). RM3 is used to expand the queries with top 10 docs as pseudo-relevant docs, 10 expansion terms and original query weight $= 0.4$. Re-retrieval is performed again using BM25 and final retrieval results are presented.

For large scale retrieval simulation for RM3, same query set is used as before. Due to longer retrieval time taken by RM3, retrievals are run on IISER-Kolkata Dirac Supercomputer and document retrievability values are estimated for all the 5 c rank cutoff values (10, 20, 30, 50 & 100).

Lorenz curve is plotted and Gini coefficient is calculated for all c values.

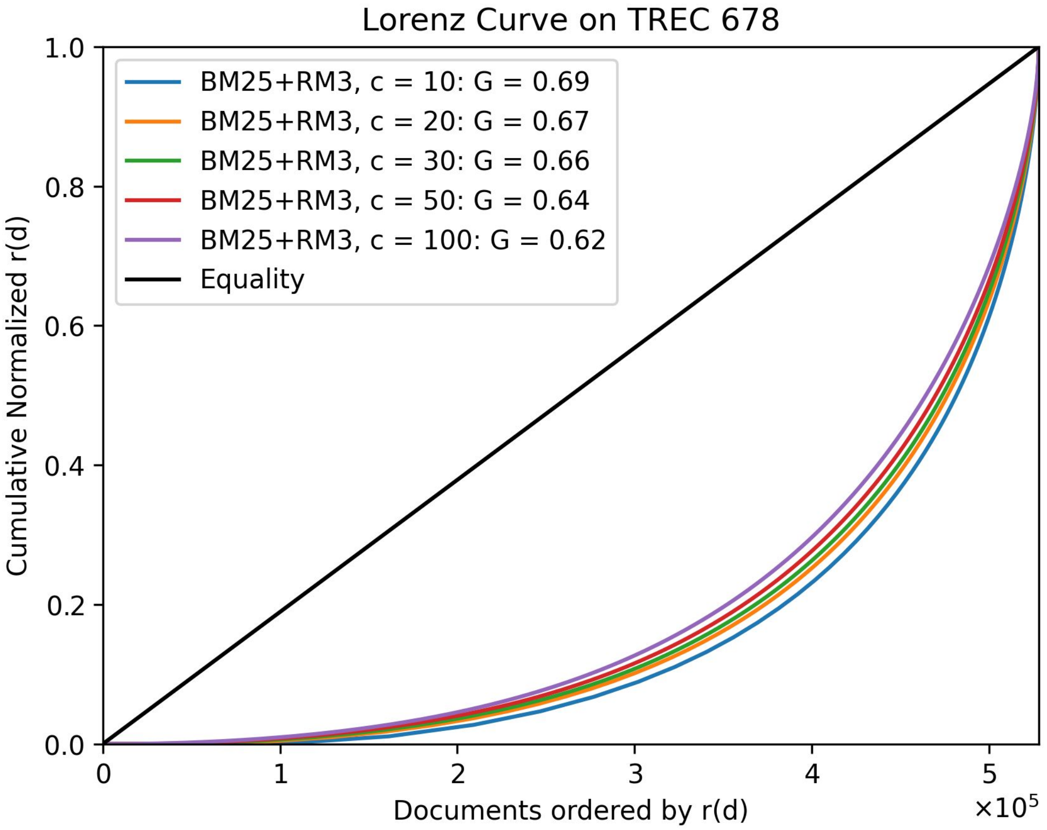
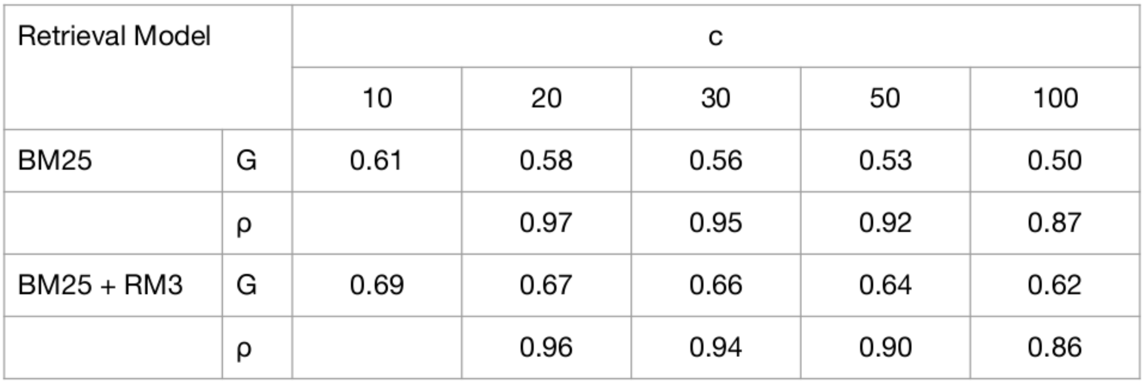


Figure 1.9: Lorenz Curve for BM25 + RM3 on TREC 678.



Retrieval Model		c				
		10	20	30	50	100
BM25	G	0.61	0.58	0.56	0.53	0.50
	ρ		0.97	0.95	0.92	0.87
BM25 + RM3	G	0.69	0.67	0.66	0.64	0.62
	ρ		0.96	0.94	0.90	0.86

Figure 1.10: Comparison between Gini coefficient calculated for BM25 and BM25+RM3 on TREC 678. Pearson's correlation coefficient ρ is also reported in the similar combination as done in Table 1.7.

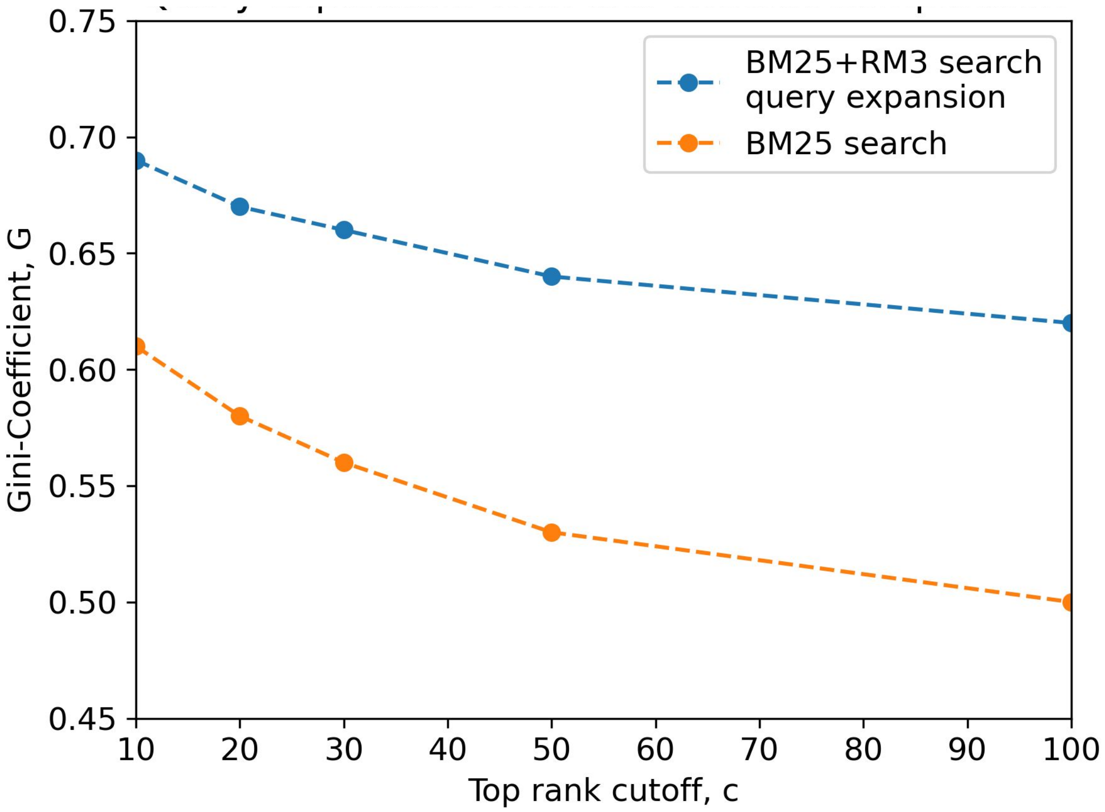


Figure 1.11: Plot of G vs c for BM25+RM3 search and BM25 search.

RM3 query expansion has increased retrievability bias. This suggests that the process of adding more terms from top documents is leading to reduction in retrievability of documents. Also, the decrease in Gini coefficient with c is slower for BM25+RM3 than BM25. Although the RM3 is known for boosting performance very well, the boost in retrievability bias that is coming along with it is concerning. This observation also highlights that increase in bias doesn't always correlate with decrease in performance, and hence the relationship between effectiveness and retrievability bias is non-trivial and not easily reducible to positive or negative correlation.

Bibliography

1. Azzopardi, L. & Vinay, V. *Retrievability: An evaluation measure for higher order information access tasks* in *Proceedings of the 17th ACM conference on Information and knowledge management* (2008), 561–570.
2. Arantes, V. & Saddler, J. N. Cellulose accessibility limits the effectiveness of minimum cellulase loading on the efficient hydrolysis of pretreated lignocellulosic substrates. *Biotechnology for biofuels* **4,** 1–17 (2011).
3. Bashir, S. & Rauber, A. On the relationship between query characteristics and IR functions retrieval bias. *Journal of the American Society for Information Science and Technology* **62,** 1515–1532 (2011).
4. Wilkie, C. & Azzopardi, L. *A retrievability analysis: Exploring the relationship between retrieval bias and retrieval performance* in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (2014), 81–90.
5. Traub, M. C. *et al. Querylog-based assessment of retrievability bias in a large newspaper corpus* in *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)* (2016), 7–16.
6. Wilkie, C. & Azzopardi, L. *An initial investigation of query expansion bias* in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval* (2017), 285–288.
7. McLellan, C. *The relationship between retrievability bias and retrieval performance* PhD thesis (University of Glasgow, 2019).