# Retrievability and PageRank

MS Project

by

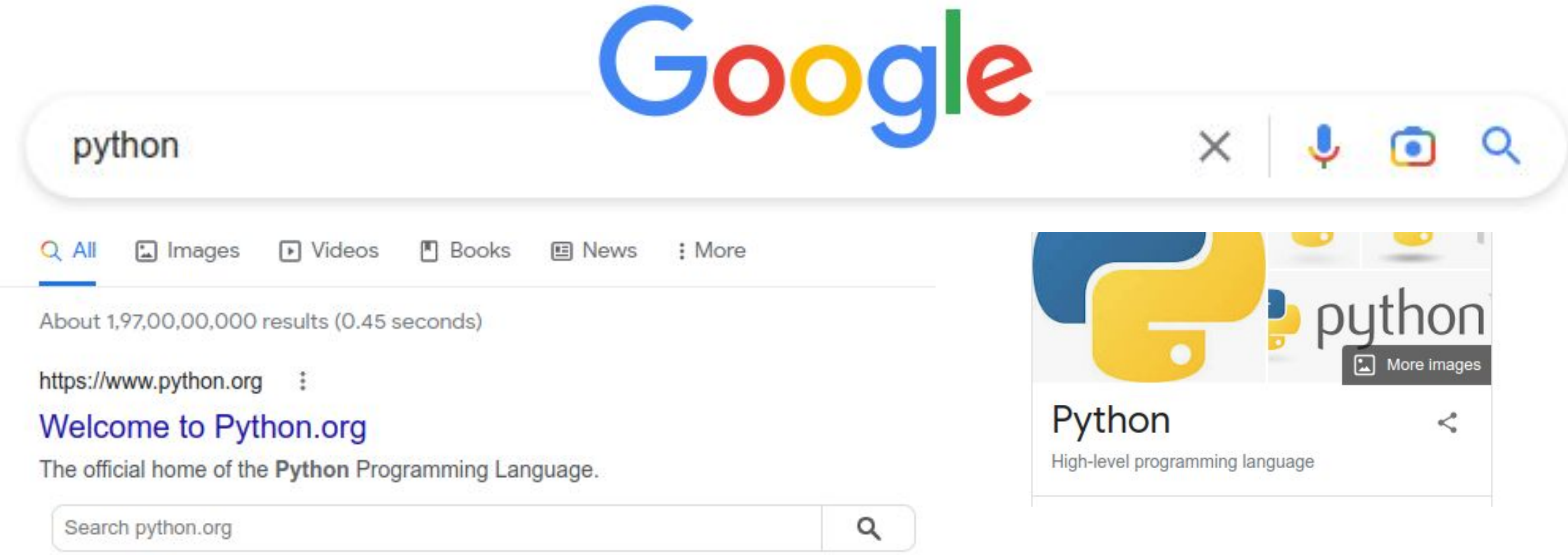
Aman Sinha

18MS065

Supervisor:
**Dr. Dwaipayan Roy**
Department of Computational and Data Sciences
IISER Kolkata

DPS Coordinator:
**Prof. Rangeet Bhattacharyya**
Department of Physical Sciences
IISER Kolkata

# Motivation




Favoritism in Search results ⇨ Bias !!!

# Motivation

- Biases in retrievals: geographical, marketing, implicit association
- Algorithmic bias from ranking function
- Positive algorithmic biases, e.g. PageRank
- Negative algorithmic biases: unintentional favouritism
- Evaluating algorithmic bias: Retrievability measure
- Can retrievability be used like/with PageRank to mitigate algorithmic bias and boost performance?

# Measure of Retrievability

Given a collection $\mathbf{D}$, an IR system accepts a user query $\mathbf{q}$ and returns a ranking of documents $\mathbf{R_q}$ from the collection $\mathbf{D}$.

Retrievability of a document $\mathbf{d}$ is a system dependent factor that measures how likely the document $\mathbf{d}$ is to be returned to the user, with respect to the collection $\mathbf{D}$ and the ranking function used by the system.

Consider $\mathbf{Q}$ as the set of all possible queries that is answerable by the collection $\mathbf{D}$. Each query $\mathbf{q} \in \mathbf{Q}$ is associated with a weight $\mathbf{o_q}$ for how likely a user will issue that query $\mathbf{q}$ to the IR system.

Then, the measure of retrievability of $\mathbf{d}$ is,

$$r(\mathbf{d}) = \sum_{\mathbf{q} \in \mathbf{Q}} o_q \cdot f(k_{dq}, c)$$

$\mathbf{f(k_{dq},c)}$ is a generalized utility/cost function where $\mathbf{k_{dq}}$ is the rank of $\mathbf{d}$ in the result for $\mathbf{q}$, and $\mathbf{c}$ is a maximum rank cutoff that a user will examine in the ranked list.

In the simplest form, cumulative scoring model, $\mathbf{f(k_{dq},c) = 1}$ if $\mathbf{k_{dq} \leq c}$, and 0 otherwise. Also, $\mathbf{o_q = 1}$

Azzopardi, Leif, and Vishwa Vinay. "Retrievability: An evaluation measure for higher order information access tasks." In Proceedings of the 17th ACM conference on Information and knowledge management, pp. 561-570. 2008.

# Retrievability score r(d) of a document d

=

how many times document d

is retrieved by the IR model

within the rank cutoff c

for the queries in universal query set Q

# Retrievability Analysis Framework

5 key steps :

1. Query set generation
2. IR model parameter selection
3. Retrievals for all the queries in the query set
4. Computing document retrievability r(d)
5. Summarising retrievability bias globally

# Query Set Generation

All possible queries for a collection D is impossible to construct, so instead, Q is a very large set of possible queries to achieve a reasonable estimate of r(d)

2 approaches to query set :

1. Real query log from a IR system (e.g., Web search engines, Library search)
2. Sampling queries from the text of documents in the collection

In the original proposal, Azzopardi and Vinay (2008) used the following method:

1. All unique unigrams that occurred ≥ 5 times
2. All uniques bigrams that occurred ≥ 20 times
3. Used the set of all these selected unigram and bigram as the query set Q

# Global Bias in Retrievability

Given the distribution of r(d) scores of all documents, we can assess the inequality between r(d) scores within a collection by using **Lorenz Curve**

**Lorenz Curve** is used to visualize the inequality of wealth in a population.

Then, computing **Gini Coefficient G** summarizes the amount of bias in the Lorenz Curve

# Lorenz Curve

Developed by Max O. Lorenz in 1905

X-axis: Individuals sorted from lowest to highest

Y-axis: Cumulative normalized sum from lowest to highest

Line of equality: when everyone has same wealth

Lorenz curve: the actual curve from the distribution of wealth

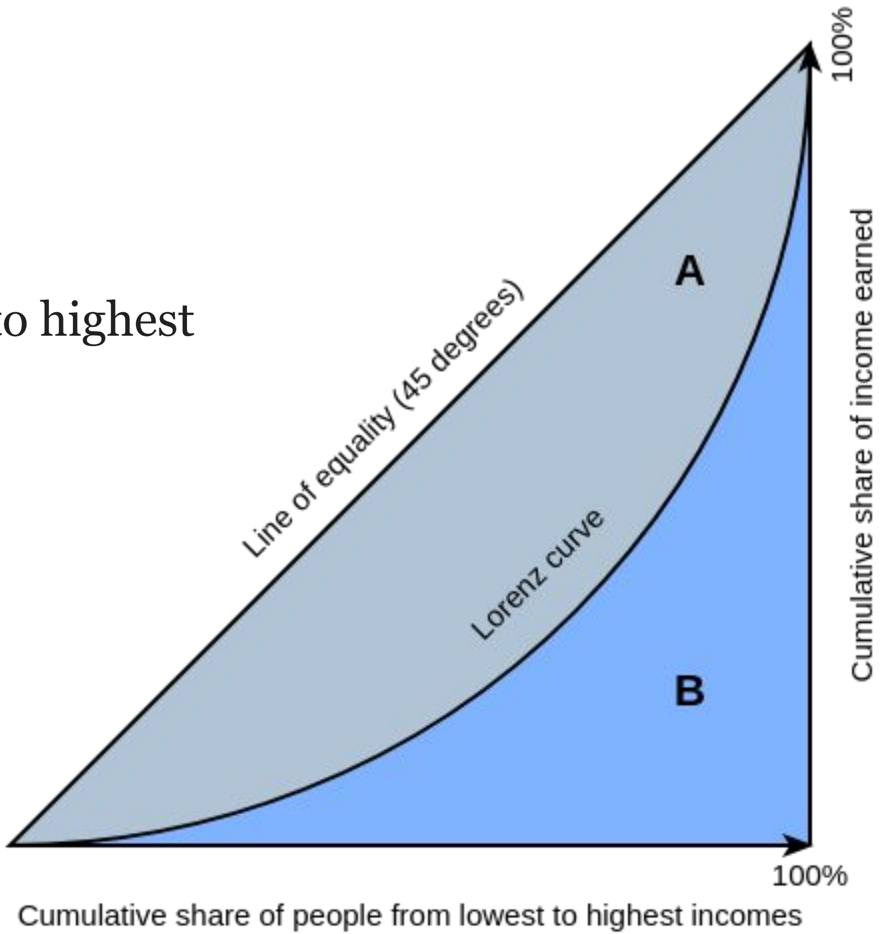
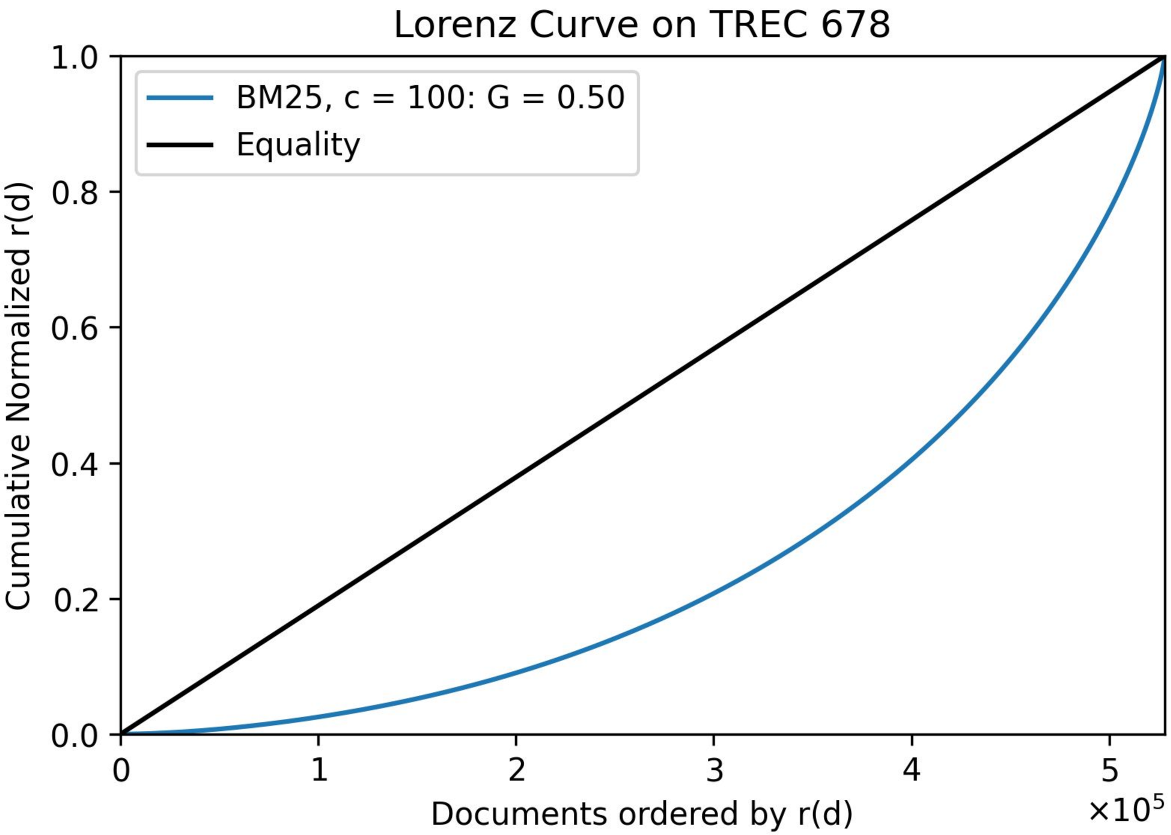



Image Source: Wikipedia. 2022. "Lorenz curve." Wikimedia Foundation. Last modified October 23, 2022. https://en.wikipedia.org/wiki/Lorenz_curve

# Lorenz Curve



# Gini Coefficient G

$$G = \frac{\sum_{i=1}^{N} (2 * i - N - 1) * r(\mathbf{d_i})}{N \sum_{j=1}^{N} r(\mathbf{d_j})}$$

where,
$r(d_i)$ is retrievability value for i-th document from documents sorted in ascending order by their r(d) values
N is the total number of documents in the collection

G = 0 : absolute equality, no bias

G = 1 : absolute inequality, highest bias; only one **d** is retrievable

# Related Work

Studies range from explorations of performance-bias relationship to applications of retrievability for clustering, query expansion and collection pruning

- **Retrievability Bias vs. Retrieval Performance** [Ref. 1,2]
  Does the retrieval algorithms that have least bias also perform better?
  *Fairness Hypothesis*
  Studies have found that in most scenarios there is a strong positive correlation
- **Estimating Retrievability** [Ref. 3,4]
  One of the biggest issue in retrievability analysis: large computational cost required to perform retrievals for millions (sometimes, billions) of queries
  Cutting no. of queries sampled from each document till it is correlated with original estimate. Found to depend on the bias of retrieval model; more biased retrieval models required less queries to reach a good estimate
  Bypassing retrievability analysis by using document features that correlate the most with retrievability estimates

# Related Work

- **Retrievability and Query Expansion (QE)** [Ref. 5,6]
  Retrievability-based clustering for relevance feedback: reduces bias in QE along with some performance improvement as well
  Reverted index for relevance feedback: queries that retrieved PRF documents as potential query expansion terms; achieves significant latency improvement and some performance improvement for QE
- **Patent Retrieval and Prior Art Search** [Ref. 7,8,9]
  Patent retrieval is largely focused on recall than precision; one missed document could lead to a hefty lawsuit for copyright infringement
  Retrievability analysis has been used to identify patents that have low retrievability using query set which better models expert users, then partitioning the corpus on that basis to provide better access
  Synthesis of hybrid retrieval models to improve access to large patent collections

# Goals

- Improve query generation method towards realistic queries
- Perform retrievability analysis using improved query set on standard retrieval models and query expansion technique along with more detailed investigation of correlation of r(d) values between models
- Study correlation between PageRank and Retrievability scores on Wikipedia articles
- Explore the amalgamation of PageRank and Retrievability scores into the retrieval models to boost performance

# Work Plan

- Survey the literature on Retrievability measure and its applications
- Retrievability experiment on TREC 678 corpus for BM25, TFIDF, LMDir retrieval models and RM3 query expansion model using a modified query generation method and AOL query log
- PageRank computation for Wikipedia articles
- Preparing a realistic query set for Wikipedia (if possible, using a query log)
- Retrievability scores computation for Wikipedia articles
- Investigation of Correlation between PageRank and Retrievability using Wikipedia dataset
- Combining PageRank and Retrievability to boost retrieval performance

# Retrievability Experiment on TREC 678 corpus

Document Collection -
**TREC disks 4 and 5** minus Congressional Records on disk 4
(referred as TREC 678 collection)

- Collection size (in GB) ~ 2 GB
- Number of documents = 528,155
- Vocabulary = 1,502,031 ~ 1.5 M

| 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 |

Apache **Lucene** and PyLucene (its python-wrapper) is used to index and search the collection
NLTK python toolkit is used for tokenizations

# Query Generation Method

Query set generated comprise of two subsets:

1. Unigram queries
2. Bigram queries

Both are extracted from the corpus documents

# Retrievability Experiment on TREC 678 corpus

## Query Generation Method

### Unigram Queries Generation Method

Steps:

1. All the document texts are tokenized and non-alphabetical tokens are removed
2. Words are converted to lowercase and stopwords are removed
3. Part-of-Speech tagging is done on words and then the words with following tags are removed:

| 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* |

# Retrievability Experiment on TREC 678 corpus

## Query Generation Method

### Unigram Queries Generation Method

Steps:

4. Unique words and their frequencies are counted ( term-frequency tf )
5. Words with tf < 5  are removed
6. Words with length = 1 is removed (i.e., all alphabets)
7. Then the list of unique words are sorted in descending order and the list is truncated at 2 million unique words if length of list more than 2 million

   This is now considered as the Unigram query set

# Retrievability Experiment on TREC 678 corpus

## Query Generation Method

### Unigram Queries

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

## Query Generation Method

### Bigram Queries Generation Method

Steps:

1. All the document texts are first blank-line tokenized and then done Punkt sentence tokenization
2. Sentences are word tokenized with non-alphabetical token removal, stopword removal and lowercasing
3. All pairs of consecutive words are extracted (bigrams)
4. Part-of-Speech tagging is done for both words in bigrams and then the bigrams with any word having a tag like in unigram method is removed
5. Term-frequencies of bigrams are computed and bigrams with tf < 20 are removed
6. Bigrams with a word of length = 1 and bigrams with both words same are removed
7. Bigrams are sorted in descending order of tf and list is truncated at 2 millions if more bigrams are present. This finally gives us our bigram query set.

# Retrievability Experiment on TREC 678 corpus

## Query Generation Method

### Bigram Queries

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

# Retrievals on Query Set for TF-IDF, BM25, LMDir models

**TF-IDF**

$$\mathrm{tfidf}(t,d,D) = (1+\log f_{t,d}) \cdot \log \frac{N}{n_t}$$

**BM25**

$$score(Q,d) = \sum_{t \in Q \cap d} \frac{tf(t,d)(1+k_1)}{tf(t,d) + k_1\left(1 - b + b \cdot \frac{|d|}{avgdl}\right)} \cdot log \frac{N - df(t) + 0.5}{df(t) + 0.5}$$

**Language Model with Dirichlet Smoothing**
**LM-Dir**

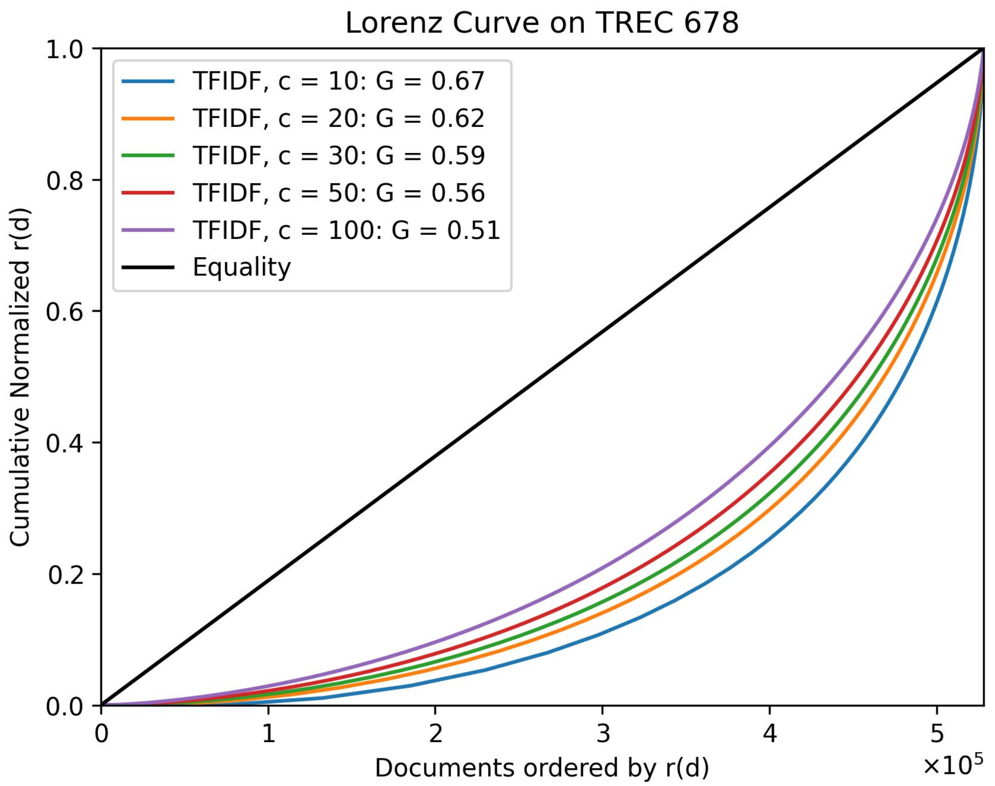
$$P_\mu(w \mid \hat{\theta}) = \frac{c(w,D) + \mu P(w \mid C)}{|D| + \mu}$$

Image Sources: Wikipedia, CS4201 (Information Retrieval and Web Search) Lecture slide, mlwiki.org

Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for TFIDF, BM25, LMDir models

# Lorenz Curve for TF-IDF model




**Observation**

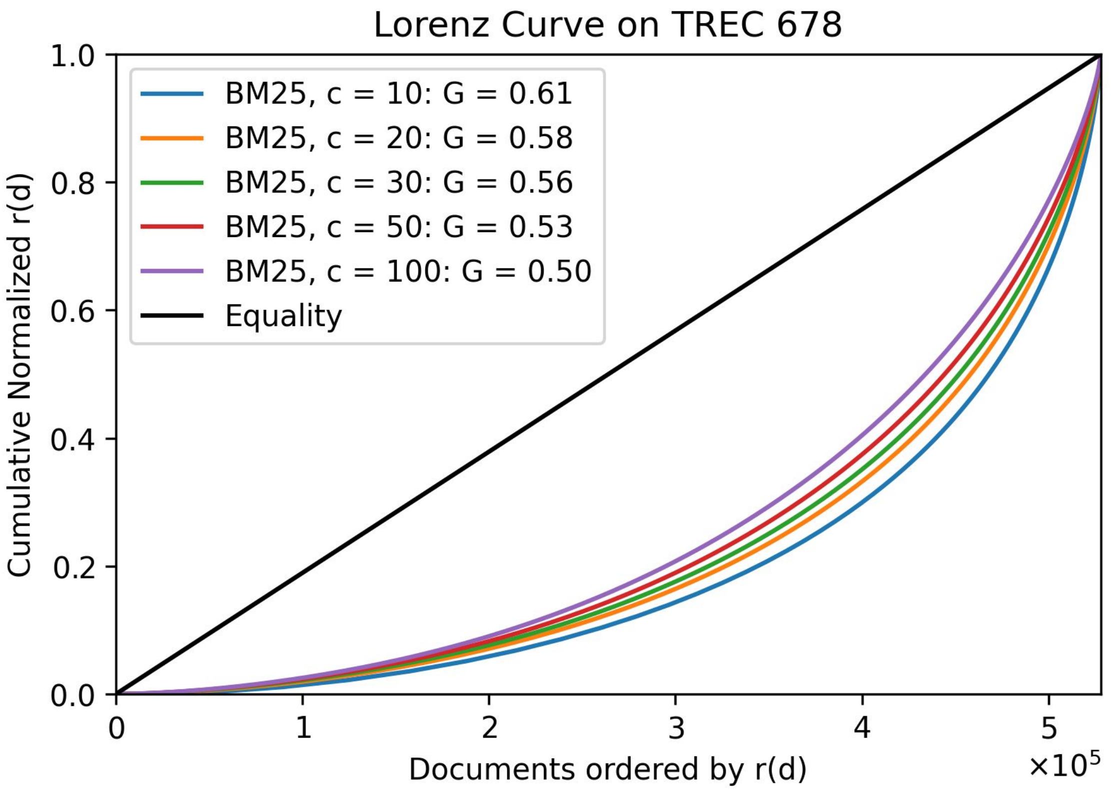
Gini coefficient G is decreasing as the rank cutoff c is increasing.

Suggesting that if explore further down the search results, the lesser we are exposed to algorithmic bias

Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for TFIDF, BM25, LMDir models

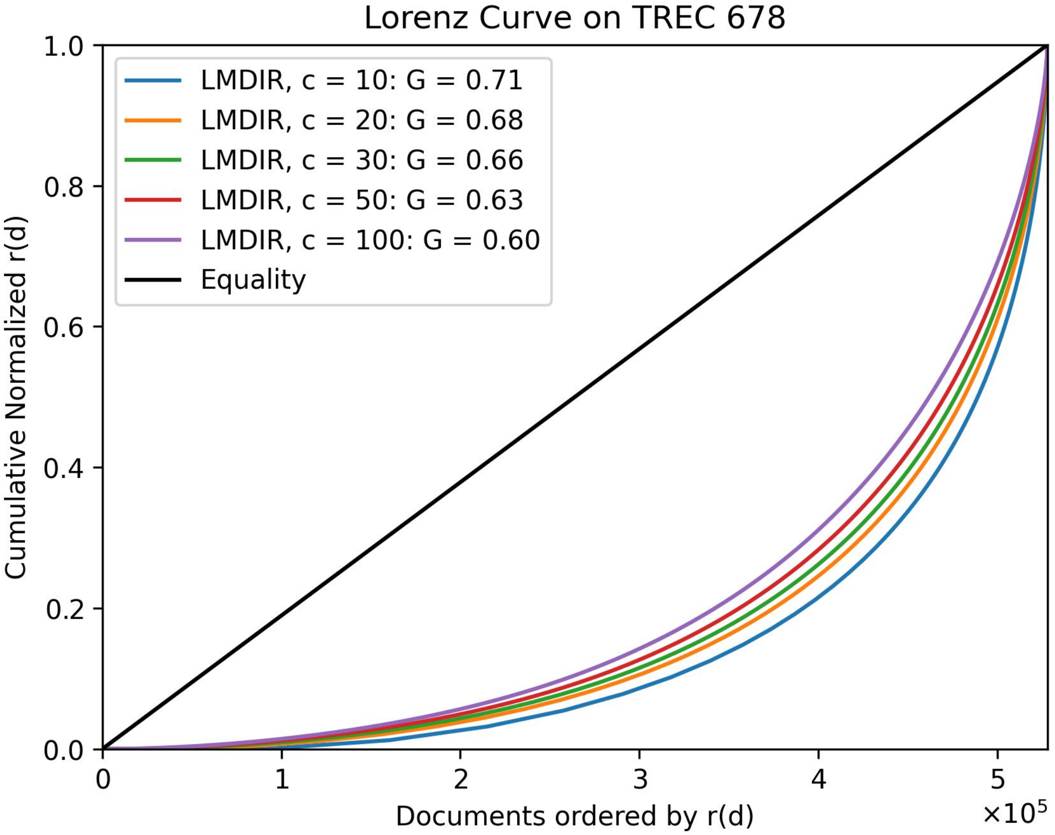
## Lorenz Curve for BM25 model ($k_1 = 0.7$, $b = 0.35$)



Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for TFIDF, BM25, LMDir models

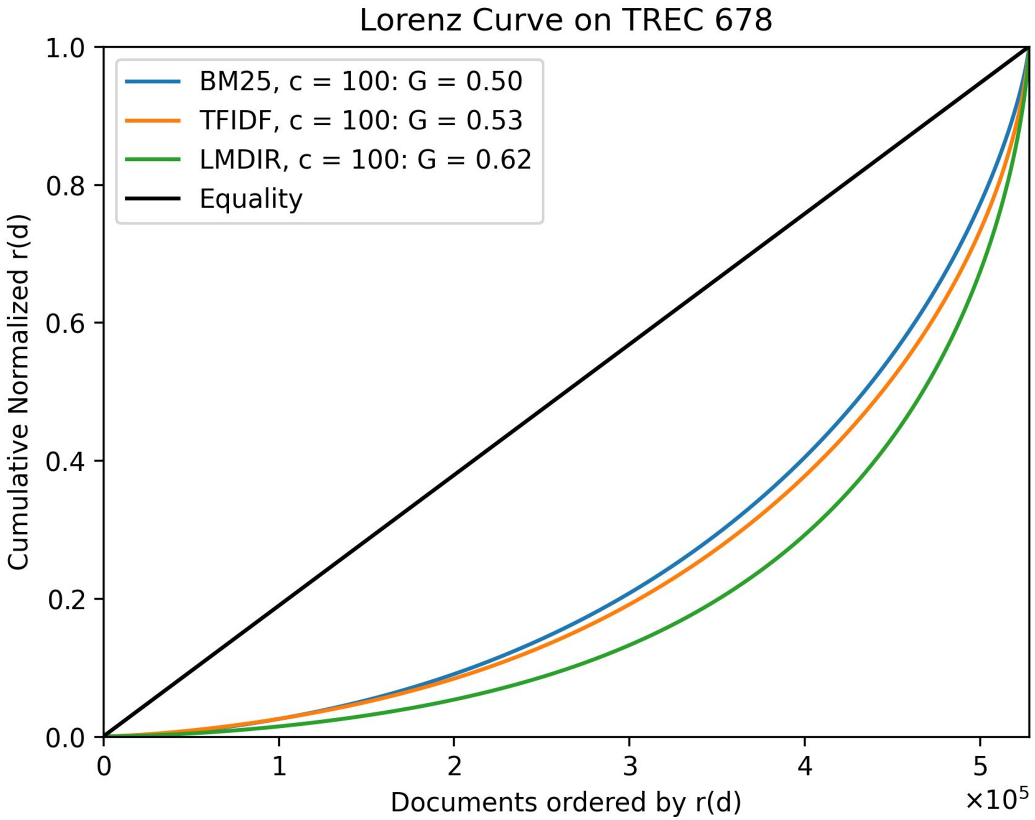
# Lorenz Curve for LMDir model (mu = 1000)



## Retrievability Experiment on TREC 678 corpus

### Retrievals on Query Set for TFIDF, BM25, LMDir models

# Lorenz Curve for c = 100




**Observation**

Bias is least for BM25 and maximum for LMDir among the 3 retrieval models explored

Even for least biased model here BM25, Gini coefficient is considerably high

Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for TFIDF, BM25, LMDir models

## Results Table

| 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 |

Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for TFIDF, BM25, LMDir models

# Retrievability of Judged vs Non-judged documents

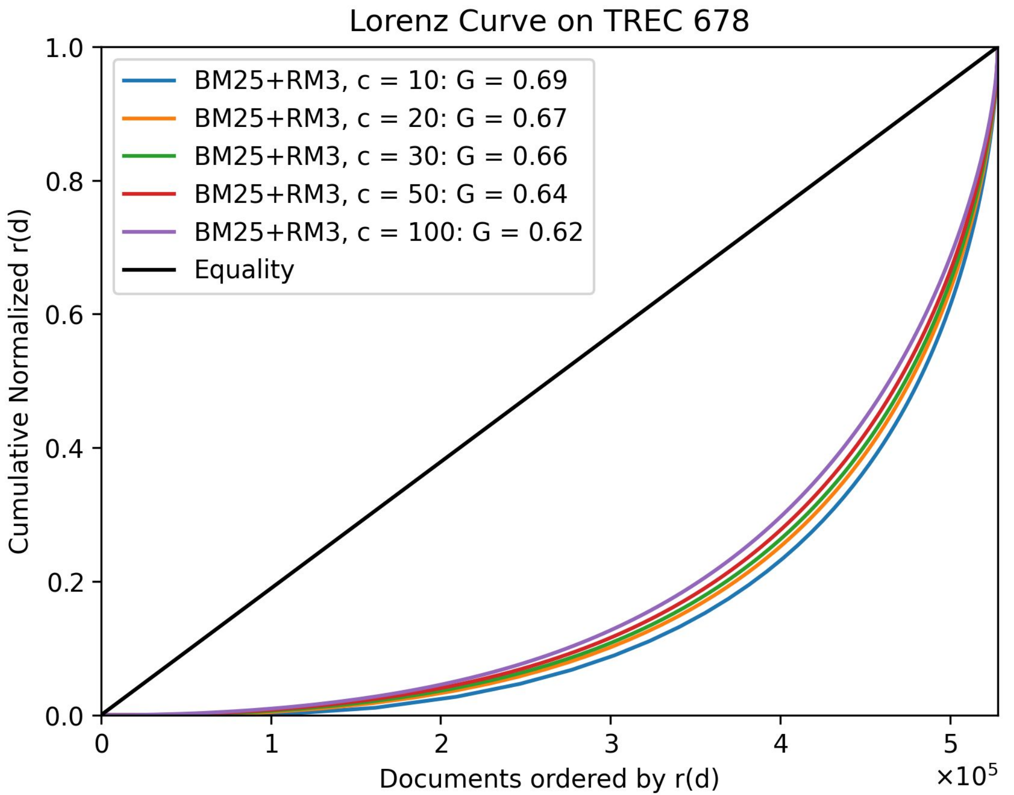
| | Judged documents | Non-judged documents |
|---|---|---|
| Count | 174787 | 353368 |
| Mean r(d) | 131.03 | 80.15 |
| Min r(d) | 0.0 | 0.0 |
| Max r(d) | 3220 | 1534 |

Judged documents tending to be biased toward highly retrievable documents

Retrievability Experiment on TREC 678 corpus

Retrievals on Query Set for RM3 Query Expansion

# Lorenz Curve for BM25+RM3 Query Expansion




**Observation**

Gini coefficient G of BM25 + RM3 is higher than BM25 alone

# Retrievability Experiment on TREC 678 corpus

## Retrievals on Query Set for RM3 Query Expansion

### Comparison Table

| 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 |

# Work Plan for Next Semester

- PageRank computation of Wikipedia articles
- Using AOL query log for Retrievability experiment on WT10G collection
- Retrievability scores of Wikipedia articles
- Investigation of Correlation between PageRank and Retrievability
- Exploration of PageRank + Retrievability to boost retrieval performance

# Practical Implications

- Assessment of bias in retrieval models will help mitigate them and avoid deploying an unintentionally biased search engine for public (which sometimes can have legal consequences)
- Implications for pooling strategies for making relevance judgement of a collection, upon which performance evaluation of IR models heavily depends
- Using the knowledge of retrievability scores to mitigate as well as boost performance; giving us a better and less biased search results

# References of Related Work section


1. Wilkie, C. and Azzopardi, L. (2013a). An Initial Investigation on the Relationship between Usage and Findability. In Advances in Information Retrieval, pages 808–811. Springer
2. Azzopardi, L. and Bache, R. (2010). On the relationship between effectiveness and accessibility. In Proc. of the 33rd ACM SIGIR, pages 889–890
3. Wilkie, C. and Azzopardi, L. (2014). Efficiently Estimating Retrievability Bias. In Advances in Information Retrieval, pages 720–726
4. Bashir, S. (2014). Estimating Retrievability Ranks of Documents Using Document Features. Neurocomput., 123:216–232
5. Bashir, S. and Rauber, A. (2009c). Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In Proc. of the 18th ACM CIKM, pages 1863–1866
6. Pickens, J., Cooper, M., and Golovchinsky, G. (2010). Reverted indexing for feedback and expansion. In Proc. of the 19th ACM CIKM, pages 1049–1058
7. Bache, R. (2011b). Patent retrieval - A question of access. World Patent Information, 33(4):345–351
8. Bashir, S. and Rauber, A. (2009a). Analyzing Document Retrievability in Patent Retrieval Settings. In Database and Expert Systems Applications, pages 753–760
9. Bashir, S. and Rauber, A. (2009b). Identification of Low/High Retrievable Patents Using Content-based Features. In Proceedings of the 2nd International Workshop on Patent Information Retrieval, PaIR ’09, pages 9–16, New York, NY, USA. ACM

# Thank You!


**Priyanshu's** happiness when he saw the CDS department board for the first time!

Note: Obviously, this slide won't be present in the final presentation