



Volume 1 · Number 1 · June 2025

Information Retrieval Research Journal
Published by RADBOUD UNIVERSITY PRESS
P.O. Box 9100, 6500 HA Nijmegen, The Netherlands
www.radbouduniversitypress.nl | radbouduniversitypress@ru.nl

ISSN: 3050-9106
E-ISSN: 3050-9114

Website: <https://irrj.org>

Cover design: Textcetera, The Hague
Print and distribution: Pumbo.nl

**RADBOUD
UNIVERSITY
PRESS**

Information Retrieval Research is published under the terms of the Creative Commons 4.0 International Licence (CC BY 4.0). This license allows you to share, copy, distribute and transmit the work; to adapt the work and to make commercial use of the work provided attribution is made to the authors.

Contents

Editorial	1
Djoerd Hiemstra, Ismail Sengor Altingovde, Solomon Atnafu, Daniela Godoy, Ben He, Makoto Kato, Shangsong Liang, Haiming Liu, Vanessa Murdock, Monica Paramita, Barbara Poblete, Negin Rahimi, Debarshi Kumar Sanyal, Johanne Trippas	
On the challenges of studying bias in Recommender Systems: The effect of data characteristics and algorithm configuration	3
Savvina Daniil, Manel Slokom, Mirjam Cuper, Cynthia Liem, Jacco van Ossenbruggen, Laura Hollink	
Don't Use LLMs to Make Relevance Judgments	29
Ian Soboroff	
Search and Society: Reimagining Information Access for Radical Futures	47
Bhaskar Mitra	
Supporting Evidence-Based Medicine by Finding Both Relevant and Significant Works	93
Sameh Frihat, Norbert Fuhr	
Annotative Indexing	109
Charles L. A. Clarke	
Graph Embeddings to Empower Entity Retrieval	137
Emma J. Gerritse, Faegheh Hasibi, Arjen P. de Vries	

Editorial board

Editor-in-chief

Djoerd Hiemstra (Radboud University, The Netherlands)

Associate editors

Ismail Sengor Altingovde (Middle East Technical University, Türkiye)

Solomon Atnafu (Addis Ababa University, Ethiopia)

Daniela Godoy (National Council for Scientific and Technological Research, Argentina)

Ben He (University Chinese Academy of Sciences, China)

Makoto Kato (University of Tsukuba, Japan)

Shangsong Liang (Sun Yat-sen University, China)

Haiming Liu (University of Southampton, United Kingdom)

Vanessa Murdock (Amazon, United States of America)

Monica Paramita (University of Sheffield, United Kingdom)

Barbara Poblete (DCC University, Chile)

Negin Rahimi (University of Massachusetts, Amherst, United States of America)

Debarshi Kumar Sanyal, (Indian Association for the Cultivation of Science, India)

Johanne Trippas (RMIT University, Australia)

Members

Nil-Jana Akpınar (Amazon, United States of America)

Hassina Aliane (CERIST, Algeria)

Alejandro Bellogin (University Autonomous of Madrid, Spain)

Patrice Bellot (Aix-Marseille University, France)

Caitlin Bentley (King's College London, United Kingdom)

Hadley Beresford (University of Sheffield, United Kingdom)

Gloria Bordogna (CNR-IREA, Italy)

Mohand Boughanem (IRIT Toulouse, France)

Pavel Braslavski (Nazarbayev University, Kazakhstan)

Emanuele Di Buccio (University of Padua, Italy)

Berkant Barla Cambazoglu (Kayra & Mergen Corp., Panama)

Malcolm Clark (Open University & UHI Inverness, United Kingdom)

Engin Demir (Hacettepe University, Türkiye)

Liana Ermakova (Université de Bretagne Occidentale, France)

Norma Fötsch (Radboud University, The Netherlands)

Ingo Frommholz (Modul University Vienna, Austria)

Darío Garigliotti (University of Bergen, Norway)

Matthias Hagen (Friedrich-Schiller-Universität Jena, Germany)

Dmitry Ignatov (HSE University, Russian Federation)

Jaap Kamps (University of Amsterdam, The Netherlands)

Udo Kruschwitz (University of Regensburg, Germany)

Saar Kuzi (Amazon, United States of America)

Matthew Lease (University of Texas at Austin, United States of America)
Siwei Liu (University of Aberdeen, United Kingdom)
Thomas Mandl (University of Hildesheim, Germany)
Bruno Martins (University of Lisbon, Portugal)
Parth Mehta (Parmonic, United States of America)
Massimo Melucci (University of Padua, Italy)
Franco Maria Nardini (ISTI-CNR, Pisa, Italy)
Javier Parapar (University of Coruña, Spain)
Benjamin Piwowarski (CNRS, Sorbonne Université, France)
Alisa Rieger (GESIS – Leibniz Institute for the Social Sciences, Germany)
Mark Sanderson (RMIT University, Australia)
Jialie Shen (City St George’s University of London, United Kingdom)
Farhad Shokraneh (University of Oxford, United Kingdom)
Mark Stevenson (University of Sheffield, United Kingdom)
Manos Tsagkias (Apple, United States of America)
Nicola Tonello (University of Pisa, Italy)
Md Zia Ullah (Edinburgh Napier University, United Kingdom)

Advisory board

Paul Kantor (Emeritus, Rutgers University, United States of America)
Stephen Robertson (formerly Microsoft Research, United Kingdom)

Production editor

Hanan Noij (Radboud University Press, The Netherlands)

Webmaster

Kay Pepping (KNAW, The Netherlands)

Mission & Scope

The Information Retrieval Research Journal (IRRJ) is a “diamond” open access journal that provides an international forum for the electronic and paper publication of high-quality scholarly articles in all areas of Information Retrieval. IRRJ commits to rigorous yet rapid reviewing. All published papers will be freely available online. Final versions are published electronically immediately upon receipt with a DOI (ISSN 3050-9114). Paper volumes are published and sold by Radboud University Press (ISSN 3050-9106). IRRJ does not charge article processing costs and aims to support researchers from low-income countries that currently have a hard time engaging with the field. IRRJ seeks unpublished papers on information retrieval research grounded in statistics, machine learning, linguistics, the cognitive sciences, and perhaps other related research fields, such as recommender systems. Papers may contain:

- new principled algorithms with sound empirical validation, and with justification of mathematical, theoretical, or psychological nature;
- experimental and/or theoretical studies yielding new insight into the design and behavior of information retrieval systems, including user-centric studies;
- reproducibility studies and applications of existing techniques that highlight the strengths and weaknesses of current methods;
- formalization of new information retrieval tasks (e.g., in the context of new applications) and methods for assessing the performance of those tasks;
- new evaluation approaches, including responsible information retrieval, fairness and non-discrimination in search;
- review and survey papers that contribute to the understanding of the state of the art in information retrieval.

Editorial

Djoerd Hiemstra

Radboud University, The Netherlands

DJOERD.HIEMSTRA@RU.NL

Ismail Sengor Altingovde

Middle East Technical University, Türkiye

ALTINGOVDE@CENG.METU.EDU.TR

Solomon Atnafu

Addis Ababa University, Ethiopia

SOLOMON.ATNAFU@AAU.EDU.ET

Daniela Godoy

National Council for Scientific and Technological Research, Argentina

DANIELA.GODOY@ISISTAN.UNICEN.EDU.AR

Ben He

University Chinese Academy of Sciences, China

BENHE@UCAS.AC.CN

Makoto Kato

University of Tsukuba, Japan

MPKATO@SLIS.TSUKUBA.AC.JP

Shangsong Liang

Sun Yat-sen University, China

LIANGSHS5@MAIL.SYSU.EDU.CN

Haiming Liu

University of Southampton, United Kingdom

H.LIU@SOTON.AC.UK

Vanessa Murdock

Amazon, United States of America

VMURDOCK@AMAZON.COM

Monica Paramita

University of Sheffield, United Kingdom

M.PARAMITA@SHEFFIELD.AC.UK

Barbara Poblete

DCC University, Chile

BPOBLETE@UCHILE.CL

Negin Rahimi

University of Massachusetts, Amherst, United States of America

RAHIMI@CS.UMASS.EDU

Debarshi Kumar Sanyal

Indian Association for the Cultivation of Science, India

DEBARSHI.SANYAL@IACS.RES.IN

Johanne Trippas

RMIT University, Australia

J.TRIPPAS@RMIT.EDU.AU

We are proud to introduce the first issue of the Information Retrieval Research Journal (IRRJ). IRRJ is the only peer-reviewed diamond open access journal that focuses exclusively on the information retrieval research community. The journal provides free and unrestricted on-line open access to papers in information retrieval, and runs fully on volunteer work by editors, reviewers, a production editor, a webmaster, and an advisory board. IRRJ does not require subscription fees nor article processing fees: At IRRJ the readers do not pay and the authors do not pay either. Instead, IRRJ plans to be completely self-funded, running on micro-donations, using resources and infrastructure provided by friend organizations and universities. We are grateful to the Radboud University and Royal Netherlands Academy of Arts and Sciences (KNAW) for providing the initial funding and infrastructure.

We believe that creating the opportunity to read and publish scientific research papers without excessive fees is of utmost importance for building a diverse and inclusive community. Currently, information retrieval papers are published at journals and conferences such as the Proceedings of ACM SIGIR, ACM Transactions on Information Systems (TOIS), Elsevier's Information Processing and Management (IP&M) and the Proceedings of ECIR (by Springer). These papers are either behind paywalls or require the authors to pay for open access, currently about \$1000 for a SIGIR paper, \$1800 for a TOIS paper, \$2890 for an ECIR paper, and no less than \$3670 for an IP&M paper. ACM, Elsevier and Springer sign open access agreements with leading universities, whose researchers never see these fees. Researchers who work at organizations that cannot afford these agreements will likely not be able to pay the article processing fees either. They cannot truly engage with the information retrieval community. Our community has become elitist; open access fees and agreements act as gatekeeping mechanisms that exclude researchers from low-income countries in Africa, South America, South East Asia, the Middle East, Eastern Europe and many other parts of the world. IRRJ aims to open our community for researchers in low-income countries, to work towards a more diverse and inclusive information retrieval research community.

IRRJ will provide all benefits that the established publishers provide, and more, including: 1) An International Standard Serial Number (ISSN 3050-9114 for electronic issues and ISSN 3050-9106 for paper issues); 2) Digital Object Identifiers (DOIs) for all articles via Crossref; 3) A dedicated editorial board; 4) Clearly stated peer review policies; 5) An established CC-BY copyright policy, where authors retain all rights; 6) An established rolling publishing schedule: Accepted papers are published electronically immediately upon receipt. We plan to publish at least 2 paper issues each year. 7) Indexing by major academic search engines, including Scopus, Web of Science, Google Scholar, DBLP, the Directory of Open Access Journals (DOAJ), etc.; 8) A Journal Impact Factor by Web of Science, to be established within three years; 9) An established archiving policy using Portico and the Public Knowledge Project (PKP) Preservation Network to create permanent archives for the purposes of preservation and restoration.

To conclude, we founded the IRRJ to open up the current publishing practices of ACM, Elsevier and Springer. Diamond open access publishing is the way forward for an inclusive and diverse information retrieval research community. Last but not least, there is a recent event that led us to establish IRRJ. For about 25 years, from 1999 to 2024, the Information Retrieval Journal – founded by Kantor and Robertson (1999) and published by Kluwer and later Springer – has served as an important journal for information retrieval. Springer discontinued the Information Retrieval Journal in January 2024 and dismissed its editorial board. To make sure such an event cannot happen to IRRJ, the publisher of IRRJ (Radboud University Press) does not own IRRJ. IRRJ is the independent, self-appointed, unofficial successor to the journal founded by Paul Kantor and Stephen Robertson, and we are honoured that they agreed to form our founding advisory board.

References

Paul Kantor and Stephen Robertson. Editorial. *Information Retrieval*, 1(1):5–5, 1999. doi: 10.1023/A:1009990414324.

On the challenges of studying bias in Recommender Systems: The effect of data characteristics and algorithm configuration

Savvina Daniil

*Centrum Wiskunde & Informatica
Amsterdam, The Netherlands*

S.DANIIL@CWI.NL

Manel Slokom

*Centrum Wiskunde & Informatica
Amsterdam, The Netherlands*

M.SLOKOM@CWI.NL

Mirjam Cuper

*National Library of the Netherlands
The Hague, The Netherlands*

MIRJAM.CUPER@KB.NL

Cynthia C.S. Liem

*Delft University of Technology
Delft, The Netherlands*

C.C.S.LIEM@TUDELFT.NL

Jacco van Ossenbruggen

*Vrije Universiteit Amsterdam
Amsterdam, The Netherlands*

JACCO.VAN.OSSENBRUGGEN@VU.NL

Laura Hollink

*Centrum Wiskunde & Informatica
Amsterdam, The Netherlands*

L.HOLLINK@CWI.NL

Editor: Shangsong Liang, Djoerd Hiemstra

Abstract

Statements on the propagation of bias by recommender systems are often hard to verify or falsify. Research on bias tends to draw from a small pool of publicly available datasets and is therefore bound by their specific properties. Additionally, implementation choices are often not explicitly described or motivated in research, while they may have an effect on bias propagation. In this paper, we explore the challenges of measuring and reporting popularity bias. We showcase the impact of data properties and algorithm configurations on popularity bias by combining real and synthetic data with well known recommender systems frameworks. First, we identify data characteristics that might impact popularity bias, and explore their presence in a set of available online datasets. Accordingly, we generate various datasets that combine these characteristics. Second, we locate algorithm configurations that vary across implementations in literature. We evaluate popularity bias for a number of datasets, three real and five synthetic, and configurations, and offer insights on their joint effect. We find that, depending on the data characteristics, various configurations of the algorithms examined can lead to different conclusions regarding the propagation of popularity bias. These results motivate the need for explicitly addressing algorithmic configuration and data properties when reporting and interpreting bias in recommender systems.

Keywords: Recommender Systems, Bias, Data Synthesis, Reproducibility

1 Introduction

Recommender systems are commonly used as a tool to encode taste based on the information available, be it user history or metadata. The wide use of recommender systems necessitates critical reflection on the issues that may arise when we allow automation to dictate our exposure to information. Specifically, bias in recommender systems is a topic of interest within the scholarly community. Bias is a complex term that can refer to various types of biases associated with interactions between users and items in a given system (Chen et al., 2023).

Many studies have focused on measuring the phenomenon of *popularity bias* in collaborative filtering systems (Klimashevskaja et al., 2024; Ahanger et al., 2022; Elahi et al., 2021; Yalcin, 2021; Abdollahpouri, 2020; Zhao et al., 2022). Despite this large research effort to track and mitigate popularity bias, there is no univocal message regarding why and when it occurs. In previous work, we found that studies that measure popularity bias propagated by commonly used algorithms on benchmark datasets report varying, sometimes contradicting results (Daniil et al., 2024). This observation raises questions; is popularity bias sensitive to properties of the system that do not receive sufficient attention? Why is a seemingly simple phenomenon so hard to study? Additionally to the evaluation strategy which was the focus of our previous work, we hypothesize that two factors that also complicate bias measuring and reporting are data characteristics and algorithm configuration.

Data characteristics Benchmark datasets are useful for academic research, as they allow researchers to evaluate their hypotheses and compare their proposed debiasing methods. However, their consistent use raises concerns that relate to the dependence on the domain and source they were constructed from, and the potentiality for blind spots that stem from outdated rating behaviour. Most importantly, by reporting on types of bias on only a small set of publicly available datasets, researchers are restricted by their specific characteristics. This specificity limits the scope of research, and obfuscates the process of examining causality. In other words, it is not trivial to conclude whether the propagation of bias or lack thereof is a result of the respective algorithm’s functionality, or of certain intricate details of the user-item interactions within these datasets. Data synthesis based on assumptions can potentially assist with shedding light on the aforementioned blind spots and gaining new insights into bias in recommender systems.

Algorithm configuration Insufficient reporting of algorithm configuration leads to a reproducibility problem within research on recommender systems. Studies have shown that papers published in big conferences often do not disclose sufficient information for replication and verification (Ferrari Dacrema et al., 2021). This issue is also relevant in the bias discussion. Even relatively simple algorithmic approaches, such as neighbour-based ones, are constructed using hyperparameters and implementation choices that might affect whether bias propagation is observed. The RecSys community proposes a set of evaluation frameworks to promote reproducibility¹, but there are important configuration differences between them that often go unmentioned (Bellogín and Said, 2021). Testing the effect of algorithm configuration can be a means of reporting on bias in a comprehensive manner.

1. <https://github.com/ACMRecSys/recsys-evaluation-frameworks>

In this paper, we experiment with data characteristics and algorithm configurations and observe the effect on popularity bias. First, we look into *data characteristics* that might have an impact on popularity bias given a rating prediction and top-10 recommendation task, a common setup among recent studies on popularity bias (Naghiaei et al., 2022; Kowald et al., 2020; Kowald and Ladic, 2022). Specifically, we delve into the relation between popularity and rating, as well as the preferences of users with large profiles. We analyze these characteristics for three real datasets: a subset of Book-Crossing (Ziegler et al., 2005) constructed by Naghiaei et al. (2022), MovieLens1M (Harper and Konstan, 2015) and Epinion (Massa and Avesani, 2007). Additionally, we form a set of data scenarios by tweaking and combining these characteristics. For each scenario, we generate a corresponding synthetic dataset of ratings, based on the interactions from the subset of Book-Crossing. Second, for a set of algorithms we identify *configuration choices* that may impact whether or not popularity bias is observed. We study three widely used algorithms, UserKNN, Biased Matrix Factorisation and Deep Matrix Factorization, implemented in three frameworks recommended by ACM RecSys, LensKit (Ekstrand, 2020), Cornac (Salah et al., 2020), and Elliot (Anelli et al., 2021). We perform the recommendation process for the subset of Book-Crossing, MovieLens1M and Epinion, as well as each synthetic dataset with varied algorithm configuration choices. We apply commonly used popularity bias metrics to evaluate the recommended lists, as well as RMSE and NDCG@10 to estimate the performance of the algorithms when it comes to rating prediction and ranking.

Our results show that whether popularity bias is observed, and to what extent, depends on much more than the algorithm that was used, or the domain in which a study is carried out: all algorithms that we tested were seen to strongly propagate popularity bias in some experimental settings, while not propagating popularity bias in other settings. The same is true for datasets: all datasets led to bias in some settings and not in others. The results further clarify that whether or not popularity bias is observed depends on, firstly, specific (often unreported) configuration and implementation details of algorithms. The implication of that is that the choice for a certain framework largely impacts the outcome of a study. It also depends, secondly, on characteristics of the dataset that is studied; specifically, the relationship between rating and popularity, as well as the preferences of users with large profiles are crucial when it comes to popularity bias propagation. Finally, it is especially the interplay between algorithm configuration and dataset characteristics that determines whether popularity bias will be observed or not.

The contributions of this paper are as follows:

- a systematic investigation into the effect of data characteristics on popularity bias, by comparing results on three commonly used datasets as well as five synthetic datasets for which we control the properties.
- a systematic investigation into the effect of implementation differences, by comparing results of algorithm configurations as well as non-configurable implementation differences in well known frameworks.
- for the more interpretable algorithms, we highlight notable results among the many, to give insights into why certain combinations of dataset characteristics and algorithm configurations lead to popularity bias.

With this work, we wish to contribute to the field by highlighting and disentangling the challenges in studying popularity bias in recommender systems.

2 Related Work

In this section, we provide a brief overview of existing work on bias in recommender systems and datasets and reproducibility.

2.1 Bias in Recommender Systems

Recommender systems are not immune to bias, even when only user consumption history is fed to the model and not other information about the users or items. Edizel et al. (2019) discuss that a model might learn sensitive information like the gender of the user in the latent space, and produce recommendations that are gender-dependent, even more so than the interactions observed in the training set itself. In a survey on the topic of bias and debias in recommender systems research, Chen et al. (2023) identify three factors that contribute to bias: user behavior’s dependence on the exposure mechanism of the system, imbalanced presence of items (and users) in the data, and the effect of feedback loops. One type of bias that arises from the interaction between an algorithm and imbalanced data is popularity bias.

Popularity bias is the phenomenon where popular items (i.e., items that are frequently interacted with in the dataset) are recommended even more frequently than their popularity would warrant (Abdollahpouri and Mansoury, 2020). It is commonly believed to be caused by the long-tail distribution that often characterizes user-item interactions: most items have been rated by only a few users, and a few items have been rated by many users (Brynjolfsson et al., 2006). Various studies have reported that frequently used recommender systems algorithms are prone to propagating popularity bias existing in the dataset they were trained on (Abdollahpouri et al., 2019b; Kowald et al., 2020; Naghiaei et al., 2022). Different metrics have been proposed to quantify popularity bias (Abdollahpouri et al., 2019b, 2017). Despite the extensive literature, our understanding of why certain algorithms and datasets are more or less prone to popularity bias is limited. In a systematic work, Deldjoo et al. (2021) used regression-based modeling to explain accuracy and fairness exhibited by a set of collaborative filtering algorithms through the lens of data characteristics such as rating and popularity distribution. In this paper, we describe scenarios of data-algorithm interaction and report the results of different metrics associated specifically with popularity bias.

2.2 Datasets and Reproducibility

The way that recommender systems researchers usually test their hypotheses, novel algorithms, and metrics is by conducting experiments on one or more publicly available datasets of user-item interactions. Surveys on the topic of recommender systems research show that the pool of datasets used is small (Bobadilla et al., 2013); user behavior data from real-world applications such as media platforms is often proprietary and therefore cannot be used for benchmarking (Khusro et al., 2016). In popularity bias studies, the use of different versions of MovieLens is exceedingly common (Wang et al., 2023), which leads us to wonder whether studies can be conclusive when they are carried out solely on a few datasets. In

our approach, we include a data synthesis step that allows us to experiment with different data distributions and observe the result. Synthetic data serves multiple purposes, each with its own specific requirements and evaluation setup (Slokom and Larson, 2021). Data synthesis is a much-discussed topic in recommender systems research, often explored in the context of privacy (Slokom, 2018; Tso and Schmidt-Thieme, 2006). Additionally, studies have employed data simulation to measure bias in recommender systems under varying data properties (Bellogín et al., 2017).

The existence of datasets for training and testing is valuable for the recommender systems community; research on publicly available data is necessary in order to ensure reproducibility (Said and Bellogín, 2014). However, as noted by Cremonesi and Jannach (2021), sharing the used data is not always sufficient to ensure basic reproducibility. Studies showed that in most cases recommender systems papers presented at top-tier conferences did not provide code for their data preprocessing or hyperparameter tuning (Ferrari Dacrema et al., 2021, 2019). This is also the case in popularity bias research; studies are often not accompanied by code, and sometimes do not describe the data filtering or hyperparameter setting (Abdollahpour et al., 2019b, 2020). Therefore, concluding that an algorithm or a dataset is prone to popularity bias becomes challenging, as it is not possible to verify or falsify the claims (Cremonesi and Jannach, 2021). Research has shown that the choice of hyperparameters can highly impact quality metrics, including average popularity of the items recommended (Jannach et al., 2015). At the same time, popular frameworks seem to have important differences that translate into different performances for the same datasets given somewhat different implementations of the same algorithm (Bellogín and Said, 2021). To further explore this issue, we experiment with algorithms implemented in different libraries and with different parameter configurations and evaluate popularity bias for each of them.

3 Identifying Data Characteristics and Algorithm Configurations

In this section, we identify data characteristics and algorithm configurations that can influence popularity bias propagation in the context of a rating prediction and top-10 recommendation task. First, we locate data characteristics that can have an effect on whether popularity bias is propagated, partly inspired by the functionality of UserKNN. UserKNN is a relatively simple algorithmic approach that simulates a ‘word-of-mouth’ setting and has lower dependence on non-intuitive parameters that impact optimization (e.g., learning rate). Accordingly, we form a set of data scenarios that combine the located characteristics. Second, we inspect UserKNN and two matrix factorization algorithms, one traditional and one neural network-based, and locate configurations that can be potentially impactful for popularity bias propagation.

3.1 Data Characteristics

Whether or not popularity bias is propagated depends on how popularity manifests in the dataset at hand. We discuss the relation between rating and popularity and the preferences of influential users.

Relation Between Rating and Popularity In the context of a rating prediction task, an algorithm aims to predict a future rating for every user of every item they have not

already consumed. Given that this is done by considering the other users’ ratings, it may be that items with high average rating will be prioritized by the system. Popularity bias studies often do not disclose whether the popular items in the dataset also have high ratings, but instead assume that their frequent recommendation is solely due to their popularity. Previous work has shown the impact of high ratings on popularity bias (Yalcin, 2021).

Influential Users In the context of UserKNN, certain users may be influential because they neighbour with many other users. For example, if only two users have rated an item and they have not rated any other items, then this item will not be recommended to anyone, because the two users are not influential at all within the system. Consequently, it is interesting to investigate the notion of user influence and whether the result is dominated by the preferences of users who, because of their large profile size, are more likely to have many neighbours.

3.1.1 REAL DATASETS

Figures 1a, 1b and 1c show the aforementioned characteristics for Book-Crossing, MovieLens1M and Epinion, respectively. Specifically, they show the correlation between item average rating and item popularity among all users, as well as among the users with the 20% largest profiles. We see that for Book-Crossing and MovieLens1M there is a slight positive correlation between rating and popularity, and a negative one for Epinion. Additionally, for the first two datasets there is no obvious difference between the tendencies of all users and the ones with large profiles, while the users with large profiles in Epinion are less favourable towards the popular items than the entire set of users. Table 1 shows the number of users, items, and interactions, and sparsity for each of the datasets.

Table 1: Basic characteristics of the real datasets. The synthetic datasets share the same characteristics as Book-Crossing.

Dataset	#users	#items	#ratings	Sparsity
Book-Crossing (subset)	6,358	6,921	88,552	99.80%
MovieLens1M	6,040	3,706	1,000,209	95.53%
Epinion	22,164	296,277	912,441	99.99%

3.1.2 DATA SCENARIOS

We synthesize data that follows a long-tail distribution for items and users, as it is discussed as a prerequisite for popularity bias to occur (Brynjolfsson et al., 2006; Celma and Cano, 2008). Specifically, we choose the interactions in a subset of the Book-Crossing dataset (Naghiaei et al., 2022; Ziegler et al., 2005) as a baseline, but remove the rating values. To reflect on the observations above, we form a set of scenarios around the relationship between popularity, rating and user influence to assign a synthesized rating to each interaction. This approach allows us to simulate a real-world scenario where consumption is long-tail, while still experimenting with data properties relevant for popularity bias. We recognize that the scenarios are not necessarily realistic. User tendencies are likely to be more subtle in real world situations. However, we believe that experimenting with extreme behaviors can

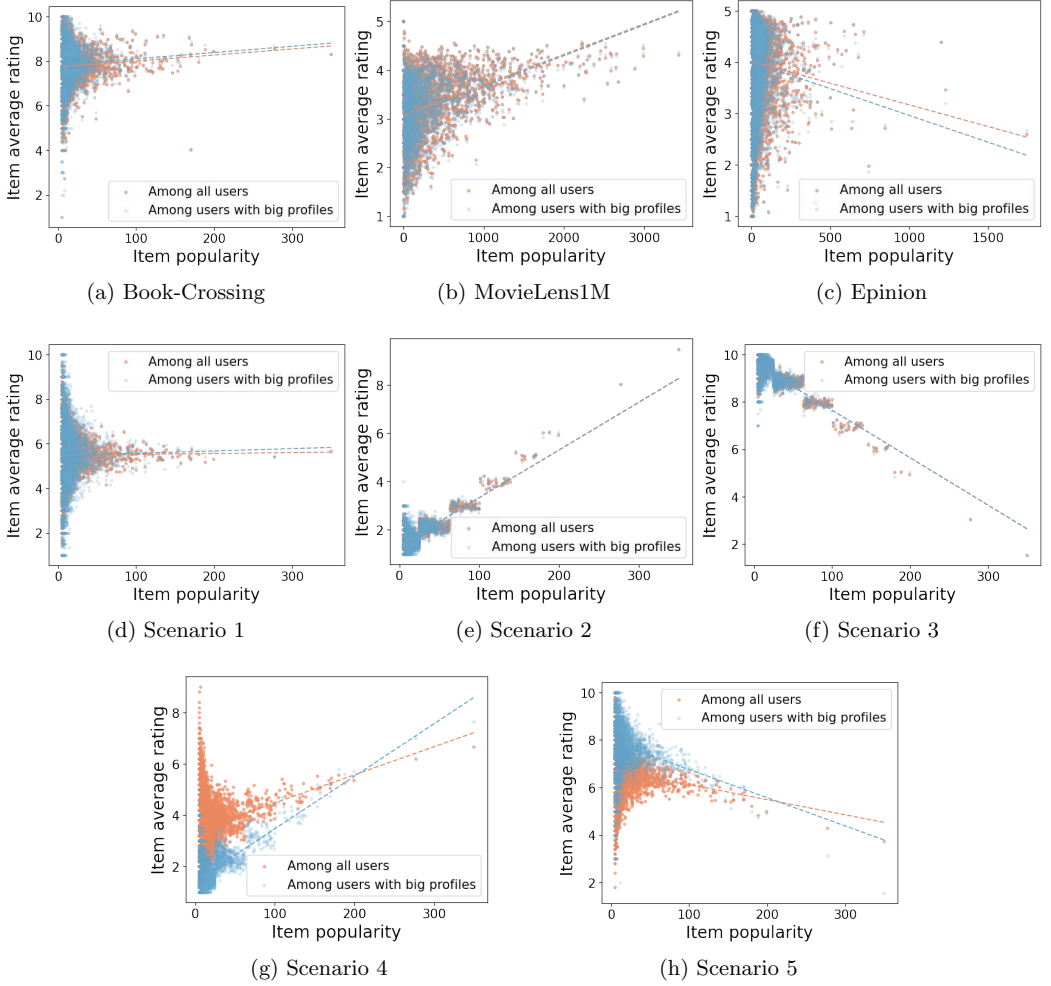


Figure 1: Relation between item average rating and item popularity among all users and among users with the 20% largest profiles, given three real and five synthetic datasets.

help us showcase the effect that we are investigating, and lead the way for more nuanced experimentation.

The scenarios, as well as the process we followed to generate each of them are as follows:

1. **Scenario 1: There is no relation between popularity and rating:** For each interaction, draw a rating value between 1 and 10 uniformly at random. In this case, the popularity of the item is not taken into account when a rating is generated.
2. **Scenario 2: Popular items are generally rated higher by the users:** For each interaction, draw a rating value between 1 and 10 from a normal distribution, where the mean is the popularity of the item normalized between 1 and 10. Since the mean of the rating distribution is the item’s normalized popularity, more popular items tend to receive higher ratings.
3. **Scenario 3: Popular items are generally rated lower by the users:** For each interaction, draw a rating value between 1 and 10 from a normal distribution, where the mean is the opposite number of the popularity of the item normalized between 1 and 10. Since the mean of the rating distribution is the opposite number of the item’s normalized popularity, more popular items tend to receive lower ratings.
4. **Scenario 4: Only users with large profiles rate popular items higher:** For each interaction, draw a rating value between 1 and 10 uniformly at random. For the users with the 20% largest profiles, replace by drawing from a Poisson distribution where the mean is the popularity of the item normalized between 1 and 10. While most users rate at random, users with large profiles tend to rate popular items higher.
5. **Scenario 5: Only users with large profiles rate popular items lower:** For each interaction, draw a rating value between 1 and 10 uniformly at random. For the users with the 20% largest profiles, replace by drawing from a Poisson distribution where the mean is the opposite of the popularity of the item normalized between 1 and 10. While most users rate at random, users with large profiles tend to rate popular items lower.

Figures 1d to 1h show the correlation between item average rating and item popularity within the five synthetic datasets. The effects are much more pronounced than for the real datasets. Scenario 1 shows no relation between average rating and popularity, as the ratings were drawn uniformly at random. Scenarios 2 and 3 showcase a very positive and a very negative correlation, respectively. For scenario 4, we see a positive correlation, which is higher for users with large profiles, and for scenario 5 a negative correlation, which is even lower for users with large profiles.

3.2 Algorithm Configurations

In this section, we describe the algorithm configurations examined for UserKNN and two matrix factorization algorithms, one traditional and one neural network based.

3.2.1 USERKNN

Despite UserKNN’s simplicity, there are configuration choices that can potentially greatly influence the result. We identified the following: minimum similarity, the items considered for similarity, and minimum neighbours.

Minimum Similarity It is common in UserKNN implementations that not all users who rated an item are considered (Desrosiers and Karypis, 2010). Instead, the notion of neighbourhood is introduced; only the users most similar to the target user are taken into account when producing a predicted score. The filtering can be done by introducing a cut off value of minimum similarity for consideration, among other techniques.

Items for Similarity Given a similarity metric (e.g., cosine similarity), a design choice still has to be made on whether similarity between two users will be calculated for their full rating vectors, or only for ratings on items these two users have in common. See (Aggarwal, 2016) for clarification.

Minimum Neighbours When the neighbourhood is constructed for a given user, then a score is predicted for each item in a list of candidates. In some implementations, the predicted score is not calculated for all potential items. Instead, the algorithm focuses on items that have been rated by at least a minimum number of neighbours of the current user.

It is worth noting that LensKit and Cornac differ when it comes to these choices as seen in Table 2. Two of the parameters tested are not configurable in Cornac, while the third is not configurable in either framework and is set to a different value in each of them.

Table 2: Configuration choices related to UserKNN made by LensKit for Python and Cornac.

Configuration choice	LensKit for Python	Cornac
Minimum similarity	Configurable (default: 0)	Fixed to -1
Items for similarity	Fixed to all items	Fixed to common items
Minimum neighbours	Configurable (default: 1)	Fixed to 1

3.2.2 TRADITIONAL MATRIX FACTORIZATION

In order to experimentally inspect the effect of implementation, we focus on a matrix factorization algorithm that is implemented by both LensKit and Cornac. Namely, we look into Biased Matrix Factorization (BMF).

BMF is a type of matrix factorization used for explicit rating prediction, and it is differently implemented in LensKit and Cornac. LensKit provides an alternating least squares implementation of BMF, and the default solver is coordinate descent (Takács et al., 2011) with weighted regularization (Zhou et al., 2008). Cornac implements a stochastic gradient descent solver for BMF (Koren et al., 2009). In both libraries, the Boolean parameter of bias is included to signify whether user and item bias are used to predict the ratings, and is by default set to True. Bias is an interesting parameter to investigate in the context of popularity bias propagation.

3.2.3 DEEP MATRIX FACTORIZATION

Along with BMF, we perform preliminary analysis on the popularity bias propagated by a neural network based matrix factorization algorithm, namely Deep Matrix Factorization (DMF). DMF uses a multi-layer perceptron to project users and items into a latent structured space (Xue et al., 2017). It takes into account explicit ratings and implicit interactions to compute the low-dimensional vectors via a neural network architecture, and then estimates the relevance of an item to a user with cosine similarity between the vectors. DMF is not available in either Cornac or LensKit. The Elliot framework (Anelli et al., 2021) includes an implementation of DMF that comes with a set of hyperparameters that can be tweaked. DMF’s hyperparameters consist of latent factors, which represent the number of units in the final MLP layer for both users and items. The regularization term controls overfitting by penalizing large weights in the model. The learning rate determines the step size during optimization. Finally, the similarity measure computes the relevance between user and item embeddings using cosine similarity.

Neural network based approaches are generally understood to be less interpretable. Therefore, predicting or even explaining the effect of their parameters on popularity bias is nontrivial. For the purposes of this study, we experiment with the number of latent factors in the final layer of the network, since they can affect the underfitting/overfitting of the model, which can have an interplay with the data characteristics.

4 Experimental Setup

In this section, we describe the experiments that we run in order to determine the effect of data characteristics and algorithm configuration on the propagation of popularity bias. We run all experiments on a Fedora Linux 40 machine with a 24-core AMD EPYC 7401 Processor and 1TiB RAM. The code we wrote to run the experiments^{2,3} has been made open source. For the Book-Crossing subset, the datasets MovieLens1M and Epinion, as well as every synthetic data scenario, we perform a recommendation process given every version of every algorithm. For UserKNN, we test the following versions given the configurations discussed in section 3.2.1:

- Min. similarity 0, over all items, 1 min. neighbour.
- Min. similarity 0, over all items, 2 min. neighbours.
- Min. similarity -1, over all items, 1 min. neighbour.
- Min. similarity -1, over all items, 2 min. neighbours.
- Min. similarity -1, over common items, 1 min. neighbour.

For BMF, we test the LensKit and Cornac version, along with the bias parameter. For DMF, we test the Elliot version, along with the number of factors in the final layer. For each algorithm version and each dataset, we perform optimization based on RMSE by splitting the dataset into 80-20% sets to find the best values for some of the non-fixed hyperparameters

2. <https://github.com/SavvinaDaniil/DiagnosingBiasRecSys>

3. <https://github.com/SavvinaDaniil/Elliot>

of the respective version, except for DMF that we based the optimization on NDCG@10 since DMF does not predict rating but relevance score. The resulting hyperparameters can be seen in our repositories. Afterwards, we divide the users into training and test users in a 5-fold cross validated way. We make sure to use the same splits for all algorithms and all versions. For every test user, we use 80% of their ratings for training and the remaining 20% for testing, which is an option in LensKit. We train the model on the training set. For each user in the test set, we predict a rating for every item they have not rated in the training set (see TrainItems in section 3.1.3 of Said and Bellogín (2014)), rank the items based on the predicted score and recommend the top-10 items, in line with recent studies on popularity bias.

We report on RMSE and NDCG@10 where applicable to estimate the effectiveness of the rating prediction and ranking, respectively. We also calculate the following widely used metrics on the recommended lists to estimate popularity bias propagation:

1. **Popularity Correlation (PopCorr)**: The correlation between popularity in training set and recommendation frequency for every item (Kowald and Lacic, 2022).
2. **Average Recommendation Popularity (ARP)**: The average popularity of the items in the recommended lists (Yin et al., 2012; Abdollahpouri et al., 2019a).
3. **Popularity Lift (PL)**: The average relative difference in popularity between the recommended items and the items in the users’ profiles (Abdollahpouri et al., 2020).

Finally, for every dataset and algorithm we perform a Mann–Whitney U test to observe whether there is a significant difference among configurations for ARP and PL, and include the result in the respective tables.

5 Results

In this section, we provide insights into how algorithm configurations impact popularity bias and performance for the different datasets by presenting the results across the set of metrics listed in section 4. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$. We abbreviate Book-Crossing as B-C and MovieLens1M as ML1M.

5.1 Popularity bias by UserKNN

5.1.1 REAL DATA

Table 3 shows the results when combining different UserKNN configuration choices with the Book-Crossing subset, MovieLens1M, and Epinion.

The extent to which popularity bias propagates in the recommendation varies across the datasets. For Book-Crossing, the algorithm performance is best when popularity bias is high, based on both RMSE and NDCG@10. When minimum neighbours are set to 1, there is no notable popularity bias according to all metrics. On the other hand, when minimum neighbours are set to 2, the PopCorr and PL metrics indicate strong popularity

Table 3: Popularity bias and performance of different UserKNN configurations given Book-Crossing, MovieLens1M, and Epinion. OverCommon set to True corresponds to the Cornac implementation, and set to False to the LensKit implementation. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$.

Dataset	Min Sim	Items for Similarity	Min Nbrs	Pop Corr↑	ARP↑	PL↑	RMSE↓	NDCG @10↑
B-C	-1	Common	1	0.010	0.002*	-32.843*	1.739	0.001
			1	-0.000	0.002*	-38.583*	1.860	0.001
			2	0.282	0.003*	15.018*	1.758	0.003
	0		1	0.071	0.003*	-17.740*	1.816	0.001
			2	0.446	0.005	67.356	1.704	0.006
ML1M	-1	Common	1	-0.093	0.000*	-99.722*	0.910	0.000
			1	-0.082	0.000*	-99.782*	0.906	0.000
			2	-0.053	0.017*	-86.270*	0.904	0.004
			10	0.247	0.175	26.134	0.894	0.055
	0		1	-0.050	0.010*	-91.918*	0.900	0.003
			2	-0.003	0.041*	-67.794*	0.894	0.013
			10	0.176	0.170*	21.997	0.898	0.048
Epinion	-1	Common	1	0.020	0.000*	20.789*	1.154	0.000
			1	0.023	0.000*	36.538*	1.212	0.000
			2	0.173	0.001*	176.224	1.168	0.000
	0		1	0.043	0.001*	65.527*	1.148	0.000
			2	0.153	0.001	165.929	1.108	0.001

bias. Additionally, minimum similarity is impactful: increasing it from -1 to 0 increases popularity bias across all metrics. Finally, which items to consider for similarity has a small effect on all metrics, but without a clear direction.

For MovieLens1M, according to NDCG@10, ranking performance is higher than for Book-Crossing and Epinion. For MovieLens1M, the minimum neighbours hyperparameter of UserKNN impacts popularity bias largely, with bias increasing when more minimum neighbours are required. In contrast to Book-Crossing, popularity bias is the highest for minimum similarity of -1, based on all metrics. Which items to consider for similarity has no discernible effect on popularity bias.

On Epinion, popularity bias is present for all versions of the algorithms according to the PopCorr and PL metrics. Again, increasing minimum neighbours increases popularity bias across metrics. The parameters of minimum similarity and which items to consider for similarity affect popularity bias, but not largely.

It is immediately observable that popularity bias varies across the different configurations of UserKNN with the parameter of minimum neighbours, one that is not configurable in one of the frameworks, being especially influential. This is true for every dataset, though the degree of popularity bias also differs between the datasets. The results indicate that popularity bias is not unavoidable when the data follows a long tail distribution, and depends on other characteristics of the datasets as well as the configuration of the algorithms.

5.1.2 SYNTHETIC DATA

To investigate which data characteristics could impact popularity bias propagation given each version of UserKNN, we present the results for the synthetic datasets in table 4.

Performance varies across the data scenarios. RMSE specifically is lower for scenarios 2 and 3 compared to the other three. In these two scenarios, users tend to agree between them on whether they like popular items or not, which facilitates the rating prediction task. NDCG@10 is the highest for scenario 2, where popular items are highly rated by the users. In this case, the rating prediction and ranking tasks are linked, since the highest ranked (i.e., popular items) are also highly rated.

Popularity bias also varies across the data scenarios, and the effect depends on the algorithm configuration. In the following paragraphs, we describe and reflect on the most impactful effects of the interaction between data and configuration.

For scenario 1 where ratings are uniformly at random generated, there is no notable popularity bias propagation observed when minimum neighbours are set to 1, while there is bias when minimum neighbours is set to 2. This observation is in line with what we noted for the real datasets, where increasing minimum neighbours results in higher popularity bias for all datasets and metrics.

In scenario 3 where all users agree that popular items are bad, popularity bias is not propagated when minimum similarity is set to 0. However, when setting minimum similarity to -1, we can observe popularity bias propagation across all metrics. The reason is that users with completely different opinions are considered and their opinions count negatively. Therefore, popular items still get recommended since everyone’s “negative” neighbours dislike them, and we can observe popularity bias propagation across all metrics.

Table 4: Popularity bias and performance of different UserKNN configurations given synthetic data based on different data scenarios. OverCommon set to True corresponds to the Cornac implementation, and set to False to the LensKit implementation. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$.

Data Scenario	Min Sim	Items for Similarity	Min Nbrs	Pop Corr↑	ARP↑	PL↑	RMSE↓	NDCG @10↑
Scenario 1	-1	Common	1	0.004	0.002*	-35.746*	3.337	0.001
			1	0.018	0.002*	-32.285*	3.502	0.001
			2	0.418	0.004*	21.252*	3.352	0.003
	0	All	1	0.101	0.003*	-12.827*	3.624	0.002
			2	0.615	0.005	65.440	3.464	0.005
Scenario 2	-1	Common	1	0.604	0.015*	305.197*	1.150	0.013
			1	0.596	0.021*	426.621*	1.188	0.019
			2	0.614	0.022*	447.618*	1.190	0.021
	0	All	1	0.552	0.027	632.300	1.040	0.023
			2	0.562	0.027	591.966	1.026	0.025
Scenario 3	-1	Common	1	0.522	0.006*	151.686*	1.151	0.001
			1	0.559	0.008*	187.197*	1.182	0.002
			2	0.728	0.008	192.127	1.182	0.002
	0	All	1	0.025	0.002*	-35.765*	1.044	0.001
			2	0.161	0.003*	-13.100*	1.034	0.004
Scenario 4	-1	Common	1	0.184	0.003*	8.669*	2.458	0.001
			1	0.253	0.003*	23.063*	2.502	0.001
			2	0.772	0.006*	97.490*	2.404	0.004
	0	All	1	0.588	0.008*	164.549*	2.500	0.004
			2	0.701	0.014	297.047	2.386	0.010
Scenario 5	-1	Common	1	0.057	0.002*	-16.243*	2.783	0.001
			1	0.087	0.003*	-7.924*	2.880	0.001
			2	0.623	0.005*	57.969	2.776	0.003
	0	All	1	0.136	0.003*	-16.122*	2.914	0.003
			2	0.612	0.005	42.849	2.794	0.006

When considering only common items to calculate similarity, users with smaller profiles have a larger influence. This is relevant in scenario 4 where users with large profiles like popular items. Table 4 shows that even though scenario 4 still leads to popularity bias, considering only common items reduces it across all metrics. Therefore, this implementation choice can have a big impact on whether popularity bias is propagated and to what extent.

Finally, the value for minimum neighbours largely influences popularity bias. Across almost all scenarios and metrics, increasing minimum neighbours from 1 to 2 leads to increased popularity bias. By setting a higher neighbour barrier for considering an item for recommendation, it follows that less popular items will be disadvantaged. This result is particularly relevant given that the parameter of minimum neighbours could only be tweaked in one of the considered frameworks, so studies that use Cornac or LensKit might reach different conclusions on the extent of popularity bias propagated by UserKNN.

Popularity bias manifests differently in different datasets; an explanation for this can be found in data characteristics, specifically the relation between item ratings, item popularity, and the influence of users with large profiles. All three configuration choices affect the observed bias, as they influence the weight of each user’s preference.

5.2 Popularity bias by Matrix Factorization algorithms

5.2.1 REAL DATA

Table 5 shows the results for BMF, trained on Book-Crossing, MovieLens1M and Epinion.

Table 5: Popularity bias and performance on the Book-Crossing, MovieLens1M and Epinion datasets given different BMF implementations. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$.

Dataset	Framework	Bias parameter	Pop Corr↑	ARP↑	PL↑	RMSE↓	NDCG @10↑
B-C	Cornac	False	-0.015	0.001*	-68.639*	1.587	0.001
		True	0.003	0.002*	-40.168*	1.535	0.001
	Lenskit	False	-0.013	0.002*	-58.518*	1.762	0.004
		True	0.108	0.005	46.533	1.560	0.004
ML1M	Cornac	False	0.266	0.193	35.186	0.856	0.064
		True	0.234	0.192	35.859	0.856	0.059
	Lenskit	False	0.151	0.125*	-14.392*	0.860	0.038
		True	0.183	0.155*	7.987*	0.866	0.040
Epinion	Cornac	False	0.001	0.000*	-53.845*	1.152	0.000
		True	0.009	0.001*	19.814*	1.029	0.000
	Lenskit	False	0.020	0.001*	-54.107*	1.240	0.000
		True	0.126	0.003	402.394	1.030	0.001

BMF tends to have a better performance than UserKNN overall. Popularity bias varies across the LensKit and Cornac implementations of BMF. Also very notable is the effect of the bias parameter.

For Book-Crossing, in both implementations the bias parameter increases popularity bias. For example, PopCorr on Book-Crossing with LensKit is -0.013 with bias set to False and 0.108 with bias set to True, as seen in Table 5. Specifically, the LensKit implementation with the bias parameter set to True is the only version where there is a positive popularity correlation, as well as positive popularity lift, with the difference being significant.

For MovieLens1M, there is higher popularity bias across metrics when the Cornac implementation is used. Additionally, the bias parameter changes PL from negative to positive when the LensKit implementation is used.

For Epinion, there is higher popularity bias across metrics when the LensKit implementation is used. The bias parameter is also very influential, as setting it to True results in higher popularity bias across all metrics, given both implementations, as was the case for Book-Crossing.

We see that the choice for a framework - LensKit or Cornac - largely determines whether popularity bias is observed when using a Matrix Factorization algorithm on widely used datasets, and the effect differs per dataset. In addition, setting the bias parameter of these algorithms impacts to what extent popularity bias is observed.

5.2.2 SYNTHETIC DATA

Additionally to the conclusions drawn from the results on the real datasets, to observe whether the data scenarios influence popularity bias propagated by BMF and the impact of the bias parameter, we present the results on the synthetic datasets in table 6.

BMF has a better performance than UserKNN when trained on the synthetic datasets. Similarly to UserKNN, BMF also results in low RMSE for scenarios 2 and 3, and high NDCG@10 for scenario 2. Experimenting with different synthetic datasets helps explore the observation from the previous section that BMF propagates popularity in some cases. Popularity bias can be observed for scenarios 2 and 4, while scenario 5 where users with large profiles do not like popular items results in no popularity bias across metrics. This indicates that the preferences of users with large profiles are influential for the system, and thus, for popularity bias.

In the previous section, we saw that the LensKit and Cornac implementations lead to varying results with respect to popularity bias. On the synthetic datasets, the effect is more apparent. The bias parameter has opposite effects between the two implementations in scenario 4 as seen in Table 6. In Cornac, using bias increases popularity bias across all metrics. On the contrary, the bias parameter decreases popularity bias in the LensKit implementation. This can be due to the use of different optimization methods by Cornac and LensKit. Further investigation is needed to better understand the impact of the bias parameter on popularity bias.

We conclude that, even though the data synthesis was based on the functionality of UserKNN, the differences between the synthesized datasets largely impact popularity bias propagated by BMF. Furthermore, the results confirm that for both BMF implementations,

Table 6: Popularity bias and performance given different BMF implementations and synthetic data based on different data scenarios. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$.

DataScenario	Framework	Bias parameter	Pop Corr↑	ARP↑	PL↑	RMSE↓	NDCG @10↑
Scenario 1	Cornac	False	-0.013	0.001*	-67.527*	3.191	0.001
		True	0.079	0.006	74.548	2.872	0.003
	Lenskit	False	-0.015	0.002*	-52.395*	3.400	0.002
		True	-0.008	0.002*	-49.725*	3.028	0.001
Scenario 2	Cornac	False	0.475	0.030	743.388	0.938	0.026
		True	0.467	0.030	743.446	0.802	0.025
	Lenskit	False	0.507	0.030*	740.064	0.940	0.025
		True	0.498	0.030*	742.366	0.818	0.025
Scenario 3	Cornac	False	-0.022	0.001*	-76.210*	0.850	0.001
		True	0.010	0.002	-31.473	0.796	0.001
	Lenskit	False	-0.091	0.001*	-73.688*	1.032	0.001
		True	0.017	0.002*	-37.387*	0.810	0.001
Scenario 4	Cornac	False	0.036	0.004*	-20.286*	2.510	0.005
		True	0.424	0.027	647.074	2.297	0.019
	Lenskit	False	0.462	0.010*	149.807*	2.746	0.012
		True	0.219	0.008*	129.271*	2.390	0.008
Scenario 5	Cornac	False	-0.018	0.001*	-73.474*	2.671	0.001
		True	-0.006	0.002*	-53.320*	2.500	0.001
	Lenskit	False	-0.045	0.001*	-59.393*	2.890	0.002
		True	-0.013	0.002	-50.113	2.582	0.001

the bias parameter affects popularity bias. The effect is different - sometimes even opposite - for different combinations of dataset and implementation.

5.3 Popularity bias by Deep Matrix Factorization

The goal of this section is to see whether the lessons learned from the detailed analysis presented in the previous sections hold for a neural network-based method.

5.3.1 REAL DATA

Table 7 shows the results when combining different DMF configuration choices with the Book-Crossing subset and MovieLens1M. Training DMF on Epinion was not possible due to memory allocation limitations.

The number of latent factors in the final layer of the neural network has a clear impact on the results. Interestingly, the impact differs between the two datasets. Specifically, when the number of factors is set to 64, popularity bias increases for Book-Crossing, with the difference being significant. In the case of MovieLens1M, the same configuration results in significantly lower values across metrics.

We observe that the combination of data and configuration also affects popularity bias propagation by neural architectures.

Table 7: Popularity bias and performance on the Book-Crossing, MovieLens1M and Epinion datasets given different DMF versions. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann-Whitney U test with $p < 0.005$.

Dataset	Factors	Pop Corr↑	ARP↑	PL↑	NDCG @10↑
B-C	32	0.012	0.002*	-38.610*	0.002
	64	0.125	0.005	35.582	0.003
ML1M	32	0.046	0.064	-54.777	0.010
	64	0.032	0.057*	-59.654*	0.008

5.3.2 SYNTHETIC DATA

To confirm and expand upon the observations of the previous section, we report on performance and popularity bias of DMF on the synthetic data in table 8.

Popularity bias and performance fluctuate among scenarios. In line with the results from previous sections, popularity bias is the highest for scenarios 2 and 4, and the low for the other scenarios. The parameter tested also has an effect, which aligns with the effect on the real data. For all scenarios, increasing the number of factors in the final layer increases popularity bias, in most cases significantly, which aligns with the results on Book-Crossing.

Note that the results presented in this section are preliminary. Further research is needed to explore the effect of data characteristics and algorithm configuration on popularity

Table 8: Popularity bias and performance given different DMF versions and synthetic data based on different data scenarios. For each metric, we embolden the highest value among configurations. For ARP and PL, we use the asterisk (*) to signify which values are significantly lower than the highest one according to a Mann–Whitney U test with $p < 0.005$.

DataScenario	Factors	Pop Corr↑	ARP↑	PL↑	NDCG @10↑
Scenario 1	32	0.057	0.003	-8.529	0.003
	64	0.059	0.003	-10.247	0.003
Scenario 2	32	0.183	0.005	42.030	0.004
	64	0.720	0.004*	-13.542	0.017
Scenario 3	32	0.039	0.003*	-19.041*	0.002
	64	0.074	0.004	4.613	0.003
Scenario 4	32	0.009	0.002*	-41.003*	0.002
	64	0.393	0.003	-34.005	0.006
Scenario 5	32	0.004	0.002*	-43.433*	0.002
	64	0.079	0.003	-10.451	0.003

bias propagated by neural network based algorithms. A future study could align with recent advancements in recommender systems by formulating a ranking prediction task and studying popularity bias propagation by the neural network based algorithms available in commonly used open source frameworks.

The preliminary results indicate that the conclusions drawn above also hold for neural network based approaches: data characteristics determine whether popularity bias occurs; popularity bias is not unavoidable even when the data follows a long tail distribution; parameters that can be set at implementation time have a large effect on the propagation of popularity bias.

6 Discussion

6.1 Implications of the present study

Our research shows that multiple data and configuration factors can have an effect on whether bias is propagated. Relying on frameworks readily available to researchers is convenient and a concrete step towards reproducibility, but requires being aware and detailed about the limitations. When simple parameters, such as minimum neighbours in UserKNN, are so influential, it raises questions on how generalizable research on recommender systems bias can be. Our results indicate that bias studies can only draw conclusions within the limits of their specific research and not further than that.

It follows that being explicit about the context within which a type of bias is studied is crucial, both in terms of data characteristics and implementation. It is a known issue in recommender systems literature that implementation details are often not disclosed by studies. Even in cases where they are, guessing the effect of different hyperparameters that are not present in an implementation or experimented with is not trivial. Bias reporting is

definitely not complete if it is not accompanied by clarity around the characteristics, goals and limitations of the system that is being studied.

6.2 Recommendations of the present study

Based on the results of the present study, we put forward two recommendations towards researchers who study bias in recommender systems.

First, researchers should analyze and report on the dataset characteristics that might impact the type of bias they are concerned with. For UserKNN, the relationship between rating and popularity, as well as the preferences of users with large profiles impact popularity bias and should be taken into account in relevant studies. For other algorithms and types of bias, there could be other relevant characteristics. Such analysis will help the reader understand the extent to which the results are a result of the dataset characteristics.

Second, researchers should test multiple algorithm configurations when measuring bias propagation. In a similar way that the community expects x-fold cross validation, since presenting the results of only one run may not be reliable, we could expect to see results on multiple algorithm configurations as well. If the conclusions are only valid for one specific configuration of the algorithm at hand, then that should be clear in the limitations of the study.

6.3 Limitations of the present study and future work

Despite our extensive testing, the results are potentially sensitive to our own experimental design, such as the method for train-test splitting or randomness in the data generation process. Similarly, instantiating the different implementations with the exact same configuration choices is not always possible due to some of the parameters not being configurable. As a result, there might be implementation differences between the frameworks that we are not aware of and cause part of the variation in results, irrespectively of the configurations tested. On this note, we noticed that our reported accuracy does not always coincide with the conclusions of other papers that study the same algorithms and/or datasets. For example, the developers of DMF reported high NDCG@10 results for MovieLens1M (Xue et al., 2017). However, their evaluation strategy was very different from ours: for every user, they only held out one item consumed by them for testing, and only ranked 100 random items instead of all the items not present in the training set like we did. It is reasonable that their performance is better than the one we report. This discrepancy further supports our claim that generalizing conclusions beyond the boundaries of a specific study is not always possible and should be done carefully, which aligns with our previous work where we emphasized the impact of evaluation strategy (Daniil et al., 2024). These observations highlight the importance of our line of research instead of hindering it, since they hint that data and implementation dependence might be present more often than we think.

We recognize that the scholar community has generally moved on from explicit user preferences and rating prediction. We do not focus on implicit feedback given that recent studies on popularity bias are often performed on datasets with explicit ratings in the context of rating prediction tasks (Naghiaei et al., 2022; Kowald et al., 2020; Kowald and Lacic, 2022). A process similar to ours can be followed in the context of implicit feedback: instead of tweaking ratings, researchers can form scenarios around the distribution of popularity in

the dataset and study distributions with varying levels of long-tail. Future work can also focus on components that we chose not to investigate in this study, such as more advanced algorithms, other open libraries for implementation, and the vulnerability of other types of bias to data and algorithms. For example, researchers can create data scenarios based on assumptions regarding the relationship between an item’s rating and the demographic characteristics of its creator. Further nuance can be introduced in the data synthesis part, by allowing for more complex relationships between popularity, rating and user influence. Future work could further investigate the impact of data characteristics and configurations on the complex relationship between system performance and popularity bias. Our findings could also be of use in the domain of bias mitigation: certain algorithm configurations appear to have a clear effect on popularity bias (e.g., increasing the minimum neighbours in UserKNN consistently increases it). It is, therefore, relevant to investigate whether appropriately configuring certain parameters assists with popularity bias mitigation.

7 Conclusion

In this study, we reflected on the need for fundamental understanding of the relationship between data, algorithms and bias in recommender systems. We focused on reporting on popularity bias, and tracked algorithm configurations and data characteristics that are of importance in its propagation. Accordingly, we generated a set of synthetic datasets, experimented with performing a recommendation process on real and synthetic datasets using different configurations of the algorithms at hand, and evaluated popularity bias using well-known metrics. We found that even when the distribution of popularity in the dataset is long-tail, popularity bias is not unavoidable. We showed that the relationship between popularity and rating, as well as the preferences of users with large profiles have an impact on bias. We highlighted the sensitivity of bias propagation to algorithm configuration and, by extension, framework implementation. Our observations point to methodology and reproducibility issues that extend further than a specific use case, to the recommender systems field at large.

Recommender systems are widely used in our online lives, and bias propagation by such systems can have serious societal impact. With this work, we hope to have called attention to the ambiguity in bias reporting and motivated researchers to strive for reproducibility and highlight specificity when appropriate.

Acknowledgments and Disclosure of Funding

This research was funded through a public-private partnership between CWI and the KB National Library of the Netherlands.

References

Himan Abdollahpouri. *Popularity bias in recommendation: a multi-stakeholder perspective*. PhD thesis, University of Colorado at Boulder, 2020.

- Himan Abdollahpouri and Masoud Mansoury. Multi-sided exposure bias in recommendation, 2020. URL <https://arxiv.org/abs/2006.15772>.
- Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the 11th ACM conference on recommender systems*, pages 42–46, 2017.
- Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking, 2019a. URL <https://arxiv.org/abs/1901.07555>.
- Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The unfairness of popularity bias in recommendation. In *Recommendation in Multi-stakeholder Environments (RMSE), in conjunction with the 13th ACM Conference on Recommender Systems*, 2019b.
- Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The connection between popularity bias, calibration, and fairness in recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 726–731, 2020.
- Charu C Aggarwal. Neighborhood-based collaborative filtering. *Recommender Systems: The Textbook*, pages 29–70, 2016.
- Abdul Basit Ahanger, Syed Wajid Aalam, Muzafar Rasool Bhat, and Assif Assad. Popularity bias in recommender systems-a review. In *International Conference on Emerging Technologies in Computer Engineering*, pages 431–444. Springer, 2022.
- Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2405–2414, 2021.
- Alejandro Bellogín and Alan Said. Improving accountability in recommender systems research through reproducibility. *User Modeling and User-Adapted Interaction*, 31(5):941–977, 2021.
- Alejandro Bellogín, Pablo Castells, and Iván Cantador. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal*, 20:606–634, 2017.
- Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.
- Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. From niches to riches: Anatomy of the long tail. *Sloan management review*, 47(4):67–71, 2006.
- Òscar Celma and Pedro Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8, 2008.

- Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Transaction on Information System*, 41(3), 2023. ISSN 1046-8188.
- Paolo Cremonesi and Dietmar Jannach. Progress in recommender systems research: Crisis? what crisis? *AI Magazine*, 42(3):43–54, 2021.
- Savvina Daniil, Mirjam Cuper, Cynthia CS Liem, Jacco van Ossenbruggen, and Laura Hollink. Reproducing popularity bias in recommendation: The effect of evaluation strategies. *ACM Transactions on Recommender Systems*, 2(1):1–39, 2024.
- Yashar Deldjoo, Alejandro Bellogin, and Tommaso Di Noia. Explaining recommender systems fairness and accuracy through the lens of data characteristics. *Information Processing & Management*, 58(5):102662, 2021.
- Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144, 2010.
- Bora Edizel, Francesco Bonchi, Sara Hajian, André Panisson, and Tamir Tassa. Fairec-sys: Mitigating algorithmic bias in recommender systems. *International Journal of Data Science and Analytics*, 9(2):197–213, 2019.
- Michael D Ekstrand. Lenskit for python: Next-generation software for recommender systems experiments. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2999–3006, 2020.
- Mehdi Elahi, Danial Khosh Kholgh, Mohammad Sina Kiarostami, Soroush Saghari, Shiva Parsa Rad, and Marko Tkalčič. Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management*, 58(5):102655, 2021.
- Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM conference on recommender systems*, pages 101–109, 2019.
- Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–49, 2021.
- F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25:427–491, 2015.
- Shah Khushro, Zafar Ali, and Irfan Ullah. Recommender systems: issues, challenges, and research opportunities. In *Information science and applications*, pages 1179–1189. Springer, 2016.

- Anastasiia Klimashevskaja, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction*, pages 1–58, 2024.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- Dominik Kowald and Emanuel Lacić. Popularity bias in collaborative filtering-based multimedia recommender systems. In *International Workshop on Algorithmic Bias in Search and Recommendation*, pages 1–11. Springer, 2022.
- Dominik Kowald, Markus Schedl, and Elisabeth Lex. The unfairness of popularity bias in music recommendation: A reproducibility study. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR, Lisbon, Portugal, Proceedings, Part II 42*, pages 35–42. Springer, 2020.
- Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, 2007.
- Mohammadmehdi Naghiaei, Hossein A Rahmani, and Mahdi Dehghan. The unfairness of popularity bias in book recommendation. In *Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, Revised Selected Papers*, pages 69–81. Springer, 2022.
- Alan Said and Alejandro Bellogín. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 129–136, 2014.
- Aghiles Salah, Quoc-Tuan Truong, and Hady W Lauw. Cornac: A comparative framework for multimodal recommender systems. *The Journal of Machine Learning Research*, 21(1):3803–3807, 2020.
- Manel Slokom. Comparing recommender systems using synthetic data. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 548–552, 2018.
- Manel Slokom and Martha Larson. Doing data right: How lessons learned working with conventional data should inform the future of synthetic data for recommender systems. *arXiv preprint arXiv:2110.03275*, 2021.
- Gábor Takács, István Pilászy, and Domonkos Tikk. Applications of the conjugate gradient method for implicit feedback collaborative filtering. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 297–300, 2011.
- Karen HL Tso and Lars Schmidt-Thieme. Empirical analysis of attribute-aware recommender system algorithms using synthetic data. *Journal of Computers*, 1(4):18–29, 2006.
- Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43, 2023.

- Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *IJCAI*, volume 17, pages 3203–3209. Melbourne, Australia, 2017.
- Emre Yalcin. Blockbuster: A new perspective on popularity-bias in recommender systems. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 107–112. IEEE, 2021.
- Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the 38th International Conference on Very Large Data Bases (VLDB Endowment)*, 5(9):896–907, 2012.
- Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In Rudolf Fleischer and Jinhui Xu, editors, *Algorithmic Aspects in Information and Management*, pages 337–348, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-68880-8.
- Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005.

Don't Use LLMs to Make Relevance Judgments

Ian Soboroff

IAN.SOBOROFF@NIST.GOV

*National Institute of Standards and Technology
Gaithersburg, Maryland, USA*

Editor: Vanessa Murdock, Djoerd Hiemstra

Abstract

Relevance judgments and other truth data for information retrieval (IR) evaluations are created manually. There is a strong temptation to use large language models (LLMs) as proxies for human judges. However, letting the LLM write your truth data handicaps the evaluation by setting that LLM as a ceiling on performance. There are ways to use LLMs in the relevance assessment process, but just generating relevance judgments with a prompt isn't one of them.¹

Keywords: Information Retrieval Evaluation, Large Language Models

1 Introduction

The Text Retrieval Conference (TREC) is a community evaluation and dataset construction activity sponsored by the U.S. National Institute of Standards and Technology (NIST). TREC has run annually since 1991. TREC is divided into *tracks* which embody specific search tasks. The canonical TREC task is *adhoc search*, with searches against a static set of documents, each search returning a single ranked list of documents. An individual search instance is called a *topic* and expresses the user's information need in long form rather than providing a query. The relevance judgments, or *qrels*, maps each topic to the documents that should be retrieved for it. The combination of the document collection, the topics, and the relevance judgments is called a *test collection*.

The relevance judgments representing ground truth are created collaboratively between participants and NIST using a process called *pooling* (Jones and van Rijsbergen, 1975). TREC participants use their IR systems to return the top K documents for each topic. The union of the top-ranked $k \ll K$ documents from each participant system is called the *pool*. The documents in the pool are reviewed by the person who invented the topic, and they decide which documents are relevant and which are not. Using the qrels as labels we can compute various measures of retrieval effectiveness such as precision and recall. The test collections let researchers rapidly innovate new search algorithms in a laboratory setting before deploying them to a live system. More information about TREC can be found in Voorhees and Harman (2005), a book covering the first ten years of the program.

The process used in TREC descends from the Cranfield indexing experiments conducted by Cyril Cleverdon in the 1960s, and so we say TREC is following the Cranfield paradigm (Cleverdon, 1967). Central to the Cranfield paradigm is a set of assumptions that simplify the search problem: the document collection and information needs are fixed,

1. This article reflects the views of the author and not necessarily those of NIST or of the U. S. Government.

all documents are labeled as relevant or not relevant to every query, relevance is modeled by topical similarity, the relevance of a document is independent of the relevance of any other document, there is a single query that is answered with a single ranked list, and the relevance judgments are representative of the user population (Voorhees, 2001). TREC can be thought of as a community effort in pushing the bounds of the Cranfield paradigm. Complete judgments were replaced with the pooling procedure, which has been shown to be sufficient for measuring the pooled systems and also useful for measuring systems which were not pooled for evaluation, as long as certain properties are maintained (Harman, 1995; Zobel, 1998; Buckley et al., 2007). Likewise, many TREC tracks push back on the notion of static documents and information needs (Frank et al., 2014; Carterette et al., 2014), relevance as topical similarity (Craswell and Hawking, 2004; Balog et al., 2011), single rankings (Owoicho et al., 2022; Aliannejadi et al., 2023), and independent relevance (Soboroff and Harman, 2005).

Making the relevance judgments for a TREC-style test collection can be complex and expensive. Relevance assessing at NIST for a typical TREC track usually involves a team of six contractors working for 2-4 weeks. Those contractors need to be trained and monitored. Software has to be written to support recording relevance judgments correctly and efficiently. Experience in both the technical and human aspects of the process counts for a lot, which is why we run evaluation campaigns rather than everyone building their own test collections. Evaluation campaigns are infrastructure for IR research.

The recent advent of large language models that produce astoundingly human-like flowing text output in response to a natural language prompt has inspired IR researchers to wonder how those models might be used in the relevance judgment collection process (Bauer et al., 2023; Faggioli et al., 2024).

At the ACM SIGIR 2024 conference, a workshop “LLM4Eval” provided a venue for this work, and featured a data challenge activity where participants reproduced TREC deep learning track judgments, as was done by Thomas et al. (2024). I was asked to give a keynote at the workshop, and this paper presents that keynote in article form.

The bottom-line-up-front message is, don’t use LLMs to create relevance judgments for TREC-style evaluations.

2 Automatic evaluation

The idea of automatic evaluation for information retrieval came from a paper I wrote with Charles Nicholas and Patrick Cahan in 2001 (Soboroff et al., 2001). I had been reading Ellen Voorhees well-known SIGIR paper from 1998 (Voorhees, 1998) which shows experimentally that while people differ in their judgments of relevance, those differences don’t affect the relative ordering of systems in a TREC evaluation. Surprised by this result, I wondered what would happen if the relevance judgments were randomly sampled from the pools. Certainly, TREC relevance judgments aren’t random, but how much can they vary towards random and still rank systems equivalent to the official system ranking?

A representative example result from that work is shown in Figure 1. The single +’s are official scores from TREC, and the ×’s with whiskers up and down are scores obtained with random documents from the pool labeled as relevant. The points are ordered according to their official TREC scores. The key point to notice is that, using random judgments, the

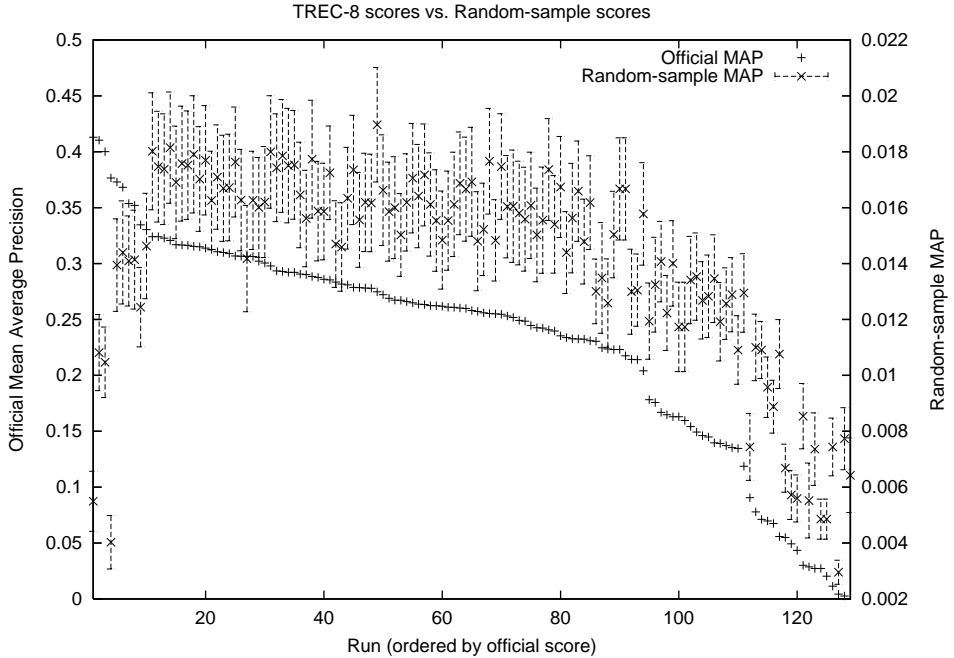


Figure 1: Sample result from Soboroff et al. (2001), TREC-8, TREC-style pooling to depth 100.

best systems (those with the highest MAP, on the left) look like the worst systems (on the right).

Automatic evaluation in this sense means making relevance judgments using an algorithm rather than people, as opposed to inducing relevance from implicit behavior cues or history. In published papers proposing automatic evaluation methods for IR, the quality of these methods is quantified by comparing the ordering of systems induced by the automatic method to the official TREC ranking based on manual assessments. That is, the qrels are used to score each system using some metric, and the systems are then ordered by their score. The common metric used is Kendall’s tau (τ), a correlation measure between rankings. Some researchers also use the more familiar Spearman’s rho (ρ), a correlation measure that takes into account the distance between the points and not just their rank order. Others have proposed versions of these correlations that emphasize the upper part of the ranking (i.e. the best systems) (Yilmaz et al., 2008). I have heard a discussion of a variation of tau where a swap in position between two systems only happens if they are

significantly different according to some statistical test, which would take advantage of the fact that the points in the ranking represent average performance over a set of topics.

Aslam and Savell (2003) published a short paper that explained my 2001 results. No matter how many systems retrieve a given document, it is only added to the pool once. There are many more nonrelevant than relevant documents, so a given relevant document was likely to have been retrieved by more than one system. (The mean number of systems retrieving a nonrelevant document in TREC-8 is 3, versus 11 for a relevant document.) By selecting the relevant documents at random, I was implicitly selecting documents retrieved by many systems. So my 2001 paper shows the results of a popularity contest. Under this approach, the worst systems and the best systems both look bad, because they fail (or succeed) by retrieving documents that other systems don't find. Another way to think about Aslam and Savell is that by using the output of a system as the ground truth, I am measuring the similarity of the two systems, how close the retrieval system is to the model that created the ground truth.

Around that time, the BLEU automatic metric for machine translation (Papineni et al., 2002) and the ROUGE automatic metric for summarization (Lin, 2004) were published. These measures compare system-generated outputs, such as translations and summaries, by the overlap of word n -grams with a model or reference output. ROUGE worked well for extractive summarization, where a summary is produced by cutting and pasting sentences from the source documents, but less well in a generative setting where word choice could vary quite a bit from the original documents.

3 Machine learning and predicting from examples

A more successful method of imputing relevance comes in the form of relevance feedback and more sophisticated machine learning algorithms. In these cases, examples of relevant (and possibly nonrelevant) documents are used to train a model to predict the relevance of other documents.

Relevance feedback (RF) in the vector space model, developed by Joseph Rocchio and Eleanor Ide in the mid to late 1960s as part of the SMART system (Salton and McGill, 1983), may be the first on-line machine learning algorithm.² In RF, the user executes an initial search and identifies one or more documents in the search results as being relevant or irrelevant. The terms in the query are augmented and reweighted based on the feedback, and the refined query is executed to rank the remaining documents in the collection. Since then, it has been adopted as a general IR technique rather than a specific algorithm and has been instantiated within nearly every ranking model. Currently, the most common implementation of RF uses the BM25 ranking algorithm with the RM3 method of term weighting (Jaleel et al., 2004). Pseudo-relevance feedback (PRF) is a modification in which instead of indicating relevance by the user, some number of documents ranked highest in the initial ranking are assumed to be relevant for a round of relevance feedback (Buckley et al., 1994). Relevance feedback is one of the most successful techniques in information retrieval, producing large improvements in performance. PRF is somewhat similar except that for some topics it fails because the initial retrieval is off base for some reason.

2. Rocchio did not describe his method as machine learning, but he did develop a theory that his relevance feedback method builds an optimal query.

Büttcher et al. (2007) proposed using the TREC relevance judgments to predict the relevance of unjudged documents retrieved by unpooled systems, and also as a method for expanding the set of relevance judgments overall. They use the qrels to train a binary classifier and then apply that to documents that were not judged but were ranked above the pool depth of TREC’s pools. Anecdotally, this technique did not perform as well when all retrieved documents (down to rank 1000) were predicted, so there is something to restricting predictions to those documents that are already ranked highly by the search ranker.

Rajput et al. (2012) describe an iterative method using *nuggets*. A nugget here is a manually selected passage from a relevant document. Starting with manual nuggets, the process identifies new high-probability shingles as new nuggets and uses those to predict the relevance of other documents. Their use of the term “nugget” is different than how the term is used for evaluation of summarization and question answering; summarization nuggets are atomic pieces of information which must be manually aligned to the generated summary, whereas these nuggets, being strings or shingles, are automatically matched. Nuggets in (Rajput et al., 2012) are essentially lexical patterns that identify relevant documents.

The BERTScore metric (Zhang et al., 2020) computes token similarity between a generated and reference output using BERT embeddings. We can think of this as the LLM equivalent of BLEU, using embeddings instead of n-grams.

4 LLM-based predictions of relevance

Modern large language models (LLMs) have inspired a new approach, where a topic and document are embedded in a prompt, which is then fed to an LLM that outputs some indicator of relevance. LLMs may be fine-tuned with relevance examples, or other relevant documents may be included in the prompt, but otherwise no examples are used as in the supervised learning methods above.

Thomas et al. (2024) describe using LLMs to predict relevance in TREC collections as well as for search results from a major commercial search engine. They develop prompts at a number of levels of richness. In their web search results, they find that the generated judgments “have proved more accurate than any third-party labeler, including staff; they are much faster end-to-end than any human judge, including crowd workers; they scale to much better throughput; and of course are many times cheaper.” The paper describes results on TREC data in greater detail, and there is an extended discussion of their prompts and their evolution.

MacAvaney and Soldaini (2023) used nearest-neighbors, classifiers, and LLM prompts to elicit relevance judgments to supplement judgments in MS-MARCO (Nguyen et al., 2016), a collection where there is only a single manually-adjudicated relevance judgment per query. By basing a system’s performance measurement on more than one document, IR metrics are found to be more stable.

Alaofi et al. (2024) investigated the agreement between LLMs and TREC assessors and found that LLM false positive decisions seemed to be related to the presence of query terms in the passage being assessed. A false positive is where the LLM votes that the passage is relevant, but the human assessor judged it to not be relevant. In many of these cases, the false positive passages included terms from the query, despite not being relevant. This

seems to imply that despite the richness of the language model, lexical cues can influence the decision more than the true meaning of the text.

Outside of the information retrieval domain, researchers seem to be eagerly jumping on a bandwagon for LLM-based automatic evaluation. As one example, Lin and Chen (2023) employ prompts to gauge generated responses in open-domain dialogues, and compare results to other automatic evaluation techniques, some of which use the LLM to identify properties of good responses (Mehri and Eskenazi, 2020b,a) and others which use the LLM to directly assess dialog responses (Chen et al., 2023; Fu et al., 2024). None of the comparison metrics is validated against manual labels of the dialogues in question.

The search for automatic metrics is long and has made use of new algorithms as they have been developed. There is a real need for automatic metrics, because manual assessment is slow and hard to scale. When the labels are created zero-shot, specifically meaning that the evaluation model is operating at the same degree of data exposure as the systems being measured, the evaluation reduces to comparing the performance of the system to the model, not to human performance. When the evaluation model has more knowledge than the systems being measured, for example relevance judgments on the topics in the test set, then the model may produce an evaluation that can stand as a useful measurement, a comparison to something more than just another system. When the evaluation model is making use of outside knowledge, for example in Mehri and Eskenazi (2020a), then the situation depends on the systems being measured. The following sections elaborate this argument.

5 Retrieval and evaluation are the same problem

Asking a computer to decide if a document is relevant is no different than using a computer to retrieve documents and rank them in order of predicted degree of relevance. In both cases, the algorithm makes the assessment of relevance.

A retrieval system, or a relevance model, is a model of relevance given available data. The system is trying to predict which documents are relevant and which documents are not. Even though real systems might try to optimize a pairwise or listwise output or compute a degree of relevance of a document or a search engine result page, it is useful to think of all these processes as predicting relevance.

During relevance assessment, we are asking the assessor to decide whether documents are relevant or not. This, too, is essentially a prediction of relevance. It’s a well-informed prediction since the person is reading the document and often composed the information need, but since the task is artificial, the assessor is basically saying that they would include this document in a report on the topic, a report which they don’t ever actually write. We can call that a prediction too.

We use one set of predictions, the relevance judgments, to measure the performance of the other set of predictions, the system outputs. In doing so, we declare the relevance judgments to be truth. In fact, you can switch the two sets of predictions, declare the system output to be truth, and measure the “effectiveness” of the assessor compared to that of the system. All evaluations which compare a system output to an answer key are making a measurement with respect to the answer key, not with respect to the universe.

Since both retrieval systems and relevance assessors are making predictions of relevance, evaluation and retrieval are the same problem. We can imagine a very slow system that would have a human read every document and assess its relevance given a query.

John Searle’s “Chinese Room” thought experiment³ posits a person in a box who receives questions through a slot and delivers answers out the slot. The questions and answers are in Chinese, a language which the person does not read or speak. Rather, the person follows a sophisticated set of instructions for generating an answer from a question, in Chinese, by manipulating symbols on the paper. Thus, the box appears to understand and communicate in human language but is basically a computer. A mechanical Chinese Room can be implemented with an LLM chatbot. Construct a prompt of a topic statement and a document and ask for the LLM to say relevant or not, for every document in the collection. Asking the language model about relevance is the mirror of evaluation.

If we believed that a model was a good assessor of relevance, then we would just use it as the system. Why would we do otherwise? We don’t use human assessors that way, because it doesn’t scale. LLMs in 2024 don’t scale, but that feels like an engineering problem more than something fundamental; we will probably solve this with better hardware and smaller models.

Since both retrieval and evaluation are prediction activities it seems natural to apply machine learning to both. The predictions don’t happen in isolation: systems know about collection frequencies and click patterns that inform the ranked list, and assessors have experience and world knowledge that informs their labels. Machine learning, the field where we train prediction systems by example, clearly has a role to play here.

As with any prediction, there are errors of omission and commission (or false negatives and false positives if you prefer), and those errors represent a maximum discriminative ability of those relevance judgments to distinguish systems. I will dive into this in more detail in the next section.

6 The ceiling on performance

Whatever we use as the answer key represents both an ideal solution and a ceiling on measurable performance. No system can outperform the evaluation’s answer key. When the answer key is made up of human-assigned labels, then we are saying that human performance is the ideal we are aiming for, and we can’t measure something better than that performance. Likewise, when the answer key is created by a machine learning model or some other mechanical process, we are saying that the model represents the idea we are aiming for, and we can’t measure something better than the performance of that model. This is the critical flaw with LLM-sourced relevance judgments.

An IR test collection is a 3-tuple:

$$C = \{D, S, R\}$$

where D is a set of documents, S is a set of search needs, and R is a function $R : S \mapsto D$ that maps search needs to relevant documents. In the original Cranfield collections, there is a value of R for every document d and search need s . In the TREC collections, most of

3. https://en.wikipedia.org/wiki/Chinese_room

those pairs are unknown and pooling lets us assume that an unjudged document is likely not relevant.

A retrieval model produces a ranking of documents $d_n : d_n \in D$ in order of predicted relevance to the search need s :

$$A(s, D) = \{d_n \mid d_n \in D\}$$

where the set here is an ordered set, a sequence of documents where each document appears once. The entire document collection is ranked although in practice we cut off the ranking much earlier.

An evaluation function $E(A, R)$ computes a real number from a k -prefix of the ranked list A^k . Often in TREC $k = 1000$ but some measures set k much lower to focus on the top of the list. If the number of relevant documents for $sR_s \geq k$, then a system can produce a ranking whose prefix consists only of relevant documents. Thus in practice we try to have search needs with many fewer relevant documents, and a fundamental difficulty with enormous collections like ClueWeb is that we can easily find thousands of relevant documents and still worry that we have not found them all (Buckley et al., 2007).

The relevance judgments in a Cranfield experiment are a model of human behavior, and since we are trying to build systems that understand information needs and documents as well as humans do, they model ideal retrieval performance. The evaluation function E is typically defined to be maximized by an ideal ranking, for example if all relevant documents are ranked ahead of any irrelevant documents. If you take the relevance judgments and turn it into a run by first listing all the relevant documents and then padding the listing to k with irrelevant documents, it gets perfect scores on the appropriate metrics.

Historically, this was the goal of IR performance. IR systems are meant to augment people by scaling up their ability to understand information, and so the performance of people is the ideal.

This ideal is also a limit on what Cranfield can measure. Under R , the best possible ranking

$$A(s, D) \mapsto \{+, +, +, +, \dots, -, -, -\}$$

orders the relevant documents ahead of any irrelevant documents. The order of relevant documents among themselves, and irrelevant documents among themselves, are not important: there is a very large number of equivalently ideal rankings by permutations among the relevant and irrelevant documents. For graded relevance regimes, this ranking orders documents by their rated degree of relevance, where those degrees are positive integers, zero for not relevant, and perhaps negative numbers for other poor outcomes like spam, and within each relevance degree or category the documents can be permuted to create equally ideal ranked lists. If two or more categories are equivalently valuable, we can replace them with a superset including all equivalently valuable documents. Without loss of generality, moving forward I will assume that rankings can have all the relevant documents ahead of all the irrelevant documents for any relevance construct.

Theorem 1 (ideal rankings) *Let C be a test collection (D, S, R) where $R : s \mapsto d$ maps search needs to relevant documents $\{+, +, +, \dots\}$. Let $A(s, D)$ be a ranking function that produces a ranking of documents $\{d_n \in D\}$ for a search need s . Let $E : A(s, D), R \mapsto \mathbb{R}$ be*

an evaluation metric that computes a real number representing the quality of the ranking A given the relevance judgments R . Then, we can define the **ideal ranking** as

$$A(s, D) \mapsto \{+, +, +, +, \dots, -, -, -\}$$

the ranking that places the relevant documents ahead of any irrelevant documents. The ideal ranking maximizes E , and there does not exist any ranking A' that obtains a higher value than A of E subject to R .

Proof Suppose a ranking A' that has one or more relevant documents that are not in ideal ranking A . For A' to be ideal, these extra relevant documents must appear at the head of the ranking. However, the ideal is defined subject to R , the full set of relevance judgments, and A is already defined to be ideal. If there are extra relevant documents missing from A , then A is not ideal. If the new “relevant” documents are not in R , then A' can't be ideal either. So A and A' must have the same set of relevant documents in their ranking, and $E(A, R)$ and $E(A', R)$ are equal and maximize E . ■

If we imagine we have a system that is better than a human, for example by finding relevant documents that are not in the relevance judgments or correctly ranking a document which was assessed incorrectly, that system will score less than perfectly when we score it using the human relevance judgments. This must be the case, because unjudged documents are assumed to be not relevant, and the documents found by this novel system are either absent from the relevance judgments or judged non-relevant when they should have been marked relevant. The system has retrieved documents which **according to the evaluation relevance judgments** are not relevant. And so this top-performing system is measured as performing less well than it does.

This is reflected in my 2001 paper, and with other papers that came later, but exemplified by Figure 1 above. The (true) top systems are under-ranked by the “model” of relevance. This must be true for any model of relevance that generates relevance judgments, be it human or machine. We cannot measure a system that is better than the relevance judgments. Or, rather, the evaluation can't distinguish such a system from one that performs less than perfectly.

As a counterexample, Büttcher et al. (2007) trained a model using an incomplete set of manual judgments to classify a larger set of documents automatically, improving the collection. In this case the evaluation model is privileged in comparison to the runs, in that it has relevance information that they do not. Relevance feedback nearly always improves performance, so we would expect a hybrid set of judgments like these to have the possibility of outperforming evaluation using the shallow judgments. This is outperforming the original human but only doing so by retrospective use of human relevance data.⁴

And so when the relevance judgments are created by a person, the model can't exceed the human ideal. If we had a model that had “super-human” performance, we would just make our IR system use that model. In the current state of the art, the most advanced

4. We still have a grounding problem, in that you may not believe that the model makes accurate predictions. The process can be improved by doing a second stage of relevance assessments on the classifier outputs in order to estimate the classifier error rates.

LLMs are used as components of systems that may be hypothetically measured by relevance judgments generated from the same models. Those systems cannot perform better than the model generating the relevance judgments.

Obviously, the human that created the relevance judgments is not entirely ideal. The assessor is not all-knowing, all-seeing, all-reading with perfect clarity. Assessors make mistakes, and TREC participants are fond of finding them. More importantly, the assessor is only one person; someone else with the same information need would make different judgments. If we compared the assessor’s judgments to those of a secondary assessor by pretending that secondary assessor is a run, it would necessarily perform less than perfectly.

That means that even if we imagine that systems exist which perform better at the task than humans do, we can’t see that improved performance in a Cranfield-style evaluation. This follows from the ideal ranking theorem above. It must also be true for **any evaluation** where we are comparing a system output to a “gold standard,” for example in machine learning or natural language processing, because the gold standard represents ideal performance, and by the ideal ranking theorem, no ranking can be measured as better than the truth data.

The so-called “super-human” performance observed on benchmark datasets⁵ is actually just measurement error. Super-human performance would be scored as less than ideal by the established ground truth, because performing better than a human entails making different decisions than those in the ground truth.

Some benchmarks are capable of showing super-human performance by differentiating between the humans performing the task and the humans that create the answer key. For example, an LLM may perform better than many people on a standardized test, but we can measure that because the humans taking the test are not the source of the correct answers. Likewise tests of solving analogies or complex math problems. In IR evaluations, we are only comparing to the answer key, not another person’s attempt to recreate the answer key.⁶ To summarize, you should not create relevance judgments using a large language model, because:

- You are declaring the model to represent ideal performance, and so you can’t measure anything that might perform better than that model.
- The model used to create relevance judgments is certainly also used as part of the systems being measured. Those systems will evaluate as performing poorly even if they actually improve on the model, because improving on the model means retrieving new relevant unjudged documents that aren’t in the answer key.
- When the next shiny model comes out, it will measure as performing less well than the old model, because it necessarily must retrieve unjudged documents or ones judged incorrectly as not relevant. And so the relevance judgments can only measure systems

5. For example, <https://openai.com/index/planning-for-agi-and-beyond/> and <https://venturebeat.com/ai/google-deepmind-unveils-superhuman-ai-system-that-excels-in-fact-checking-saving-costs-and-improving-accuracy/> For a contrasting view, see Tedeschi et al. (2023).

6. We wouldn’t do that because assessor disagreement is reality. There are no absolutely correct answers outside the Cranfield room.

that perform worse than the state of the art at the time the relevance judgments were created.⁷

7 Limitations

The argument in this paper makes the case that using an LLM to generate the ground truth for an IR evaluation results in a substandard evaluation. However, it could certainly be the case that LLMs could play different roles in the evaluation process than inventing the answer key.

LLMs can be used to create the answer key if they have more knowledge about relevance than the systems being measured. If the evaluation model has access to privileged information, for example by being fine-tuned on manual relevance judgments on the evaluation topics, then those relevance judgments should still be able to measure systems that use the untuned model. While it might be tempting to assume that the LLM has information about relevance in the training data, we should avoid this assumption since we don't have access to the training data.

Blessing the evaluation model with extra relevance information is what makes Büttcher et al's (Büttcher et al., 2007) results work: the model is trained on relevance data, and so the trained model has the advantage over any system it is measuring that doesn't have access to that relevance data.

We actually already knew this: the fact that relevance feedback improves retrieval is a basic result in IR. If we have relevance information gleaned from many systems as we do when pooling, the outputs will perform better than any individual system and thus we can measure any individual systems with less information about relevance than the collective pool.

This still has the problem that the evaluation isn't future-proof: we might have a new model that outperforms the relevance feedback of the prior generation. We haven't seen this yet when the collections are pooled from older systems only, if the collection is well-judged (Voorhees et al., 2022). Or new models might have TREC triples (topic, document, relevance) as an explicit component in their training data and be able to make use of that in a retrieval setting.

One simple idea that seems promising is to employ a LLM to follow the assessor and look out for mistakes. This can't be done by simply asking, "Did you mean 'relevant' instead?" since people are primed to trust the computer more than they should (Logg et al., 2019; Bogert et al., 2021).⁸ But it may be possible to automate a quality-control process using a model.

There are evaluation activities that don't involve creating an answer key. For example, in a user study, researchers observe user behavior and analyze those observations to draw conclusions about the experiment. LLMs might be useful in supporting the observational

7. This might be a good thing. Since new models will look worse than old models, when their developers run them through leaderboard-style benchmarks they will cast them aside because they seem to perform poorly. Then we won't ever have to worry about having a new model.

8. Logg et al. (2019) is also interesting as much behavioral literature four to five years prior seemed to find that people distrusted algorithmic recommendations. Perhaps our perspectives are changing with exposure. But see also Sparrow et al. (2011) on search's effects on memory.

process (perhaps by transcribing mouse movements and clicks in a readable way) or the analysis process (much as we use statistical models to determine significance).

At the SIGIR workshop, a questioner asked if an LLM-generated evaluation might still be useful even given its flaws. For example, Thomas et al. (2024) found LLM judgments to be as useful as crowdsourced judgments, but not better than curated judgments from a trained team. In their setup, crowd judgments represented a low rung in a tiered hierarchy of relevance judgments and system measurements. If the judgments are not meant to support a rigorous evaluation but rather as noisy training data, then the LLM judgments may be useful. But if the LLM creating the truth data is part of the search system, or is of an older generation than the search system, the results may under-report performance and not be able to distinguish improvements, as shown above. In all cases it should be kept in mind that the ideal used as a comparison point is not human performance, but model performance.

8 Conclusions

I have discussed the limitations of using models to create relevance judgments. You don't want to do that, because then you have limited what you can measure to the level of the generating model. If the generating model is also part of the evaluated systems, you are stuck in a loop, or perhaps falling into a bottomless pit.

This is similar to model collapse (Guo et al., 2024). When you train the model using its own outputs, the performance of the model decreases. The collapse mechanism is the measurement error that comes from generating the truth data using the evaluated system. In this case, evaluation is a loss function computed based on the generated truth data.

This doesn't mean that LLMs can't allow us to do amazing things. As someone who got his start in IR working with LSI (Deerwester et al., 1990), which is essentially an optimal linear embedding, I am very excited by the idea of nonlinear embeddings. IR systems that use LLMs to surmount the vocabulary boundary have enormous promise for real users.

All models have limits, and humans do too. If we want to use the model to evaluate performance, we first need to consider if we are doing something past the ability of the model as used in that evaluation paradigm. The relevance judgments barrier is a fundamental limitation of evaluations that measure systems against ground truth.

Acknowledgments and Disclosure of Funding

I thank Ellen Voorhees and Rikiya Takehi for their comments on the talk and on early versions of this paper. I also thank the attendees of the SIGIR 2024 LLM4Eval workshop for their insightful questions and continuing discussions. No external funding was received in support of this work. The TREC activity has been annually reviewed by NIST's Research Protection Office and determined to not be human subjects research. Any company, product or service mentioned in this paper should not be taken as an endorsement of that company, product, or service by NIST. Nothing in this paper should be read as a comparison to or among commercial products.

References

- Marwah Alaofi, Paul Thomas, Falk Scholer, and Mark Sanderson. LLMs can be fooled into labelling a document as relevant: best café near me; this paper is perfectly relevant. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP)*, page 32–41, 2024. Association for Computing Machinery. ISBN 9798400707247. doi: 10.1145/3673791.3698431. URL <https://doi.org/10.1145/3673791.3698431>.
- Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. TREC IKAT 2023: The interactive knowledge assistance track overview. In Ian Soboroff and Angela Ellis, editors, *The 32nd Text REtrieval Conference Proceedings (TREC)*, volume 1328 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2023. URL https://trec.nist.gov/pubs/trec32/papers/Overview_ikat.pdf.
- Javed A. Aslam and Robert Savell. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 361–362, 2003. Association for Computing Machinery. ISBN 1581136463. doi: 10.1145/860435.860501. URL <https://doi.org/10.1145/860435.860501>.
- Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. Overview of the TREC 2011 entity track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Twentieth Text REtrieval Conference, (TREC)*, volume 500-296 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2011. doi: 10.6028/NIST.SP.500-296.entity-overview. URL <http://trec.nist.gov/pubs/trec20/papers/ENTITY.OVERVIEW.pdf>.
- Christine Bauer, Ben Carterette, Nicola Ferro, Norbert Fuhr, Joeran Beel, Timo Breuer, Charles L. A. Clarke, Anita Crescenzi, Gianluca Demartini, Giorgio Maria Di Nunzio, Laura Dietz, Guglielmo Faggioli, Bruce Ferwerda, Maik Fröbe, Matthias Hagen, Allan Hanbury, Claudia Hauff, Dietmar Jannach, Noriko Kando, Evangelos Kanoulas, Bart P. Knijnenburg, Udo Kruschwitz, Meijie Li, Maria Maistro, Lien Michiels, Andrea Papenmeier, Martin Potthast, Paolo Rosso, Alan Said, Philipp Schaer, Christin Seifert, Damiano Spina, Benno Stein, Nava Tintarev, Julián Urbano, Henning Wachsmuth, Martijn C. Willemsen, and Justin Zobel. Report on the Dagstuhl seminar on frontiers of information access experimentation for research and education. *SIGIR Forum*, 57(1), 2023. ISSN 0163-5840. doi: 10.1145/3636341.3636351. URL <https://doi.org/10.1145/3636341.3636351>.
- Eric Bogert, Aaron Schechter, and Richard T. Watson. Humans rely more on algorithms than social influence as a task becomes more difficult. *Scientific Reports*, 11(8028), 2021.
- Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART: TREC 3. In *Proceedings of The Third Text REtrieval Conference, (TREC)*, volume 500-225 of *NIST Special Publication*, pages 69–80. National Institute of Standards and Technology (NIST), 1994.

- Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. Bias and the limits of pooling for large collections. *Information Retrieval*, 10(6):491–508, 2007. ISSN 1386-4564. doi: 10.1007/s10791-007-9032-x. URL <https://doi.org/10.1007/s10791-007-9032-x>.
- Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 63–70, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277755. URL <https://doi.org/10.1145/1277741.1277755>.
- Ben Carterette, Evangelos Kanoulas, Mark M. Hall, and Paul D. Clough. Overview of the TREC 2014 session track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Third Text REtrieval Conference, (TREC)*, volume 500-308 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2014. doi: 10.6028/NIST.SP.500-308.session-overview. URL <http://trec.nist.gov/pubs/trec23/papers/overview-session.pdf>.
- Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-ijcnlp.32. URL <https://aclanthology.org/2023.findings-ijcnlp.32/>.
- C. W. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19: 173–192, 1967. (Reprinted in *Readings in Information Retrieval*, K. Sparck Jones and P. Willett, eds, Morgan Kaufmann, 1997).
- Nick Craswell and David Hawking. Overview of the TREC 2004 web track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenth Text REtrieval Conference, (TREC)*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2004. URL <http://trec.nist.gov/pubs/trec13/papers/WEB.OVERVIEW.pdf>.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41(6):391–407, 1990. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASIT3.0.CO;2-9.
- Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. Who determines what is relevant? Humans or AI? Why not both? *Communications of the ACM*, 67(4):31–34, 2024. ISSN 0001-0782. doi: 10.1145/3624730. URL <https://doi.org/10.1145/3624730>.

- John R. Frank, Max Kleiman-Weiner, Daniel A. Roberts, Ellen M. Voorhees, and Ian Soboroff. Evaluating stream filtering for entity profile updates in TREC 2012, 2013, and 2014. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Third Text REtrieval Conference, (TREC)*, volume 500-308 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2014. doi: 10.6028/NIST.SP.500-308.kba-overview. URL <http://trec.nist.gov/pubs/trec23/papers/overview-kba.pdf>.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.365. URL <https://aclanthology.org/2024.naacl-long.365/>.
- Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text, *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. URL <https://arxiv.org/abs/2311.09807>.
- Donna Harman. Overview of the fourth Text Retrieval Conference (TREC-4). In *TREC*, volume 500-236 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1995.
- Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. UMass at TREC 2004: Novelty and HARD. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenth Text REtrieval Conference (TREC)*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2004. URL <http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf>.
- K. Sparck Jones and C. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- Yen-Ting Lin and Yun-Nung Chen. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In Yun-Nung Chen and Abhinav Rastogi, editors, *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI)*, pages 47–58, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlp4convai-1.5. URL <https://aclanthology.org/2023.nlp4convai-1.5/>.
- Jennifer M. Logg, Julia A. Minson, and Don A. Moore. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision*

Processes, 151:90–103, 2019. ISSN 0749-5978. doi: <https://doi.org/10.1016/j.obhdp.2018.12.005>. URL <https://www.sciencedirect.com/science/article/pii/S0749597818303388>.

Sean MacAvaney and Luca Soldaini. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2230–2235, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3592032. URL <https://doi.org/10.1145/3539618.3592032>.

Shikib Mehri and Maxine Eskenazi. Unsupervised evaluation of interactive dialog with DialoGPT. In Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes, editors, *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.sigdial-1.28. URL <https://aclanthology.org/2020.sigdial-1.28/>.

Shikib Mehri and Maxine Eskenazi. USR: An unsupervised and reference free evaluation metric for dialog generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 681–707, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.64. URL <https://aclanthology.org/2020.acl-main.64/>.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computation (CoCo@NIPS)*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R. Trippas, and Svitlana Vakulenko. TREC CAsT 2022: Going beyond user ask and system retrieve with initiative and response generation. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the 31st Text REtrieval Conference, (TREC) volume 500-338 of NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022. doi: 10.6028/NIST.SP.500-338.cast-overview. URL https://trec.nist.gov/pubs/trec31/papers/Overview_cast.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.

Shahzad Rajput, Matthew Ekstrand-Abueg, Virgil Pavlu, and Javed A. Aslam. Constructing test collections by inferring document relevance via extracted relevant information. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, page 145–154, 2012. Association for Com-

puting Machinery. ISBN 9781450311564. doi: 10.1145/2396761.2396783. URL <https://doi.org/10.1145/2396761.2396783>.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., USA, 1983. ISBN 0070544840. URL <https://sigir.org/resources/museum/>.

Ian Soboroff and Donna Harman. Novelty detection: The TREC experience. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 105–112, 2005. Association for Computational Linguistics. URL <https://aclanthology.org/H05-1014/>.

Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 66–73, 2001. Association for Computing Machinery. ISBN 1581133316. doi: 10.1145/383952.383961. URL <https://doi.org/10.1145/383952.383961>.

Betsy Sparrow, Jenny Liu, and Daniel M. Wegner. Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333(6043):776–778, 2011. doi: 10.1126/science.1207745. URL <https://www.science.org/doi/abs/10.1126/science.1207745>.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. What’s the meaning of superhuman performance in today’s NLU? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.697. URL <https://aclanthology.org/2023.acl-long.697>.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1930–1940, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657707. URL <https://doi.org/10.1145/3626772.3657707>.

Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 315–323, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291017. URL <https://doi.org/10.1145/290941.291017>.

Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems (CLEF)*, page 355–370, 2001. Springer-Verlag. ISBN 3540440429. URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151546.

- Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.
- Ellen M. Voorhees, Ian Soboroff, and Jimmy Lin. Can old TREC collections reliably evaluate modern neural retrieval models? *CoRR*, abs/2201.11086, 2022. URL <https://arxiv.org/abs/2201.11086>.
- Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 587–594, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390435. URL <https://doi.org/10.1145/1390334.1390435>.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=SkeHuCVFDr>.
- Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 307–314, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291014. URL <https://doi.org/10.1145/290941.291014>.

Search and Society: Reimagining Information Access for Radical Futures

Bhaskar Mitra
Microsoft Research
Montréal, Canada

BMITRA@MICROSOFT.COM

Editor: Monica Lestari Paramita

Abstract

Information retrieval (IR) research must understand and contend with the social implications of the technology it produces. Instead of adopting a reactionary strategy of trying to mitigate potential social harms from emerging technologies, the community should aim to proactively set the research agenda for the kinds of systems we *should* build inspired by diverse explicitly stated sociotechnical imaginaries. The sociotechnical imaginaries that underpin the design and development of information access technologies needs to be explicitly articulated, and we need to develop theories of change in context of these diverse perspectives. Our guiding future imaginaries must be informed by other academic fields, such as human-computer interaction, information sciences, media studies, design, science and technology studies, social sciences, humanities, democratic theory, and critical theory, as well as legal and policy experts, civil rights and social justice activists, and artists, among others. In this perspective paper, we motivate why the community must consider this radical shift in how we do research and what we work on, and sketch a path forward towards this transformation.

Keywords: IR and society, Sociotechnical imaginaries, Theory of change, Technology and power

1 Introduction

Information retrieval (IR) research must understand and contend with the social implications of the technology it produces. Nearly half a century ago, Belkin and Robertson (1976) concluded that IR research should acknowledge its responsibility to society and “must become both theoretically self-conscious and self-consciously based upon a social ideology”. This perspective has gained traction in the IR community in recent years. Researchers in attendance at the third Strategic Workshop in Information Retrieval in Lorne (SWIRL) (Culpepper et al., 2018) identified fairness, accountability, confidentiality, and transparency in IR (“FACT IR”) as socially consequential and strategically important research directions for the field. The following year, the FACTS-IR workshop (Olteanu et al., 2019) added “safety” as a fifth pillar. Subsequently, a large body of recent IR literature has grappled with questions of fairness, transparency, and explainability in the context of information access.

However, it is our perspective that this growing focus on fairness and ethics in IR—despite having played a critical role in bringing much-needed attention to the societal implications of IR systems and advancing the conversation about the IR community’s responsibility to broader society—operates within a severely constrained frame that leaves the many

underlying values, politics, and socioeconomic incentives that guide IR research largely unchallenged. For example, faced with the applications of generative artificial intelligence (AI) for information access the IR community has focused on concerns of fair ranking and representation and limiting model “hallucinations”¹ for good reasons, but have largely ignored other significant consequences of these technologies on society, such as for the information ecosystem and how these systems concentrate power and control (Mitra et al., 2024). In machine learning (ML), there has been similar recent perspectives (Blodgett et al., 2020; Miceli et al., 2022) that, for example, calls for shifting the lens from fairness and bias to the power differentials that exists between those who build technology, those who use it, and those who are subject to it. Others in the ML community have brought attention to the questions of how these technologies shift power (Kalluri et al., 2020) and simultaneously constrain ethics interventions in practice (Widder et al., 2023) and shape our collective sociotechnical futures. Even by accepting the frame that we should develop fairness and transparency mechanisms for certain systems, we may inadvertently ignore the alternative perspective that some of these technologies should be dismantled, not made fairer, nor more transparent (Barocas et al., 2020; Wilkinson et al., 2023; Merchant, 2023). Consequently, at the recent fourth SWIRL workshop², researchers in attendance called for expanding the “FACTS-IR” framing to center IR research on societal, democratic, and emancipatory values. Similar sentiments, including re-centering IR on societal needs and informing IR research with democratic theories, were also discussed at the first Search Futures workshop (Azzopardi et al., 2024b).

What do we propose? In this paper, we argue that IR research, instead of adopting a reactionary strategy of trying to mitigate potential social harms from emerging technologies by developing new fairness and transparency interventions, should aim to proactively set the research agenda for the kinds of systems we *should* build inspired by diverse explicitly stated sociotechnical imaginaries. Towards that goal, IR research needs to explicitly articulate the sociotechnical imaginaries (Jasanoff and Kim, 2009, 2015) that underpin the design and development of information access technologies, and develop theories of change (Weiss, 1995; Brest, 2010; Taplin and Clark, 2012; Wikipedia contributors, 2013). Jasanoff and Kim (2015) define *sociotechnical imaginaries* as “collectively held, institutionally stabilized, and publicly performed visions of desirable futures, animated by shared understandings of forms of social life and social order attainable through, and supportive of, advances in science and technology”. These shared visions do not only imagine but also co-produce our futures through development and government of digital technologies (Mager and Katzenbach, 2021). Diverse imaginaries promoted by different corporations, professional communities, political organizations, and social movements can coexist “in tension or in a productive dialectical relationship” (Jasanoff and Kim, 2015).

In technological research and development, these diverse, and often diverging, perspectives and visions that guide the community frequently remain implicit and unstated (Wilkinson et al., 2023) despite the significant influence they exert on what the community focuses on and produces. Because of the consequential role that access to information plays in political participation by citizens in democratic societies and social transformation (Higgins and

1. We acknowledge that the term “hallucination” anthropomorphizes AI models and its usage should be discouraged. However, given the popular usage of that term in the IR community, we make an exception here for clarity.

2. <https://sites.google.com/view/swirl2025>

Gregory, 2013; Polizzi, 2020; Goldstein, 2020; Correia, 2002; González, 2021), and as a social determinant of health (Moretti et al., 2012) and economic progress (Yu, 2002; Mutula, 2008), it is even more important to critically reflect on the values and motivations that guide the design and deployment of popular information access systems. *What sociotechnical futures do IR researchers and system designers envision and how do they influence the design of current and future IR systems? Whose sociotechnical imaginaries are granted normative status and what myriad of radically alternative futures are we overlooking?* For example, what are the implications of the reliance of popular search and social media platforms on advertising as the primary source of revenue generation (Ang, 2022) and how does Big Tech’s (Oremus, 2017) increasing dominance over academic research (Whittaker, 2021) influences and/or homogenizes the kinds of IR systems we build? *What is our role, as IR researchers, to safeguard communities from falling victim to crisis of imaginations (Haiven, 2014) and how do we become more open to welcoming influences from radically new sociotechnical imaginaries?* For example, what would IR systems look like if designed for futures informed by feminist, queer, decolonial, anti-racist, anti-casteist, anti-ableist, and abolitionist thoughts, and if the focus of IR research was not to prop up colonial cisheteropatriarchal capitalist structures but to dismantle them? We believe that explicating, critiquing, and consciously choosing the values and sociotechnical imaginaries that shape IR research is critical to realizing positive social outcomes through IR research. As Benjamin (2024) argues, exercising our imagination is “an invitation to rid our mental and social structures from the tyranny of dominant imaginaries”, or as Le Guin put it:

“The exercise of imagination is dangerous to those who profit from the way things are because it has the power to show that the way things are is not permanent, not universal, not necessary.”

– Ursula K. Le Guin

The Wave in the Mind: Talks and Essays on the Writer, the Reader, and the Imagination (Le Guin, 2004)

For IR to concretely support diverse sociotechnical imaginaries the research community also needs to develop their own theories of change. Theory of change (Weiss, 1995; Brest, 2010; Taplin and Clark, 2012; Wikipedia contributors, 2013) can be defined as a participatory process whereby groups and stakeholders articulate their long-term goals and identify necessary preconditions in a planning process. Consequently, in IR adjacent fields—*e.g.*, in the Fairness, Accountability, and Transparency (FAccT)³ community—there has been similar recent calls to reimagine our sociotechnical futures (da Hora et al., 2024) and develop theories of change (Wilkinson et al., 2023) to make explicit the visions for desired futures of responsible computing and the strategic pathways that lead to those desired futures. Here, we argue that IR research similarly needs to both explicitly articulate and support diverse imagined futures and develop corresponding theories of change for how new information access technologies can take us towards these desired worlds. Theories of change in this context may benefit IR research by encouraging community members to make their goals

3. <https://facctconference.org>

and assumptions explicit, making it feasible to test stated theories, encouraging the community to work towards building consensus, and even aid in developing potential means of evaluation of desired progress (Weiss, 1995).

Maximizing social good and minimizing harm in this context should not just be the concerns of the few in our community working on fairness, transparency, ethics, and related areas, but the domain of all IR research that should be guided by theories of change towards these envisioned futures. In this context, we largely agree with the perspective of Belkin and Robertson (1976) but diverge on a critical point which is that we believe IR research should not seek any singular notions of “universal” social ideology but explicitly adopt pluralistic humanistic and emancipatory values and make space for diverse visions and perspectives.

Consequently, the task for IR researchers here is not to put themselves in positions to pick the guiding social ideologies nor push technosolutionism to address today’s social problems. Instead, our guiding future imaginaries must be co-developed with scholars from diverse fields such as human-computer interaction (HCI), information sciences, media studies, design, science and technology studies (STS), social sciences, and political sciences, as well as legal and policy experts, civil rights and social justice activists, and artist, to name a few. Not all sociotechnical imaginaries are equal in this respect, and what futures guide our research must be informed by the values and ethics of our community, which should be constantly discussed, debated, and challenged by the community as part of our sociotechnical research and be open to external critique. To summarize, as a research community we should invest our energies and resources to: (i) Nurture digital spaces where radical visions and projects for human emancipation, social progress, and equity and justice can take shape, (ii) encourage experimentation within our research community with new approaches to information access informed by new sociotechnical imaginaries cross-pollinating through interdisciplinary scholarship, and (iii) ensure that the tools and artefacts we produce as a community do not uphold systems of oppression nor contribute towards other systemic social harms.

So far, we have argued that IR research should explicate the sociotechnical futures we want to realize and develop theories of change towards these desired futures. In Section 2 we present a brief overview of existing literature on fairness and ethics in IR, and share our critical perspectives on it to motivate our work. Then, in Section 3 we draw from relevant movements in IR-adjacent fields, specifically those with explicit values and prefigurative politics; we share some ideas on how the community can get started on this journey; and who should be doing this work. In Section 4 we motivate why now is an appropriate time for the community to consider this shift. We conclude in Section 5 with some final remarks on potential pitfalls and desired outcomes. Our goal with this paper is to raise sociopolitical consciousness in the IR research community so that we *all* see our research embedded in projects of future world making and recognize our collective responsibility to affect social good; and to dismantle the artificial separation between the work on fairness and ethics in IR and the rest of IR research.

2 Background

IR systems act as intermediaries between information seekers and information artefacts. These artefacts may represent: (i) economic and other opportunities for consumers, (ii) mon-

etization opportunities for content creators and publishers, (iii) specific sociopolitical frames and ideologies, and (iv) lenses to view individuals and groups that are subjects of representation by the content. These systems infer the information need from highly incomplete expressions of interests (*e.g.*, short keyword-based search queries for web search) or implicit signals (*e.g.*, history of previously accessed artefacts in case of recommender systems), and make subjective estimates of an artefact’s relevance to the information need. Consequently, these systems are not neutral tools for lookup (Noble, 2018) and the choices these systems make exert systemic influence over what information is exposed and consumed at scale. These systems bear a responsibility to society to not only mitigate potential harms, like allocative and representational harms (Crawford, 2017), but also to maximize social good.

Representational harms may happen due to reinforcement of negative stereotypes (*e.g.*, by disproportionately suggesting arrest record searches in ads corresponding to searches for black-identifying first names (Sweeney, 2013) or suggesting racist stereotypes in query autocompletion (Noble, 2018)), by pandering to the white male gaze (*e.g.*, by sexualizing women of color in search results (Noble, 2018; Urman and Makhortykh, 2024)), and through erasure (*e.g.*, underrepresenting women and other historically marginalized peoples in image search for occupational roles (Kay et al., 2015)). Allocative harms may manifest from disparate exposure in search and recommendation results (Singh and Joachims, 2018)—*e.g.*, when women are recommended lower-paying jobs in ads (Datta et al., 2014) or by influencing traffic to websites that depend on ad-monetization. Beyond direct representational and allocative harms, these systems also hold tremendous power to shape political discourse and culture (Grimmelmann, 2008; Gillespie, 2019; Hallinan and Striphas, 2016).

In light of these, there has been multiple calls (Culpepper et al., 2018; Olteanu et al., 2019) in the IR community to study and address these potential harms. Bernard and Balog (2023) report a significant rise in publications in this area after 2016, with fairness and transparency receiving the most attention. The increasing focus on these sociotechnical aspects of information access has been at least partly in response to recent advances in foundation models (Bommasani et al., 2021) and their implications for the future of information access (Shah and Bender, 2022; Mitra et al., 2024).

Fairness in ranking has garnered so much interests that there are numerous recent surveys (Ekstrand et al., 2021; Zehlike et al., 2021, 2022a,b; Pitoura et al., 2022; Dinnessen and Bauer, 2022; Aalam et al., 2022; Patro et al., 2022; Wang et al., 2023; Li et al., 2023; Deldjoo et al., 2023) and tutorials (Ekstrand et al., 2019; Gao and Shah, 2020; Li et al., 2021; Fang et al., 2022; Bigdeli et al., 2022) summarizing this emerging body of work, as well as shared tasks (Biega et al., 2020, 2021; Ekstrand et al., 2023). The fairness questions have typically been framed around disparate quality-of-service—*e.g.*, (Mehrotra et al., 2017, 2018; Neophytou et al., 2022; Wu et al., 2024)—and disparate exposure—*e.g.*, (Biega et al., 2018; Singh and Joachims, 2018, 2019; Diaz et al., 2020; Zehlike and Castillo, 2020; Patro et al., 2020; Wu et al., 2022b). Several recent works (Smith and Beattie, 2022; Raj and Ekstrand, 2020, 2022; Boratto et al., 2022, 2023) have also systemically compared various fairness metrics proposed in the literature.

Beyond fairness, there has been renewed interests in questions of transparency and explainability (Zhang et al., 2020; Anand et al., 2022), addressing misinformation (Zhou and Zafarani, 2018; Kumar and Shah, 2018; Sharma et al., 2019; Collins et al., 2021; Zhou and Zafarani, 2020; Saracco et al., 2021; Guo et al., 2022), and broader ethical concerns in

IR (Schedl et al., 2022). Transparency in IR covers a broad range of scenarios and concerns. Examples include transparency about how the system behaves (Singh and Anand, 2019; Verma and Ganguly, 2019; Singh and Anand, 2018; Zhuang et al., 2020) and how data subjects are represented in search results (Biega et al., 2017; Li et al., 2022). Transparency needs may specifically arise in the context of how information and knowledge access systems modulate *what* and *who* get exposure and influence how we see ourselves and others (Cortiñas-Lorenzo et al., 2024). Different notions of transparency may be relevant here, including but not limited to: System transparency (*i.e.*, how does the system work?), procedural transparency (*i.e.*, in what social norms and processes is the system use embedded?), or transparency of outcomes (*i.e.*, what is the impact of the system’s use on individuals and society?). New transparency needs (Liao and Vaughan, 2023) may also arise in the context of emerging technologies, such as large language models (LLMs) (OpenAI, 2023; Thoppilan et al., 2022; Chowdhery et al., 2022; Touvron et al., 2023).

2.1 Critical perspectives on fairness and ethics research in IR

It is important that we critically assess how the body of fairness and ethics research in IR translates to real world impact. But on this, there is little in the published literature to go on. It is plausible, for example, that the research on ranking fairness has been operationalized in, or at least has influenced the designs of, popular IR systems in recent years, but institutions who build these systems have rarely publicly disclosed any information in that regard, may be to obfuscate details of system design from bad actors or for competitive reasons. Alternatively, it is also plausible that in fact many of these approaches have *not* been operationalized, or only been operationalized in very constrained settings, because industry adoption is lagging behind research or that existing fairness research is built on abstractions and assumptions that are incompatible with real world deployment.

Studies of search logs (Jiang et al., 2013; Chuklin et al., 2015) have historically served an important role in IR research for understanding user and system behaviors and how the two interact. Similarly, online experimentation (Kohavi et al., 2007, 2009, 2020) has been key to validating the outputs of IR research in the real world. In contrast, to the best of our knowledge, there are very few fairness studies—*e.g.*, (Mehrotra et al., 2017, 2018; Raj et al., 2023)—in IR that make use of search logs, while it may be argued that the user interaction data in these logs are exactly where we *should* be looking to identify which social harms are common in practice and understand how exactly they manifest. Similarly, there is an urgent need for validating proposed fairness interventions from the literature through online experimentation, involving real users and real information needs, to ensure that fairness research is grounded in actual needs of the society and does not amount to just academic intellectual exercises. While the lack of log-based studies and online experimentation in fairness research is likely due to the lack of access to commercially-deployed systems and corresponding log data for academic research, we must critically enquire why this manifests so much more severely in fairness research compared to other areas of IR, such as in ranking. To do so, we must expand the frame beyond questions of algorithmic fairness, and examine the very sociopolitical context in which this research is being conducted, the economic incentives and risks that shape it, and the power differentials between institutions and individuals that determine what research is allowed and who is allowed to do it (Whittaker,

2021; Widder et al., 2023). This lack of access to data and systems not only makes it difficult to reproduce, validate, and challenge the claims in existing fairness studies,⁴ but also limits what fairness questions the community is allowed to investigate.

We should also critically reflect on what questions should or should not be framed as fairness problems, and the societal consequences of doing so. For example, one of the motivating scenarios described by Morik et al. (2020) is exposure fairness for search results across different ends of the political spectrum. A similar question has also been studied by Kulshrestha et al. (2017). Casting this as a fairness issue, however, has several problematic implications and consequences. Firstly, this assumes an overly-simplistic frame in which complicated intersecting political ideologies are mapped to a linear spectrum (*e.g.*, left *vs.* right and Democrats *vs.* Republicans) and holds the two ends static rather than a continuously shifting window of acceptable discourse (Lehman, 2014; Giridharadas, 2019a). Furthermore, it also assumes that two ends of a political discourse have equal merit and deserve equal exposure, which amounts to *algorithmic bothsiding*. Finally, in arguing that builders of IR systems should shift exposure out of fairness concerns, it inadvertently normalizes the idea that it is acceptable for institutions and individuals who own these systems to exert enormous influence over public discourse of social and moral import. Instead, it is our perspective that IR needs a fundamentally different and cross-disciplinary approach to these questions, one that centers on engaging and co-producing with other academic sub-fields, such as STS (Hackett et al., 2008) and critical theory of technology (Feenberg et al., 1991; Feenberg, 2002).

We must also assess the validity of constructs that we employ in fairness research. For example, Jacobs and Wallach (2021) point out that race and gender, that are often the focus of group fairness research, are contested constructs, and indeed so is the construct of fairness itself. While several papers on fairness have employed gender as a group variable, Pinney et al. (2023) caution us that much more care should be taken in this practice, for example, to ensure that we respect everyone’s right to self-identify their gender and recognize the fairness concerns of non-binary peoples. Patro et al. (2022) encourage us to move beyond fairness definitions that are grounded in discrete moments and to consider the long-term impact of fairness interventions. Fairness research itself may contribute towards certain negative externalities in the long-term, such as encouraging more pervasive collection of protected demographic attributes and further intensification of data surveillance (Zuboff, 2023) of already marginalized groups in a misguided attempt to bridge the data gaps that may be responsible for the system’s disparate quality of service across groups.

Similarly, an important question that all transparency research must contend with is: *transparency towards what end?* Some works (Polley et al., 2021; Schmitt et al., 2022) motivate transparency as a means to increase user trust in the system. However, we should be critical of whether that trust is warranted, or whether transparency mechanisms could in

4. For example, some studies on log-based fairness audits depend on the availability of user provided demographic attributes. In our experience, such attributes are often available only for a small subset of users—such as for signed-in users—who typically are more loyal users of the product and on average much better satisfied with system performance than the general population. Audits based on this sub-population may significantly under-report bias and unfairness issues faced by users. Better understanding of such practical challenges can motivate the community to work on specific research questions, such as fairness considerations under distributional shifts and with noisy (Ghazimatin et al., 2022; Mehrotra and Vishnoi, 2022) or missing demographic attributes (Lazovich et al., 2022; Do and Usunier, 2022).

fact draw users into a false sense of safety and distract them from noticing how the system surveils them and subtly manipulates their behavior (Ravenscraft, 2020; Morrison, 2021). Indeed, Hollanek (2023) argue that “only the sort of transparency that arises from critique—a method of theoretical examination that, by revealing pre-existing power structures, aims to challenge them—can help us produce technological systems that are less deceptive and more just”. It is that kind of critical reflection that we believe should inform transparency research in IR and lead the community towards explicit goals to challenge power (*e.g.*, the power that system owners hold over users) and safeguard user agency.

On the same lines, misinformation research should be motivated by sociotechnical visions for the future of democratic societies, public health, and knowledge production. When we broaden that frame, it becomes apparent that the community must not only focus on automated fact checking, an important research problem, but also understand the social, political, and economic conditions under which misinformation and disinformation is produced and disseminated. The focus of misinformation research then should include identifying, understanding, and addressing the structural mechanisms of misinformation—*e.g.*, data voids (Golebiewski and Boyd, 2019)—as well as ground itself in the articulation of IR’s role in online knowledge production, public health education, and information literacy.

Our perspective here on ongoing research on fairness, transparency, and ethics in IR should **not** be misconstrued as an argument for doing less of this kind of work. Instead, we believe that the community should be explicit and more ambitious about the changes it wants to affect in broader society and conduct research with a clear mapping between the research goals and the desired social impact. Just as Johnson (2014) challenges the notion that open data directly leads to information justice, we want the IR community to be cautious in their assumption that working on narrowly defined questions of fairness, transparency, and ethics necessarily contributes towards practical social good. To be effective in that endeavour, we believe that we should be explicit in articulating our collective visions for our sociotechnical futures, the changes we want to affect in society, and how we envision our research can bring about those changes.

3 Towards sociotechnical change

Sociotechnical imaginaries are not born in vacuum. They are moulded and shaped by our values and our politics. Deliberation over what futures we want to bring into being *is* essentially political, and challenges us to critically reflect on our community’s shared, and yet pluralistic, political values. While the call for explicit political reflection in IR may come as a surprise to some, we need to recognize that our research and the artefacts we produce do not exist outside of the current sociopolitical order (Friedman and Kahn Jr, 2007; Flanagan et al., 2008; Miller, 2021) but as essential cogs in the system, and the absence of political reflection does not imply an absence of politics in our work, but rather translates to implicit complicity in propping up the status quo and neoliberalism (Dourish, 2010; Feltwell et al., 2018; Keyes et al., 2019). Instead, we can learn from how some of our neighboring fields, *e.g.*, HCI and AI, have engaged with these questions, and reflect on how politics shapes and intersects with our own research agendas.

3.1 Prefigurative politics in other IR-adjacent fields

There are several strands of research in IR-adjacent fields that explicate prefigurative politics (Asad, 2019) and ground research in humanistic (Bardzell and Bardzell, 2016a, 2015; Werthner et al., 2024), anti-oppressive and emancipatory (Smyth and Dimond, 2014; Bardzell and Bardzell, 2016a; Kane et al., 2021; Monroe-White, 2021; Saxena et al., 2023), feminist (Wajcman, 2004, 2010; Bardzell, 2010; Bardzell and Bardzell, 2011; Bardzell et al., 2011; Bardzell and Bardzell, 2016b; Bardzell, 2018; D’ignazio and Klein, 2020), queer (Light, 2011; Klippfahn-Karge et al., 2024; Guyan, 2022), postcolonial and decolonial (Irani et al., 2010; Philip et al., 2012; Dourish and Mainwaring, 2012; Sun, 2013; Ali, 2014, 2016; Akama et al., 2016; Irani and Silberman, 2016; Adams, 2021; Mohamed et al., 2020), anti-racist (Abebe et al., 2022), anti-casteist (Kalyanakrishnan et al., 2018; Sambasivan et al., 2021; Vaghela et al., 2022a,b; Shubham, 2022; Kanjilal, 2023), anti-ableist (Williams et al., 2021; Sum et al., 2024), anti-fascist (McQuillan, 2022), abolitionist (Benjamin, 2019; Barabas, 2020; Earl, 2021; Jones and Melo, 2021; Williams and Haring, 2023), post-capitalistic (Feltwell et al., 2018; Browne and Green, 2022), and anarchist (Keyes et al., 2019; Linehan and Kirman, 2014; Asad et al., 2017) epistemologies. Reviewing this full body of literature is out-of-scope of this work but we briefly present a sample to draw from and motivate new IR research agendas for sociotechnical change.

Bardzell and Bardzell (2016a) define humanistic HCI as “any HCI research or practice that deploys humanistic epistemologies (e.g., theories and conceptual systems) and methodologies (e.g., critical analysis of designs, processes, and implementations; historical genealogies; conceptual analysis; emancipatory criticism) in service of HCI processes, theories, methods, agenda-setting, and practices”, and include emancipatory HCI as an aspiration of humanistic HCI. Kane et al. (2021) propose to incorporate emancipatory pedagogy (Freire, 2020) that does “not advocate the oppressed simply rise and overthrow their oppressors. Instead, [...] the oppressors and oppressed create new educational processes that would allow them to work together to create a new type of society that was emancipatory for all”.

In STS, there is a body of work (Longino, 1987; Wajcman, 1991; Hubbard, 2001; Turkle, 2004; Herring et al., 2006; von Hellens et al., 2007; Haraway, 2013; Michelfelder et al., 2017) on gendered inequities caused by technology, and how technology and gender relations mutually shape each other (Wajcman, 2004, 2010). In HCI, Bardzell et al. (Bardzell, 2010; Bardzell and Bardzell, 2011; Bardzell et al., 2011; Bardzell and Bardzell, 2016b; Bardzell, 2018) propose to incorporate feminist theories (Kolmar and Bartkowski, 1999; Friedan, 2010) into research and practice. Bardzell (2010) posits that feminist theories can contribute to interaction design both by critiquing and by generating new insights that inform and shape designs and design processes. We can see feminist HCI in practice in the works of Dimond et al. (Dimond, 2012; Dimond et al., 2013). In the context of speculative design (Auger, 2013), Martins (2014) emphasize the need for intersectional (Crenshaw, 1989; McCall, 2005; Crenshaw, 2013) feminist lens in critiquing and dismantling structures of oppression. Feminist theory, methods, and epistemologies have also influenced AI research (Adam, 1995, 2013; Wellner and Rothman, 2020; Browne et al., 2023; Toupin, 2024). Erscoi et al. (2023) highlight how women are erased from and by AI technologies. Leavy et al. (2021) propose ethical data curation approaches grounded in feminist principles. Using feminist epistemology, Huang et al. (2022) critique existing practices of explainable AI, and Varon and Peña

(2021) critique practices for obtaining digital consent in data extractivist practices in AI. Gender theory have also been employed in these fields in the forms of Queer HCI (Light, 2011) and Queer AI (Klipphahn-Karge et al., 2024). Both highlight *queering* (Sta, 1997; Brooks et al., 2021) as a tactic to challenge the basis on which categories are constructed.

Irani et al. (2010) define postcolonial computing as one that is “centered on the questions of power, authority, legitimacy, participation, and intelligibility in the contexts of cultural encounter, particularly in the context of contemporary globalization. [...] It asserts a series of questions and concerns inspired by the conditions of postcoloniality”. Avle et al. (2017) criticize the “colonizing impulse” to valorize “universal methods” that are supposedly appropriate across cultural and geopolitical boundaries; instead we can draw from works (Alsheikh et al., 2011; Wong, 2012; Winschiers-Theophilus and Bidwell, 2013; Shaw et al., 2014; Fox and Le Dantec, 2014; Ahmed et al., 2015; Akama et al., 2016; Sambasivan et al., 2021) that center on and incorporate indigenous and non-western values and ethics in the critique and development of technologies. Ali (2014) argue for decolonial computing over postcolonial which he criticizes as “Eurocentric critique of Eurocentrism” that “tends to privilege cultural issues over political-economic concerns” and “is noticeably silent about past injustices and does not engage with the matter of reparations”.

Kaba (2021) define abolition as “a long-term project and a practice around creating the conditions that would allow for the dismantling of prisons, policing, and surveillance and the creation of new institutions that actually work to keep us safe and are not fundamentally oppressive”. The movement challenges us to move beyond the default assumptions and world views of the carceral state and to dismantle the prison-industrial complex. Incorporating abolitionist values in computing requires us to oppose carceral technologies, surveillance technologies, and military applications (Earl, 2021).

Post-capitalist computing assumes “a socio-economic model that completely replaces capital as the primary method of organising society” (Feltwell et al., 2018). Among other subjects, research in this area contends with, “the racialized dynamics of labor competition” (Irani, 2018), dismantling Big Tech’s concentration of power (Verdegem, 2022; Srnicek, 2017), and imagining post-work futures (Browne and Green, 2022; Butler, 2018; Srnicek and Williams, 2015). Perhaps, the challenges in this area are best summed up in the words of Jameson (2003): “it is easier to imagine the end of the world than to imagine the end of capitalism”.

These different political lens lead to imagining new futures of computing but there are some themes that cut across them. Firstly, they all recognize technology and society as mutually shaping, and reject both technological determinism (Greene et al., 2019) and the frame in which technology exists in, what Pfaffenberger (1988) calls, a fetishised form (Marx, 1867) where technology is disembodied and disconnected from social relations. Secondly, they recognize that the perspectives, goals, and approaches across this spectrum while sometime distinct are also intersecting. Finally, they all call for structural changes and progress towards alternative futures for society and computing. Perhaps, these aspirations are best articulated by Keyes et al. (2019): “radically reorienting the field towards creating prefigurative counterpower—systems and spaces that exemplify the world we wish to see, as we go about building the revolution in increment”. To affect said changes we need to both recognize the politics of our work and ground it in broader context of political actions (Wickenden, 2018; Moore, 2020; Green, 2021; Widder et al., 2023; Young et al., 2021).

3.2 Proposals for IR

The survey of works presented in Section 3.1 hopefully provides some seeds of ideas for how IR research can be driven by radical new sociotechnical imaginaries. This is not to imply that these other IR-adjacent fields have achieved the desired success from these approaches, in fact there are some evidence (Chivukula and Gray, 2020) that point otherwise. Rather, we should recognize that how values and politics can inform computing research is still an open question, and they may apply differently to IR than these other fields. The challenge then for the community is to collectively engage and push towards sociotechnical change. In the remaining of this section, we discuss how we imagine some of these frames and values can guide us towards open challenges in information access. However, these examples should be interpreted as just that, as examples, not our recommendation for specific research questions the community should focus on. The actual research agenda should be developed through participatory processes that simultaneously focuses on both identifying technical research questions and building diverse communities with shared understanding of these challenges and shared commitments to address them.

Through the lens of feminist, queer, and anti-racist IR, we could critique existing approaches to ranking fairness, not only in terms their use of socially constructed categories, such as race and gender (Pinney et al., 2023), but question if it is the appropriate framing at all for the problems it purports to solve. For example, instead of trying to algorithmically fix under-representation of women and people of color in image search results for occupational roles, we could reclaim that digital space as a site of resistance and emancipatory pedagogy by allowing feminist, queer, and anti-racist scholars, activists, and artists to create experiences that teach the history of these movements and struggles.

In context of decolonial IR, ongoing fairness research may co-develop relevant local intervention strategies with legal scholars in recognition of significant differences in legal treatment of topics such as *affirmative action* across geographies. This shifts fairness research away from abstract universal notions of bias and fairness towards locally-significant societal impact (*i.e.*, *think local, act local*).

Anti-oppressive IR research may concern itself with questions such as: *Can we translate Freire’s (Freire, 2020) anti-oppressive pedagogy to strategies for anti-oppressive information access? Can search result pages support dialogical interactions between searchers that allows for communities of searchers to add context to the search results, as an alternative to centralized moderation?* Unlike conversation search, that is framed as interaction between the user and the system, the idea of dialogical search interfaces challenges us to build sophisticated sociotechnical solutions to support dialog between searchers in context of specific search intents in ways that leads to knowledge production and better digital literacy. Anti-oppressive and anti-capitalist perspectives may also motivate us to reimagine search and recommender systems as decentralized and federated.

IR research may also employ these lenses as instruments of critique. For example, in the enterprise context, Gausen et al. (2023) adopt decolonial and anti-capitalist lens to expose how information and knowledge access systems may commodify and appropriate knowledge from workers. We should also critically challenge the employment of what Gray and Suri (2019) calls Ghost Work in IR research both as a labor issue and through the lens of decolonization. In abolitionist IR, we must ensure that the technologies we build cannot

be used for surveillance or any other military or carceral applications. The community may also consider more radical direct actions such as developing critical theories of information access, or collectively organizing to abolish Big Tech (Kwet, 2020).

3.3 Where do we start?

We are calling for not only a significant shift in what the IR community works on, but fundamentally changing the arrangements within our community that determine on an ongoing basis our research agendas. Beyond explicating our values and sociotechnical imaginaries, we need to develop frameworks that help us appropriately prioritize societal needs against the needs of the user, the publisher, and the platform owners. We also need new research that reimagines how IR can be informed by different epistemologies and political theories. Finally, we must also critically reexamine the arrangements within our community and create spaces for shared sense-making in collaboration with those outside of our field. We elaborate on these further in this section.

In both academic research and industrial deployment, IR places a strong emphasis on the needs of the user (consumer). This focus motivates various lines of research including: understanding user needs (through lab studies, log analysis, surveys, *etc.*), optimizing the search system towards those needs (*e.g.*, relevance optimization, personalization, and improving response time), and validating that proposed system changes indeed benefit the user (through online experimentations and further lab studies). Salient in industry settings are the needs of the system owners—*e.g.*, revenue, market share, and brand—that drive significant decisions for system design and deployment, but have historically been of lesser concern to academic IR research. Real-world IR system deployments also engage with content producers and publishers, *e.g.*, web publishers and the search engine optimization community for web search engines, and artists for music recommender systems); although how their needs are weighed against the needs of system owners and users may vary, *e.g.*, (Guttenberg, 2012; Plaugic, 2015). IR research has considered questions of fairness between producers, but have rarely focused on the power differentials between system owners and producers, and its implications for producers.

Societal concerns in both IR research and industry settings have commonly been framed through a narrow lens of harm mitigation, such as “*how do we make the IR system more fair?*” and “*how do we reduce misinformation in search results?*”, without fundamentally challenging the frames in which these systems are designed and deployed, *e.g.*, centralized control and profit incentives (Mager, 2012; Taplin, 2017). IR systems are deeply embedded in sociopolitical and organization context. However, instead of grounding IR research in questions around its role in online and institutional knowledge production, literacy and informed citizenry, public health education, and social justice, the community has typically constrained themselves to improving measurable system attributes like relevance and efficiency.

Articulating different stakeholder concerns is a prerequisite to any conversation about reprioritizing our research agendas and recentering IR research on societal needs. In Figure 1, we propose a hierarchy of stakeholder needs that IR research should concern with. Contrary to the status quo, we believe that IR system design and research should explicitly reflect how these systems should contribute to knowledge production, public education, and social

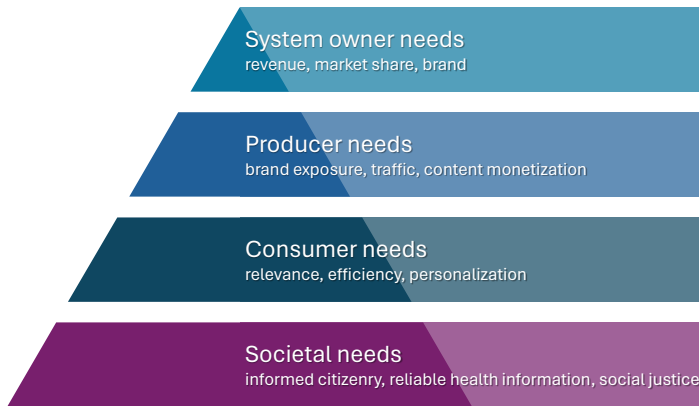


Figure 1: Hierarchy of IR stakeholder needs. More fundamental and critical needs are at the bottom of the pyramid. This figure is inspired by Maslow’s Hierarchy of Needs (Maslow, 1958) and Siksika (Blackfoot) way of life (Ravilochan, 2021).

movements, and that broader framing of societal concerns should be the most fundamental stakeholder need that should inform and shape IR research. This should then be followed by concerns of the consumer and producer needs, and lastly the needs of the system owners themselves. The needs of the consumers, producers, and system owners should not override the need of the collective society, just as the needs of the system owners should not be prioritized over the needs of the consumer and the producers. This is the shift in our research thinking, agendas, and impact that we are pushing for in this paper.

To realize structural changes in IR and meaningfully challenge dominant imaginaries we must also invest in research that specifically explores and builds on the connections between IR and different epistemologies and social and political theories. Examples of such work may consider how IR system design can support different models of democracy (Vrijenhoek et al., 2021; Helberger, 2021) and emancipatory aspirations (Mitra, 2025). We must also ensure that such research does not happen in isolation but is grounded in a collective effort to build a movement within the IR community. This requires us to acknowledge and value community building as an important part of IR research. In this context, it is important to emphasize that our call for re-centering societal needs in IR research must not be confused for a call for technosolutionism. Quite the contrary, we believe we need safe spaces where IR researchers can engage with scholars from other diverse fields, legal and policy experts, activists, and artists, in a recognition of a collective struggle to develop shared understandings of the core challenges and what IR research can offer to this process. These spaces are critical for cross-pollination of ideas and shared sense-making, which are vital for realizing structural changes. As Tsing aptly puts it:

“We are contaminated by our encounters; they change who we are as we make way for others. As contamination changes world-making projects, mutual worlds—and new directions—may emerge. Everyone carries a history of contamination; pu-

urity is not an option. [...] staying alive—for every species—requires livable collaborations. Collaboration means working across difference, which leads to contamination. Without collaborations, we all die.”

– Anna Lowenhaupt Tsing

The mushroom at the end of the world:

On the possibility of life in capitalist ruins (Tsing, 2015)

Through these exchanges we must build relations of solidarity; work together to articulate pluralistic desirable sociotechnical futures, co-develop theories of change and new research agendas to support our aspirations; vigilantly critique our own assumptions and the structures that we exist in and conduct our research in; and critically assess the impact of our work not by publications or scholarly metrics but in terms of affecting real social change. In other words, before we can transform our platforms and systems we need to transform our communities and how we conduct research. And it is vital that we approach these spaces with curiosity and humility; in recognition of our own incomplete understanding of the world; open to change and be changed by these encounters. In this context, it would do IR researchers good to keep in mind the words of activist Lilla Watson.

“If you have come here to help me you are wasting your time, but if you have come because your liberation is bound up with mine, then let us work together.”

– Lilla Watson and other members of an
Aboriginal Rights group in Queensland⁵

This praxis will be unfamiliar and the learning curve frustrating; but hopefully the mutual shaping of IR technology and society will be ultimately rewarding for us all. And we must embark on this transformative journey fully acknowledging that challenging dominant imaginaries is challenging power itself. To affect counter-imaginaries we must therefore actualize counter-structures and alternative funding mechanisms that can sustain this research in the face of likely reprisals from those whose power and visions are threatened by our proposed transformations.

3.4 Who should do this work?

In his highly influential work “Pedagogy of the oppressed”, Freire (2020) criticizes the “banking” concept of education: “Education thus becomes an act of depositing, in which the students are the depositories and the teacher is the depositor. [...] knowledge is a gift bestowed by those who consider themselves knowledgeable upon those whom they consider to know nothing. [...] But the humanist, revolutionary educator cannot wait for this possibility to materialize. From the outset, her efforts must coincide with those of the students to engage in critical thinking and the quest for mutual humanization. His efforts must be imbued with a profound trust in people and their creative power. To achieve this, they must be partners of the students in their relations with them.” These profound words present a

5. While often credited for this quote, Watson explained (Reblog of report from Northland Poster) that this came out of a collective process by an Aboriginal rights group in Queensland that she was part of.

critique that we believe is also relevant to how we conduct fairness and ethics research in the IR community today.

Our default modes of doing research, like pedagogy, takes for granted the validity of *experts* and *expertise* as integral to knowledge production. While this may be effective when our research involves improving ranking or developing new evaluation measures, we posit it is the wrong approach when the goal of the research is to affect sociotechnical change. It would be narcissistic to imagine that sociotechnical research of moral import can be conducted by the few in our community and then “deposited” to the rest. Instead these concerns should be central to all IR research and we should collectively engage in dialogical praxis. Ultimately, research that attempts to affect sociotechnical change does not just transform technology, but also the researcher, and both are necessary for progress.

To challenge the homogeneity of the future imaginaries—saliently bound by colonial, cisheteropatriarchal, and capitalist ways of knowing the world—that shape our research, we need broad and diverse participation from our community. But it is also in that very context that we must critically reflect on the topic of membership in our community itself. ACM SIGIR has a commendable emphasis on the topic of Diversity, Equity, and Inclusivity (DEI) (Verberne et al., 2024; Kobayashi, 2017; Goharian and Bast, 2022; Goharian et al., 2023, 2021). For our sociotechnical imaginaries to be informed by pluralistic social, cultural, and political perspectives we not only need significantly improved representation from historically underrepresented and marginalized groups in our community, but also be inclusive of their politics and world views. Inclusion of people without inclusion of their history and struggles is simply tokenism and epistemic injustice (Fricker, 2007). That is why we believe that we, as a community, should go beyond just diversity and inclusivity (D&I), and enshrine as our goal Justice, Equity, and Diversity & Inclusivity (JEDI)—in which context D&I is both a means towards justice and equity, and also an end in itself. As Keyes et al. (2019) put it: “This must be about more than just bodies: it is not diversity if we only accept marginalised people who are stripped of the epistemic models that underpin experiences of being Other, or have the work they draw from those models held to an unequal standard of legitimacy”.

Lastly, we reiterate the important role of industry researchers in this process. They should take advantage of their proximity to large-scale systems to identify, understand, and communicate concerns of societal import and partner with academia to work on those challenges. The spaces they occupy are also sites for resistance (Wickenden, 2018; Widder et al., 2023).

4 Why now?

The arguments we present in this paper to reimagine our sociotechnical futures and center IR research on societal needs have always been relevant to the field. However, there is a confluence of several factors that makes this discourse particularly relevant in the present moment. Research communities constantly evolve, shaped by ideas and developments from both within the field and adjacent communities, and in response to real world events and changing societal needs. In case of IR, we believe we are seeing significant developments in both context at present: (i) A fast-changing landscape across IR and adjacent fields, such as natural language processing (NLP), HCI, ML, and AI, spurred by recent progress in LLMs

and other generative AI approaches, and (ii) an increasing recognition of the role of technology, and the communities that build it, in determining our collective futures. Consequently, there is a shared sense in the community that right now both IR technologies and IR research have been made malleable and are undergoing transformative changes under these forces of emerging new computational capabilities and evolving societal needs (Azzopardi et al., 2024a). This presents a timely opportunity for the field to consciously, collectively, and ambitiously engage in purposeful dialog about the future of the field while metaphorically the “IR(on)” is hot and before it is irrevocably shaped by unexamined imaginaries of those with power and influence over present day IR research. In doing so, we must also critically reflect on “*where do we want to go?*” (i.e., our sociotechnical imaginaries), “*how do we get there?*” (i.e., our theories of change), and “*who will we go there with?*” (i.e., our relationships, and that of our work, with other disciplines, governments, industry, and society). These considerations should drive future IR research as a whole, and we should accept this opportunity to re-center our research agendas on societal needs while dismantling the artificial separation between the work on fairness and ethics in IR and the rest of IR research.

LLMs are changing how we access information. The natural language generation capabilities of LLMs are having a profound effect on how we access information and in what context. Conversational search interfaces have gone from being aspirational (Anand et al., 2021; Metzler et al., 2021) to being deployed at web-scale (e.g., Bing Chat⁶ and Google Bard⁷) in a span of two years. While the long-term social implications of inserting an LLM between a retrieval system and the information seeker should rightly be met with rigorous skepticism (Shah and Bender, 2022), natural language interfaces are already impacting how we interact with IR systems. The ubiquitous search box is being challenged as IR becomes more *context-driven* than *user-driven* as a consequence of LLMs increasingly embedding themselves in user’s work processes—e.g., Microsoft Copilot for M365 (Mehdi, 2024; Warren, 2024)—and interacting with the IR system on the user’s behalf, under retrieval-augmentation (Lewis et al., 2020; Zamani et al., 2022).

While we should be excited with the new prospects that these emerging AI technologies unlock and recognize that they will shape how we access and interact with information in the future, we must not be duped by AI techno-determinism into believing that there is a single pre-determined path forward. Instead, we must hold pluralistic views of what IR’s future, one that is yet to be determined, looks like and how these technologies will take us there. In a study of top-cited AI papers, many of which are coauthored by researchers affiliated with industry or elite universities, Birhane et al. (2022) find that the dominant values expressed and operationalized support concentration of power. So, we must ask: *in what new ways can we imagine accessing and interacting with information, aided by LLMs, if large-scale IR systems were not just a purview of Big Tech? How would LLMs empower us to reimagine IR systems whose explicit goal is to dismantle hierarchies and redistribute power, not to centralize it? What role would AI technologies play in information access that is built explicitly to facilitate dialogical social processes for knowledge production, world building, and our collective struggles for universal emancipation?* It is critical that we have

6. <https://chat.bing.com/> (now Microsoft Copilot (Mehdi, 2023))

7. <https://bard.google.com/>

these conversations now in the face of ongoing massive technology-driven power shifts in favor of dominant established platforms that grants their visions of the future normative status and shrinks the space for any critique, resilience, or counter-imaginaries.

LLMs are shifting priorities of IR research. Over the last decade, deep learning technologies became the new hammer in the toolbox for IR research (Mitra and Craswell, 2018; Lin et al., 2020; Fan et al., 2022), dominating IR publications with nearly four out of five papers at the ACM SIGIR 2020 conference being related to deep learning by some estimates (Mitra, 2021). One particular focus of neural IR has been on estimating relevance of information artefacts (*e.g.*, documents) to an information intent (*e.g.*, as expressed by a search query) for ranking, a central problem in IR. Curiously, many of the key ingredients for this research, such as the Transformer architecture (Vaswani et al., 2017) and the idea of pretrained LLMs, like BERT (Devlin et al., 2019), came from fields adjacent to IR; correspondingly, shifting the focus within the field more towards adapting these models for the relevance estimation task—*e.g.*, (Nogueira and Cho, 2019)—and making them more efficient (Fröbe et al., 2024).

More recently, Thomas et al. (2024) demonstrated that LLMs, like GPT-4 (OpenAI, 2023), are able to estimate the actual searcher’s preference for documents, given their query, better than several populations of human relevance assessors. This technology has already been deployed in production at Bing.⁸ Putting it bluntly, these LLMs may be getting close to the best we can expect with machine learned general purpose relevance estimators. If these claims stand the test of time, it may mark a watershed moment for IR research. Speculatively, we may see the IR community further shifting towards: (i) Improving efficiency of these models, (ii) focusing on more specialized information needs—*e.g.*, tip-of-the-tongue information needs (Arguello et al., 2021), and (iii) increasing investments in measurement and evaluation—*e.g.*, for emerging new IR scenarios, such as retrieval-enhanced machine learning (Zamani et al., 2022). Alternatively, we may ask: *How can the IR community meet this moment, not with apprehension nor with unchallenged exuberance for progress happening in adjacent fields, but truly grasp this opportunity to redefine what it means to work on IR research? Can we be unabashedly discontent with imagining the future of IR based wholly on what new AI progress makes plausible, and instead reimagine our field as a place where knowledge, culture, and radical aspirations meet to demand of technology to make new futures possible?* Alternatively, if we fail to articulate an aspiring vision for IR research, we risk as a field being reduced to just an application of AI.

LLMs are raising new sociotechnical concerns. It is well-known that language models reproduce, and even amplify, harmful stereotypes and biases of moral import (Friedman and Nissenbaum, 1996) that are present in their training data (Bolukbasi et al., 2016; Caliskan et al., 2017; Gonen and Goldberg, 2019; Blodgett et al., 2020; Bender et al., 2021; Abid et al., 2021). One particular mitigation strategy involves using ML approaches that learn from human preferences, such as reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019), to align with “human values” (Kasirzadeh and Gabriel, 2023; Tamkin et al., 2021). While RLHF has been quite effective in constraining LLMs from producing certain types of offensive and harmful content, we must be wary of any framing of AI ethics, such as *AI alignment* (Russell et al., 2015; Gabriel, 2020; Gabriel and Ghazavi,

8. https://twitter.com/IR_oldie/status/1659413086007328768

2021), that presupposes the existence of *universal values* but that assumption does not hold true in reality (Prabhakaran et al., 2022; Birhane and Cummins, 2019; Jobin et al., 2019; Png, 2022; Sambasivan et al., 2021). This is particularly concerning if we look at this in the context of power asymmetries that exist between powerful private corporations, who have outsized influence over what values these models are optimized for, and those who use these models or are represented in some fashion in the model outputs. This is further compounded by the lack of appropriate mechanisms for civil society to participate in and challenge these choices. Indeed, by placing these controls in the hands of the privileged few, we risk further concentration of power. The concerns of biases in what these models produce, and even biases in what context they refuse to generate (Urman and Makhortykh, 2023), and who gets to influence those decisions have serious implications for information access and society. Finally, the development of LLMs themselves may involve potential harms to authors (Davis, 2023; Lawler, 2023; Browne, 2023; Shetler, 2024; Milmo, 2024), crowdworkers (Gray and Suri, 2017; Ekbja and Nardi, 2017; Gray and Suri, 2019; Roberts, 2019; Jones-Imhotep, 2020; Roberts, 2021; Williams et al., 2022; Perrigo, 2023; Dzieza, 2023), and even the environment (Bender et al., 2021; Patterson et al., 2021; Bommasani et al., 2021; Wu et al., 2022a; Dodge et al., 2022; Patterson et al., 2022).

In a critical perspective, Shah and Bender (2022) recommend that IR research should focus on developing appropriate guardrails in anticipation of the social implications of these emerging technologies and not be constrained by a singular LLM-powered conversational search vision for IR. With this we agree, we should be excited by the new capabilities unlocked by recent progress in LLMs but must not limit our imaginations and aspirations by only what LLMs make plausible. We must also consider the broad sociotechnical implications of deploying these emerging technologies in the context of information access (Mitra et al., 2024), and their systemic consequences and risks.

Our relationships with adjacent communities are changing and so is how we do research. Not long ago, many in the IR community would tout IR, and specifically web search, as a rare success story of real-world application of ML, AI, and NLP technologies. A sea change in these adjacent communities in the last decade have shifted this balance. Now, many see IR as just another NLP task, sometimes included in NLP benchmarks (*e.g.*, HELM (Liang et al., 2022)) for evaluating ML and AI models. With retrieval-augmented LLMs, IR is auditioning for a new role, as a tool for AI models, curiously inverting the relationship between these technologies, where AI was one of many in the IR toolbox.

These communities are also undergoing significant changes in research culture, often influencing each other on the way. One particular trend in NLP, and the broader ML and AI communities, that has influenced IR, is *leaderboard-driven research*. Several NLP leaderboards (Bajaj et al., 2016; Goyal et al., 2017; Joshi et al., 2017; Rajpurkar et al., 2018; Wang et al., 2018; Talmor et al., 2018; Wang et al., 2019; Kwiatkowski et al., 2019; Yang et al., 2018; Liang et al., 2020) have been instrumental in encouraging progress on specific tasks. IR has a long history of focus on shared tasks and benchmarks, notably TREC (Voorhees et al., 2005) that has been a venue for developing new tasks and benchmarks, as well as building research communities with shared interests around them. What differentiates IR benchmarking in venues such as TREC from NLP leaderboards is that the former is framed not as a *competition*, but as a *coopetition*. In a competition, the goal of the participant is to

outperform others, while in a coopetition the participants share a collective goal to develop a better understanding of both the task and the models in the spirit of scientific enquiry. In the words of IR researcher Ian Soboroff:⁹ “The datasets were not built to be solved. They were built as tools to understand the problem and the systems we build to ‘solve’ them.”; or as Voorhees (2021) put it: “Coopetition is defined as competitors cooperating for the common good... While competition can give one a bigger piece of the pie, cooperation makes the whole pie bigger.” By emphasizing the goals of community development and understanding of the tasks and the models, these evaluation effort try to promote scientific enquiry over sportive competition. Even the MS MARCO ranking task that initially started as a competition later reframed itself as a coopetitive evaluation effort (Craswell et al., 2021; Lin et al., 2022).

As a community we should cautiously embrace insights and trends from our neighboring fields. However, we should not let IR be minimized to just an evaluation task which undermines the critical responsibility that IR researchers owe to the broader society. Similarly, while leaderboards and competitions may be effective in creating excitement and increasing participation in certain tasks, we must be mindful of the implications of potentially a large section of the IR community being driven predominantly by these practices. When the goal is to win, then scientific inquiry takes a back seat, and the ones with the most compute and data resources take the metaphorical steering wheel. It risks, what Gausen et al. (2023) call, albeit in a different context, shifting from *praxis*—*i.e.*, “reflection and action directed at the structures to be transformed” (Freire, 2020)—to *proxies*, *i.e.*, optimizing towards proxy quantitative measures of outcomes. Actions in this context may refer to any research activity, including but not limited to: formalization, design, experimentation, publishing, artefact creation, open sourcing, and community building; and examples of proxies include state-of-the-art (SOTA) performance on benchmarks and leaderboard rankings that do not translate to better scientific understanding or positive impact on people.¹⁰

Yet another relationship that we must critically examine is the one between industry and academia. Whittaker (2021) point out that the concentration of data and compute resources, two key ingredients in recent advances in AI, in the hands of few large tech corporations is giving these same institutions tremendous power to shape academic research agenda. Big tech also shapes academic research agendas in various other ways, including academic engagements and employments. In IR, the MS MARCO dataset (Bajaj et al., 2016) and leaderboard (Craswell et al., 2021; Lin et al., 2021, 2022), and the TREC Deep Learning track (Craswell et al., 2020), that has been broadly adopted for benchmarking deep ranking models were produced and is currently maintained by industry researchers. Indeed, the organizers behind these efforts themselves recognized (Craswell et al., 2021; Lin et al., 2022) the critical responsibility that comes with defining critical research tasks for the community—effectively playing “the Pied Piper guiding a significant section of the community down specific lanes of research”—and recommend all benchmark developers to engage in open and inclusive discussions with the rest of the community to critically examine

9. https://twitter.com/ian_soboroff/status/1426901262369439751

10. This is a case of Goodhart’s law (Chrystal and Mizen, 2001; Goodhart, 1975; Hoskin, 1996; Thomas and Uminsky, 2022) whereby improvements on benchmarks and corresponding metrics do not translate to progress on the problem the benchmark was created for, as has been argued for example by Hsia et al. (2023).

their impact. While academia-industry collaboration is critically important for the field to ground our research in real large-scale systems and see our research outputs materialize into real impact on system users, we must also resist the homogenization of our research agendas towards a singular world view put forth by Big Tech capitalism.

The world is changing and so is our relationship to that world. Our world at large is experiencing a confluence of many simultaneous, and mutually reinforcing, forces that are increasingly pushing us towards precarity, including but not limited to: increasing global wealth and income inequality (Chancel et al., 2022), rising global conflicts (Taylor, 2023; United Nations Meetings Coverage and Press Releases), pandemics (Scientist, 2021; Taylor, 2022; for Disease Control et al., 2022), and impending climate catastrophes (Parmesan et al., 2022; IPCC, 2013; Poynting and Rivault, 2024). At a moment when the world needs global solidarity built on trust and consensus, and informed citizenry with robust access to reliable information, online disinformation and misinformation are undermining both (Turrentine, 2022; Treen et al., 2020; Kata, 2010; Allcott and Gentzkow, 2017; Doubek, 2017; Beaumont et al., 2020; Zadrozny, 2024; Swenson and Fernando, 2023). While these complex global challenges require sophisticated and multifaceted response that spans across the political, legal, economic, and technological realms, one thing is for certain that information access research has a role to play. *So, will we answer the call?*

5 Conclusion

Despite good intentions. We must be vigilant and reflexively critique our impact, whether under the model of existing fairness and ethics research in IR or under the proposed shift. The call for pluralistic sociotechnical imaginaries must not in this context be confused with uncritical acceptance of all possible futures as equally valid or desirable. Instead, this is a call for critical examination of our community’s existing normative values and future aspirations. This work not only involves explicating our sociotechnical imaginaries but also engaging critically with the history of technology—*e.g.*, (Merchant, 2023)—and challenging harmful silicon valley ideologies that are counter to the goals of universal emancipation and justice—*e.g.*, (Geburu and Torres, 2023). Above all, we should be wary of any promises of the future that further concentrates wealth and power, or advances any notion of altruism in place of structural change (Giridharadas, 2019b).

Desired outcomes. Having emphasized the importance of theories of change and ensuring that our research has the desired societal impact and not merely constitute an intellectual exercise, it is only fair that we explicate our own desired outcome of this particular work. We authored this paper because we sincerely believe that information access has a critical role to play in determining our collective futures; and that real change can not be realized by fairness and ethics research happening in silos but only when combined with raising social consciousness, organizing, and movement building. We would consider it a failure if this paper is only cited in future IR papers as a passing remark on social responsibility of IR research. Instead, we hope this work sparks many passionate conversations and debates within the community, and radicalizes us to work on issues of social import in collaboration with other disciplines and civil society. But above all, we hope this paper serves as a clarion call to all IR researchers to reflect on why we do what we do. Personally, we hope that the

community continues to build technology not just because we love technology itself, but as an act of radical love for all peoples and the worlds we share. So, we conclude with one final quote for our readers.

“Another world is not only possible, she is on her way. On a quiet day, I can hear her breathing.”

– Arundhati Roy
War talk (Roy, 2003)

Positionality statement

The author of this paper works at a large technology company in the global north. However, the perspectives presented in this work is intended to challenge Big Tech and global north’s view of technology and our collective futures.

Acknowledgments and Disclosure of Funding


The author gratefully acknowledges feedback from Asia Biega, Michael D. Ekstrand, and Ida Larsen-Ledet on the various drafts of this paper. No external funding was received in support of this work.

References

- Syed Wajid Aalam, Abdul Basit Ahanger, Muzafar Rasool Bhat, and Assif Assad. Evaluation of fairness in recommender systems: A review. In *International Conference on Emerging Technologies in Computer Engineering*, pages 456–465. Springer, 2022.
- Veronica Abebe, Gagik Amaryan, Marina Beshai, Ilene, Ali Ekin Gurgun, Wendy Ho, Naaji R Hylton, Daniel Kim, Christy Lee, Carina Lewandowski, et al. Anti-racist HCI: notes on an emerging critical technical practice. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–12, 2022.
- Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 298–306, 2021.
- Alison Adam. A feminist critique of artificial intelligence. *European Journal of Women’s Studies*, 2(3):355–377, 1995.
- Alison Adam. Feminist AI projects and cyberfutures. In *The Gendered Cyborg*, pages 276–290. Routledge, 2013.
- Rachel Adams. Can artificial intelligence be decolonized? *Interdisciplinary Science Reviews*, 46(1-2):176–197, 2021.
- Syed Ishtiaque Ahmed, Nusrat Jahan Mim, and Steven J. Jackson. Residual mobilities: infrastructural displacement and post-colonial computing in bangladesh. In *Proceedings*

- of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pages 437–446, 2015.
- Yoko Akama, Seth Keen, and Peter West. Speculative design and heterogeneity in indigenous nation building. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 895–899, 2016.
- Syed Mustafa Ali. Towards a decolonial computing. In *In Ambiguous Technologies: Philosophical Issues, Practical Solutions, Human Nature*, pages 28–35, 2014.
- Syed Mustafa Ali. A brief introduction to decolonial computing. *XRDS: Crossroads, The ACM Magazine for Students*, 22(4):16–21, 2016.
- Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–236, 2017.
- Tamara Alsheikh, Jennifer A Rode, and Siân E Lindley. (Whose) value-sensitive design: a study of long-distance relationships in an Arabic cultural context. In *Proceedings of the ACM conference on Computer supported cooperative work*, pages 75–84, 2011.
- Avishek Anand, Lawrence Cavedon, Matthias Hagen, Hideo Joho, Mark Sanderson, and Benno Stein. Dagstuhl seminar 19461 on conversational search: seminar goals and working group outcomes. In *ACM SIGIR Forum*, volume 54, pages 1–11. ACM, 2021.
- Avishek Anand, Lijun Lyu, Maximilian Idahl, Yumeng Wang, Jonas Wallat, and Zijian Zhang. Explainable information retrieval: A survey. *arXiv preprint arXiv:2211.02405*, 2022.
- Carmen Ang. How do big tech giants make their billions? *Visual Capitalist*, 25 April 2022. URL <https://www.visualcapitalist.com/how-big-tech-makes-their-billions-2022/>.
- Jaime Arguello, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz. Tip of the tongue known-item retrieval: A case study in movie identification. In *Proceedings of the Conference on Human Information Interaction and Retrieval*, 2021.
- Mariam Asad. Prefigurative design as a method for research justice. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–18, 2019.
- Mariam Asad, Christopher A Le Dantec, Becky Nielsen, and Kate Diedrick. Creating a sociotechnical api: Designing city-scale community engagement. In *Proceedings of the CHI conference on human factors in computing systems*, pages 2295–2306, 2017.
- James Auger. Speculative design: Crafting the speculation. *Digital Creativity*, 24(1):11–35, 2013.
- Seyram Avle, Silvia Lindtner, and Kaiton Williams. How methods make designers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 472–483, 2017.

- Leif Azzopardi, Charles LA Clarke, Paul Kantor, Bhaskar Mitra, Johanne Trippas, Zhaochun Ren, Mohammad Aliannejadi, Negar Arabzadeh, Raman Chandrasekar, Maarten de Rijke, et al. Report on the search futures workshop at ECIR 2024. In *ACM SIGIR Forum*, volume 58, pages 1–41. ACM, 2024a.
- Leif Azzopardi, Charlie Clarke, Paul Kantor, Bhaskar Mitra, Johanne Trippas, and Zhaochun Ren. The search futures workshop (ecir2024), 2024b. URL <https://searchfutures.github.io/>.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.
- Chelsea Barabas. Beyond bias: Re-imagining the terms of “ethical AI” in criminal law. *Georgetown Journal of Law and Modern Critical Race Perspectives*, 12:83, 2020.
- Jeffrey Bardzell and Shaowen Bardzell. What is humanistic HCI? In *Humanistic HCI*, pages 13–32. Springer, 2015.
- Jeffrey Bardzell and Shaowen Bardzell. Humanistic HCI. *Interactions*, 23(2):20–29, 2016a.
- Shaowen Bardzell. Feminist HCI: taking stock and outlining an agenda for design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1301–1310, 2010.
- Shaowen Bardzell. Utopias of participation: Feminism, design, and the futures. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(1):1–24, 2018.
- Shaowen Bardzell and Jeffrey Bardzell. Towards a feminist HCI methodology: social science, feminism, and HCI. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 675–684, 2011.
- Shaowen Bardzell and Jeffrey Bardzell. Feminist design in computing. *The Wiley Blackwell Encyclopedia of Gender and Sexuality Studies*, pages 1–7, 2016b.
- Shaowen Bardzell, Elizabeth Churchill, Jeffrey Bardzell, Jodi Forlizzi, Rebecca Grinter, and Deborah Tatar. Feminism and interaction design. In *CHI’11 Extended Abstracts on Human Factors in Computing Systems*, pages 1–4. 2011.
- Solon Barocas, Asia J Biega, Benjamin Fish, Jędrzej Niklas, and Luke Stark. When not to design, build, or deploy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 695–695, 2020.
- Peter Beaumont, Julian Borger, and Daniel Boffey. Malicious forces creating “perfect storm” of coronavirus disinformation. *The Guardian*, 24, 2020.
- Nicolas Belkin and Stephen Robertson. Some ethical and political implications of theoretical research in information science. In *Proceedings of the ASIS Annual Meeting*, 1976.

- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021.
- Ruha Benjamin. Race after technology: Abolitionist tools for the new jim code. *Social Forces*, 2019.
- Ruha Benjamin. *Imagination: A manifesto*. WW Norton & Company, 2024.
- Nolwenn Bernard and Krisztian Balog. A systematic review of fairness, accountability, transparency and ethics in information retrieval. *ACM Computing Surveys*, 2023.
- Asia J Biega, Azin Ghazimatin, Hakan Ferhatosmanoglu, Krishna P Gummadi, and Gerhard Weikum. Learning to un-rank: quantifying search exposure for users in online communities. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 267–276, 2017.
- Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR ’18, pages 405–414, New York, NY, USA, 2018. ACM.
- Asia J Biega, Fernando Diaz, Michael D Ekstrand, and Sebastian Kohlmeier. Overview of the TREC 2019 fair ranking track. *arXiv preprint arXiv:2003.11650*, 2020.
- Asia J Biega, Fernando Diaz, Michael D Ekstrand, Sergey Feldman, and Sebastian Kohlmeier. Overview of the TREC 2020 fair ranking track. *arXiv preprint arXiv:2108.05135*, 2021.
- Amin Bigdeli, Negar Arabzadeh, Shirin SeyedSalehi, Morteza Zihayat, and Ebrahim Bagheri. Gender fairness in information retrieval systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3436–3439, 2022.
- Abeba Birhane and Fred Cummins. Algorithmic injustices: Towards a relational ethics. *arXiv preprint arXiv:1912.07376*, 2019.
- Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. The values encoded in machine learning research. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pages 173–184, 2022.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of “bias” in NLP. *arXiv preprint arXiv:2005.14050*, 2020.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

- Ludovico Boratto, Gianni Fenu, Mirko Marras, and Giacomo Medda. Consumer fairness in recommender systems: Contextualizing definitions and mitigations. In *European Conference on Information Retrieval*, pages 552–566. Springer, 2022.
- Ludovico Boratto, Gianni Fenu, Mirko Marras, and Giacomo Medda. Consumer fairness benchmark in recommendation. In *Proceedings of the 13th Italian Information Retrieval Workshop (IIR 2023)*. Pisa, Italy, pages 60–65, 2023.
- Paul Brest. The power of theories of change, *Stanford Social Innovation Review*, 2010.
- Lonny J Avi Brooks, Jason Tester, Eli Kosminsky, and Anthony D Weeks. Queering: Liberation futures with afrofuturism. In *Routledge Handbook of Social Futures*, pages 260–274. Routledge, 2021.
- Jacob Browne and Laurel Green. The future of work is no work: A call to action for designers in the abolition of work. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–8, 2022.
- Jude Browne, Stephen Cave, Eleanor Drage, and Kerry McInerney. *Feminist AI: Critical Perspectives on Algorithms, Data, and Intelligent Machines*. Oxford University Press, 2023.
- Ryan Browne. New York Times sues Microsoft, ChatGPT maker OpenAI over copyright infringement. 2023. URL <https://www.cnbc.com/2023/12/27/new-york-times-sue-s-microsoft-chatgpt-maker-openai-over-copyright-infringement.html>.
- Lise Butler. Interview: Technology, capitalism, and the future of the left. *Renewal: A Journal of Social Democracy*, 26(1):18–31, 2018.
- Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.
- Lucas Chancel, Thomas Piketty, Emmanuel Saez, and Gabriel Zucman. *World inequality report 2022*. Harvard University Press, 2022.
- Shruthi Sai Chivukula and Colin M Gray. Bardzell’s “feminist HCF” legacy: Analyzing citational patterns. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- K. Alec Chrystal and Paul D. Mizen. Goodhart’s law: Its origins, meaning and implications for monetary policy. Prepared for the Festschrift in honour of Charles Goodhart, 2001.

- Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click models for web search. *Synthesis lectures on information concepts, retrieval, and services*, 7(3):1–115, 2015.
- Botambu Collins, Dinh Tuyen Hoang, Ngoc Thanh Nguyen, and Dosam Hwang. Trends in combating fake news on social media—a survey. *Journal of Information and Telecommunication*, 5(2):247–266, 2021.
- Ana Maria Ramalho Correia. Information literacy for an active and effective citizenship. In *White Paper prepared for UNESCO, the US National Commission on Libraries and Information Science, and the National Forum on Information Literacy, for use at the Information Literacy Meeting of Experts, Prague, The Czech Republic*, 2002.
- Karina Cortiñas-Lorenzo, Siân Lindley, Ida Larsen-Ledet, and Bhaskar Mitra. Through the looking-glass: Transparency implications and challenges in enterprise AI knowledge systems. *arXiv preprint arXiv:2401.09410*, 2024.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2019 deep learning track. In *Proceedings of the Text REtrieval Conference (TREC)*, 2020.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2021.
- Kate Crawford. The trouble with bias. In *Conference on Neural Information Processing Systems, invited speaker*, 2017.
- Kimberlé Williams Crenshaw. Mapping the margins: Intersectionality, identity politics, and violence against women of color. In *The public nature of private violence*, pages 93–118. Routledge, 2013.
- KW Crenshaw. Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine (pp. 139–168). In *University of Chicago legal forum*, 1989.
- J Shane Culpepper, Fernando Diaz, and Mark D Smucker. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (SWIRL 2018). In *ACM SIGIR Forum*, volume 52, pages 34–90. ACM New York, NY, USA, 2018.
- Nina da Hora, Joana Varon, and Annette Zimmermann. Better utopias: resisting silicon valley ideology and decolonizing our imaginaries of the future, 2024. URL <https://faccconference.org/2024/acceptedcraft>.
- Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *arXiv preprint arXiv:1408.6491*, 2014.
- Wes Davis. Sarah silverman is suing OpenAI and Meta for copyright infringement. *The Verge*, 9 July 2023. URL <https://www.theverge.com/2023/7/9/23788741/sarah-sil>

verman-openai-meta-chatgpt-llama-copyright-infringement-chatbots-artificial-intelligence-ai.

Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. Fairness in recommender systems: research landscape and future directions. *User Modeling and User-Adapted Interaction*, pages 1–50, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://www.aclweb.org/anthology/N19-1423>.

Fernando Diaz, Bhaskar Mitra, Michael D Ekstrand, Asia J Biega, and Ben Carterette. Evaluating stochastic rankings with expected exposure. In *Proc. CIKM*, pages 275–284, 2020.

Catherine D’ignazio and Lauren F Klein. *Data feminism*. MIT press, 2020.

Jill P Dimond. *Feminist HCI for real: Designing technology in support of a social movement*. Georgia Institute of Technology, 2012.

Jill P Dimond, Michaelanne Dye, Daphne LaRose, and Amy S Bruckman. Hollaback! the role of storytelling online in a social movement organization. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 477–490, 2013.

Karlijn Dinnissen and Christine Bauer. Fairness in music recommender systems: A stakeholder-centered mini review. *Frontiers in big Data*, 5:913608, 2022.

Virginie Do and Nicolas Usunier. Optimizing generalized gini indices for fairness in rankings. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 737–747, 2022.

Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A Smith, Nicole DeCario, and Will Buchanan. Measuring the carbon intensity of AI in cloud instances. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 1877–1894, 2022.

James Doubek. How disinformation and distortions on social media affected elections worldwide. *NPR*, 16 November 2017.

Paul Dourish. HCI and environmental sustainability: the politics of design and the design of politics. In *Proceedings of the 8th ACM conference on designing interactive systems*, pages 1–10, 2010.

Paul Dourish and Scott D Mainwaring. Ubicomp’s colonial impulse. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 133–142, 2012.

- Josh Dzieza. AI is a lot of work. *The Verge*, 20 July 2023. URL <https://www.theverge.com/features/23764584/ai-artificial-intelligence-data-notation-labor-scale-surge-remotasks-openai-chatbots>.
- Charles C Earl. Towards an abolitionist ai: the role of historically black colleges and universities. *arXiv preprint arXiv:2101.02011*, 2021.
- Hamid R Ekbia and Bonnie A Nardi. *Heteromation, and other stories of computing and capitalism*. MIT Press, 2017.
- Michael D Ekstrand, Robin Burke, and Fernando Diaz. Fairness and discrimination in retrieval and recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1403–1404, 2019.
- Michael D Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. Fairness and discrimination in information access systems. *arXiv preprint arXiv:2105.05779*, 2021.
- Michael D Ekstrand, Graham McDonald, Amifa Raj, and Isaac Johnson. Overview of the TREC 2022 fair ranking track. *arXiv preprint arXiv:2302.05558*, 2023.
- Lelia Erscoi, Annelies V Kleinherenbrink, and Olivia Guest. Pygmalion displacement: When humanising AI dehumanises women. *SocArXiv. February*, 11, 2023.
- Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. Pre-training methods in information retrieval. *Foundations and Trends in Information Retrieval*, 16(3):178–317, 2022.
- Yi Fang, Hongfu Liu, Zhiqiang Tao, and Mikhail Yurochkin. Fairness of machine learning in search engines. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 5132–5135, 2022.
- Andrew Feenberg. *Transforming technology: A critical theory revisited*. Oxford University Press, 2002.
- Andrew Feenberg et al. *Critical theory of technology*, volume 5. Oxford University Press New York, 1991.
- Tom Feltwell, Shaun Lawson, Enrique Encinas, Conor Linehan, Ben Kirman, Deborah Maxwell, Tom Jenkins, and Stacey Kuznetsov. “Grand visions” for post-capitalist human-computer interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2018.
- Mary Flanagan, Daniel C Howe, and Helen Nissenbaum. Embodying values in technology: Theory and practice. In *Information Technology and Moral Philosophy*, pages 322–353. Cambridge University Press, 2008.
- Centers for Disease Control, Prevention. CDC museum COVID-19 timeline. 2022. URL <https://www.cdc.gov/museum/timeline/covid19.html>

- Sarah Fox and Christopher Le Dantec. Community historians: scaffolding community engagement through culture and heritage. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 785–794, 2014.
- Paulo Freire. Pedagogy of the oppressed. In *Toward a sociology of education*, pages 374–386. Routledge, 2020.
- Miranda Fricker. *Epistemic injustice: Power and the ethics of knowing*. Oxford University Press, 2007.
- Betty Friedan. *The feminine mystique*. WW Norton & Company, 2010.
- Batya Friedman and Peter H Kahn Jr. Human values, ethics, and design. In *The human-computer interaction handbook*, pages 1267–1292. CRC press, 2007.
- Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Transactions on Information Systems (TOIS)*, 14(3):330–347, 1996.
- Maik Fröbe, Joel Mackenzie, Bhaskar Mitra, Franco Maria Nardini, and Martin Potthast. ReNeuIR at SIGIR 2024: The third workshop on reaching efficiency in neural information retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3051–3054, 2024.
- Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3): 411–437, 2020.
- Iason Gabriel and Vafa Ghazavi. The challenge of value alignment: From fairer algorithms to AI safety. *arXiv preprint arXiv:2101.06060*, 2021.
- Ruoyuan Gao and Chirag Shah. Counteracting bias and increasing fairness in search and recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 745–747, 2020.
- Anna Gausen, Bhaskar Mitra, and Siân Lindley. A framework for exploring the consequences of ai-mediated enterprise knowledge access and identifying risks to workers. *arXiv preprint arXiv:2312.10076*, 2023.
- Timnit Gebru and Émile P Torres. Eugenics and the promise of utopia through artificial general intelligence. *First Monday*, 2023.
- Azin Ghazimatin, Matthaus Kleindessner, Chris Russell, Ziawasch Abedjan, and Jacek Golebiowski. Measuring fairness of rankings under noisy sensitive information. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pages 2263–2279, 2022.
- Tarleton Gillespie. Algorithmically recognizable: Santorum’s Google problem, and Google’s santorum problem. In *The Social Power of Algorithms*, pages 63–80. Routledge, 2019.
- A. Giridharadas. How America’s elites lost their grip. *Time Magazine*, 2019a.

- Anand Giridharadas. *Winners take all: The elite charade of changing the world*. Vintage, 2019b.
- Nazli Goharian and Hannah Bast. Report on women in ir (wir 2021) at sigir 2021. In *ACM SIGIR Forum*, volume 55, pages 1–3. ACM, 2022.
- Nazli Goharian, Xin Ma, and Suzan Verberne. Women and disparities in leadership and wages. In *ACM SIGIR Forum*, volume 54, pages 1–3. ACM New York, NY, USA, 2021.
- Nazli Goharian, Faegheh Hasibi, Maria Maistro, and Suzan Verberne. Report on the SIGIR 2022 session on Women in IR (WIR). In *ACM SIGIR Forum*, volume 56, pages 1–2. ACM New York, NY, USA, 2023.
- Stéphane Goldstein. *Informed Societies*. Facet publishing, 2020.
- Michael Golebiewski and Danah Boyd. Data voids: Where missing data can easily be exploited. *Data & Society*, 2019.
- Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proc. NAACL*, pages 609–614, 2019.
- M González. A better-informed society is a freer society. 2021. URL <https://www.unesco.org/en/articles/better-informed-society-freer-society>.
- Charles A E Goodhart. Problems of monetary management: The UK experience. In *Papers in Monetary Economics*, volume 1. Reserve Bank of Australia, 1975.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.
- Mary L Gray and Siddharth Suri. The humans working behind the AI curtain. *Harvard Business Review*, 9(1):2–5, 2017. URL <https://hbr.org/2017/01/the-humans-working-behind-the-ai-curtain>.
- Mary L Gray and Siddharth Suri. *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books, 2019.
- Ben Green. Data science as political action: Grounding data science in a politics of justice. *Journal of Social Computing*, 2(3):249–265, 2021.
- Daniel Greene, Anna Lauren Hoffmann, and Luke Stark. Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning. 2019.
- James Grimmelman. The Google dilemma. *New York Law School, Law Review*, 53:939, 2008.

- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 2022.
- Steve Guttenberg. Is Spotify unfair to musicians?, 2012. URL <https://www.cnet.com/tech/home-entertainment/is-spotify-unfair-to-musicians/>.
- Kevin Guyan. Queer data. 2022.
- Edward J Hackett, Olga Amsterdamska, Michael Lynch, Judy Wajcman, et al. *The handbook of science and technology studies*. Mit Press Cambridge, 2008.
- Max Haiven. *Crises of imagination, crises of power: Capitalism, creativity and the commons*. Bloomsbury Publishing, 2014.
- Blake Hallinan and Ted Striphas. Recommended for you: The Netflix prize and the production of algorithmic culture. *New media & society*, 18(1):117–137, 2016.
- Donna Haraway. A cyborg manifesto: Science, technology, and socialist-feminism in the late twentieth century. In *The transgender studies reader*, pages 103–118. Routledge, 2013.
- Natali Helberger. On the democratic role of news recommenders. In *Algorithms, automation, and news*, pages 14–33. Routledge, 2021.
- Susan C Herring, Christine Ogan, Manju Ahuja, and Jean C Robinson. Gender and the culture of computing in applied it education. In *Encyclopedia of gender and information technology*, pages 474–481. IGI Global, 2006.
- Shana Higgins and Lua Gregory. *Information literacy and social justice: Radical professional praxis*. Library Juice Press, 2013.
- Tomasz Hollanek. AI transparency: a matter of reconciling design with critique. *AI & Society*, 38(5):2071–2079, 2023.
- Keith Hoskin. The ‘awful’ idea of accountability: Inscribing people into the measurement of objects. In R Munro and J Mouritsen, editors, *Accountability: Power, ethos and technologies of managing*. International Thompson Business Press, London, 1996.
- Jennifer Hsia, Danish Pruthi, Aarti Singh, and Zachary C Lipton. Goodhart’s law applies to NLP’s explanation benchmarks. *arXiv preprint arXiv:2308.14272*, 2023.
- Linus Ta-Lun Huang, Hsiang-Yun Chen, Ying-Tung Lin, Tsung-Ren Huang, and Tzu-Wei Hung. Ameliorating algorithmic bias, or why explainable AI needs feminist philosophy. *Feminist Philosophy Quarterly*, 8(3/4), 2022.
- Ruth Hubbard. Science, facts, and feminism. *Women, science, and technology: A reader in feminist science studies*, pages 148–154, 2001.
- IPCC. The physical science basis, Working group I contribution to the UN IPCC’s fifth assessment report (WG1 AR5), 2013. URL <https://www.ipcc.ch/report/ar5/wg1/>

- Lilly Irani. “Design thinking”: Defending silicon valley at the apex of global labor hierarchies. *Catalyst: Feminism, Theory, Technoscience*, 4(1):1–19, 2018.
- Lilly Irani, Janet Vertesi, Paul Dourish, Kavita Philip, and Rebecca E Grinter. Postcolonial computing: a lens on design and development. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1311–1320, 2010.
- Lilly C Irani and M Six Silberman. Stories we tell about labor: Turkopticon and the trouble with “design”. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4573–4586, 2016.
- Abigail Z Jacobs and Hanna Wallach. Measurement and fairness. In *Proceedings of the ACM conference on fairness, accountability, and transparency*, pages 375–385, 2021.
- Fredric Jameson. Future city. *New left review*, 21:65, 2003.
- Sheila Jasanoff and Sang-Hyun Kim. Containing the atom: Sociotechnical imaginaries and nuclear power in the United States and South Korea. *Minerva*, 47:119–146, 2009.
- Sheila Jasanoff and Sang-Hyun Kim. *Dreamscapes of modernity: Sociotechnical imaginaries and the fabrication of power*. University of Chicago Press, 2015.
- Daxin Jiang, Jian Pei, and Hang Li. Mining search and browse logs for web search: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):1–37, 2013.
- Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of AI ethics guidelines. *Nature machine intelligence*, 1(9):389–399, 2019.
- Jeffrey Alan Johnson. From open data to information justice. *Ethics and Information Technology*, 16:263–274, 2014.
- Stephanie T Jones and Natalie Araujo Melo. We tell these stories to survive: Towards abolition in computer science education. *Canadian Journal of Science, Mathematics and Technology Education*, 21:290–308, 2021.
- Edward Jones-Imhotep. The ghost factories: histories of automata and artificial life. *History and technology*, 36(1):3–29, 2020.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.
- Mariame Kaba. *We do this ‘til we free us: Abolitionist organizing and transforming justice*, volume 1. Haymarket Books, 2021.
- Pratyusha Kalluri et al. Don’t ask if artificial intelligence is good or fair, ask how it shifts power. *Nature*, 583(7815):169–169, 2020.
- Shivaram Kalyanakrishnan, Rahul Alex Panicker, Sarayu Natarajan, and Shreya Rao. Opportunities and challenges for artificial intelligence in India. In *Proceedings of the 2018 AAAI/ACM conference on AI, Ethics, and Society*, pages 164–170, 2018.

- Gerald C Kane, Amber G Young, Ann Majchrzak, and Sam Ransbotham. Avoiding an oppressive future of machine learning: A design theory for emancipatory assistants. *Mis Quarterly*, 45(1):371–396, 2021.
- Sucharita Kanjilal. The digital life of caste: Affect, synesthesia and the social body online. *Feminist Media Studies*, pages 1–16, 2023.
- Atoosa Kasirzadeh and Iason Gabriel. In conversation with artificial intelligence: aligning language models with human values. *Philosophy & Technology*, 36(2):1–24, 2023.
- Anna Kata. A postmodern pandora’s box: anti-vaccination misinformation on the internet. *Vaccine*, 28(7):1709–1716, 2010.
- Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828, 2015.
- Os Keyes, Josephine Hoy, and Margaret Drouhard. Human-computer insurrection: Notes on an anarchist HCI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13, 2019.
- Michael Klippbahn-Karge, Ann-Kathrin Koster, and Sara Morais dos Santos Bruss. Introduction: Queer AI. In *Queer Reflections on AI*, pages 1–19. Routledge, 2024.
- Mei Kobayashi. Opportunities for women, minorities in information retrieval. *Communications of the ACM*, 60(11):10–11, 2017.
- Ron Kohavi, Randal M Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 959–967, 2007.
- Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy online controlled experiments: A practical guide to a/b testing*. Cambridge University Press, 2020.
- Ronny Kohavi, Thomas Crook, Roger Longbotham, Brian Frasca, Randy Henne, Juan Lavista Ferres, and Tamir Melamed. Online experimentation at microsoft. *Data Mining Case Studies*, 11(2009):39, 2009.
- Wendy K Kolmar and Frances Bartkowski. *Feminist theory: A reader*. 1999.
- Juhi Kulshrestha, Motahhare Eslami, Johnnatan Messias, Muhammad Bilal Zafar, Saptarshi Ghosh, Krishna P Gummadi, and Karrie Karahalios. Quantifying search bias: Investigating sources of bias for political searches in social media. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 417–432, 2017.
- Srijan Kumar and Neil Shah. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*, 2018.

- Michael Kwet. A digital tech deal: Digital socialism, decolonization, and reparations for a sustainable global economy. *Global Information Society Watch*, 2020.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
- Kelly Lawler. AI already is making its mark in Hollywood. *USA Today*, pages 01D–01D, 2023.
- Tomo Lazovich, Luca Belli, Aaron Gonzales, Amanda Bower, Uthaipon Tantipongpipat, Kristian Lum, Ferenc Huszar, and Rumman Chowdhury. Measuring disparate outcomes of content recommendation algorithms with distributional inequality metrics. *Patterns*, 3(8), 2022.
- Ursula K Le Guin. *The Wave in the Mind: Talks and Essays on the Writer, the Reader, and the Imagination*. Shambhala Publications, 2004.
- Susan Leavy, Eugenia Siapera, and Barry O’Sullivan. Ethical data curation for AI: An approach based on feminist epistemology and critical theories of race. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 695–703, 2021.
- Joseph Lehman. A brief explanation of the Overton window. *Mackinac Center for Public Policy*, 2014.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Ruohan Li, Jianxiang Li, Bhaskar Mitra, Fernando Diaz, and Asia J Biega. Exposing query identification for search transparency. In *Proceedings of the ACM Web Conference 2022*, pages 3662–3672, 2022.
- Yunqi Li, Yingqiang Ge, and Yongfeng Zhang. Tutorial on fairness of machine learning in recommender systems. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2654–2657, 2021.
- Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in recommendation: Foundations, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 14(5):1–48, 2023.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*, 2020.

- Q Vera Liao and Jennifer Wortman Vaughan. AI transparency in the age of LLMs: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 2023.
- Ann Light. HCI as heterodoxy: Technologies of identity and the queering of interaction with computers. *Interacting with computers*, 23(5):430–438, 2011.
- Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *arXiv preprint arXiv:2010.06467*, 2020.
- Jimmy Lin, Daniel Campos, Nick Craswell, Bhaskar Mitra, and Emine Yilmaz. Significant improvements over the state of the art? A case study of the MS MARCO document ranking leaderboard. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2283–2287. ACM, 2021.
- Jimmy Lin, Daniel Campos, Nick Craswell, Bhaskar Mitra, and Emine Yilmaz. Fostering coopetition while plugging leaks: The design and implementation of the MS MARCO leaderboards. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022.
- Conor Linehan and Ben Kirman. Never mind the bollocks, I wanna be anarchi: a manifesto for punk HCI. In *CHI’14 Extended Abstracts on Human Factors in Computing Systems*, pages 741–748. 2014.
- Helen E Longino. Can there be a feminist science? *Hypatia*, 2(3):51–64, 1987.
- Astrid Mager. Algorithmic ideology: How capitalist society shapes search engines. *Information, Communication & Society*, 15(5):769–787, 2012.
- Astrid Mager and Christian Katzenbach. Future imaginaries in the making and governing of digital technology: Multiple, contested, commodified, *New Media & Society* 23(2), Sage Journals, 2021.
- Luiza Prado de O Martins. Privilege and oppression: Towards a feminist speculative design. *Design’s Big Debates DRS International Conference*, 2014.
- Karl Marx. The fetishism of commodities and the secret thereof. *Capital: A Critique of Political Economy*, 1:71–83, 1867.
- Abraham Harold Maslow. A dynamic theory of human motivation. 1958.
- Leslie McCall. The complexity of intersectionality. *Signs: Journal of women in culture and society*, 30(3):1771–1800, 2005.
- Dan McQuillan. Anti-fascist AI. In *Resisting AI*, pages 135–148. Bristol University Press, 2022.
- Yusuf Mehdi. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web, 2023. URL <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/>.

- Yusuf Mehdi. Bringing the full power of Copilot to more people and businesses, 2024. URL <https://blogs.microsoft.com/blog/2024/01/15/bringing-the-full-power-of-copilot-to-more-people-and-businesses/>.
- Anay Mehrotra and Nisheet Vishnoi. Fair ranking with noisy protected attributes. *Advances in Neural Information Processing Systems*, 35:31711–31725, 2022.
- Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. Auditing search engines for differential satisfaction across demographics. In *Proc. WWW*, pages 626–633, 2017.
- Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 2243–2251, 2018.
- Brian Merchant. *Blood in the Machine: The Origins of the Rebellion Against Big Tech*. Little, Brown, 2023.
- Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking search: making domain experts out of dilettantes. In *Acm sigir forum*, volume 55, pages 1–27. ACM New York, NY, USA, 2021.
- Milagros Miceli, Julian Posada, and Tianling Yang. Studying up machine learning data: Why talk about bias when we mean power? *Proceedings of the ACM on Human-Computer Interaction*, 6(GROUP):1–14, 2022.
- Diane P Michelfelder, Galit Wellner, and Heather Wiltse. Designing differently: Toward a methodology for an ethics of feminist technology design. *The ethics of technology: methods and approaches*, pages 193–218, 2017.
- Boaz Miller. Is technology value-neutral? *Science, Technology, & Human Values*, 46(1): 53–80, 2021.
- Dan Milmo. ‘Impossible’ to create AI tools like ChatGPT without copyrighted material, OpenAI says. *The Guardian*, 8 January 2024. URL <https://www.theguardian.com/technology/2024/jan/08/ai-tools-chatgpt-copyrighted-material-openai>.
- Bhaskar Mitra. *Neural Methods for Effective, Efficient, and Exposure-Aware Information Retrieval*. PhD thesis, UCL (University College London), 2021.
- Bhaskar Mitra. Emancipatory information retrieval. *arXiv preprint arXiv:2501.19241*, 2025.
- Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, 2018.
- Bhaskar Mitra, Henriette Cramer, and Olya Gurevich. Sociotechnical implications of generative artificial intelligence for information access. *arXiv preprint arXiv:2405.11612*, 2024.

- Shakir Mohamed, Marie-Therese Png, and William Isaac. Decolonial ai: Decolonial theory as sociotechnical foresight in artificial intelligence. *Philosophy & Technology*, 33:659–684, 2020.
- Thema Monroe-White. Emancipatory data science. 2021.
- Jared Moore. Towards a more representative politics in the ethics of computer science. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 414–424, 2020.
- Felipe Azevedo Moretti, Vanessa Elias de Oliveira, and Edina Mariko Koga da Silva. Access to health information on the internet: a public health issue? *Revista da Associação Médica Brasileira*, 58:650–658, 2012.
- Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 429–438, 2020.
- Sara Morrison. Dark patterns, the tricks websites use to make you say yes, explained. *Vox*, 1 April 2021.
- Stephen M Mutula. Digital divide and economic development: Case study of sub-saharan africa. *The Electronic Library*, 26(4):468–489, 2008.
- Nicola Neophytou, Bhaskar Mitra, and Catherine Stinson. Revisiting popularity and demographic biases in recommender evaluation and effectiveness. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2022.
- Safiya Umoja Noble. Algorithms of oppression. In *Algorithms of oppression*. New York university press, 2018.
- Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019. URL <http://arxiv.org/abs/1901.04085>.
- Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, Michael D Ekstrand, Adam Roegiest, Aldo Lipani, Alex Beutel, Alexandra Olteanu, Ana Lucic, Ana-Andreea Stoica, et al. Facts-ir: fairness, accountability, confidentiality, transparency, and safety in information retrieval. In *ACM SIGIR Forum*, volume 53, pages 20–43. ACM New York, NY, USA, 2019.
- OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Will Oremus. Big tobacco. big pharma. big tech? 2017. URL <https://slate.com/technology/2017/11/how-silicon-valley-became-big-tech.html>.
- Camille Parmesan, Mike D Morecroft, and Yongyut Trisurat. Climate change 2022: Impacts, adaptation and vulnerability, 2022.

- Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *WWW*, pages 1194–1204. ACM / IW3C2, 2020.
- Gourab K Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. Fair ranking: a critical review, challenges, and future directions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1929–1942, 2022.
- David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. 2021.
- David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022.
- Billy Perrigo. Exclusive: OpenAI used Kenyan workers on less than \$2 per hour to make chatgpt less toxic. *Time*, 18 January, 2023. URL <https://time.com/6247678/openai-chatgpt-kenya-workers/>.
- Bryan Pfaffenberger. Fetishised objects and humanised nature: towards an anthropology of technology. *Man*, pages 236–252, 1988.
- Kavita Philip, Lilly Irani, and Paul Dourish. Postcolonial computing: A tactical survey. *Science, Technology, & Human Values*, 37(1):3–29, 2012.
- Christine Pinney, Amifa Raj, Alex Hanna, and Michael D Ekstrand. Much ado about gender: Current practices and future recommendations for appropriate gender-aware information access. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pages 269–279, 2023.
- Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: an overview. *The VLDB Journal*, pages 1–28, 2022.
- Lizzie Plaugic. Spotify’s year in music shows just how little we pay artists for their music. *The Verge*, 2015.
- Marie-Therese Png. At the tensions of south and north: Critical roles of global south stakeholders in AI governance. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1434–1445, 2022.
- Gianfranco Polizzi. Information literacy in the digital age: Why critical digital literacy matters for democracy. *Informed Societies: Why information literacy matters for citizenship, participation and democracy*, pages 1–23, 2020.
- Sayantan Polley, Rashmi Raju Koparde, Akshaya Bindu Gowri, Maneendra Perera, and Andreas Nuernberger. Towards trustworthiness in the context of explainable search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2580–2584, 2021.

- Mark Poynting and Erwan Rivault. 2023 confirmed as world’s hottest year on record, *BBC*, 9 January 2024.
- Vinodkumar Prabhakaran, Margaret Mitchell, Timnit Gebru, and Iason Gabriel. A human rights-based approach to responsible AI. *arXiv preprint arXiv:2210.02667*, 2022.
- Amifa Raj and Michael D Ekstrand. Comparing fair ranking metrics. *arXiv preprint arXiv:2009.01311*, 2020.
- Amifa Raj and Michael D Ekstrand. Measuring fairness in ranked results: An analytical and empirical comparison. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–736, 2022.
- Amifa Raj, Bhaskar Mitra, Nick Craswell, and Michael Ekstrand. Patterns of gender-specializing query reformulation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2241–2245, 2023.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.
- Eric Ravenscraft. How to spot—and avoid—dark patterns on the web. *Wired*, 29 July 2020.
- Teju Ravilochan. The Blackfoot wisdom that inspired Maslow’s hierarchy. *Resilience*. June, 18:2021, 2021.
- Reblog of report from Northland Poster. Attributing words, 2006. URL <https://unnecessaryevils.blogspot.com/2008/11/attributing-words.html>.
- Sarah T Roberts. *Behind the screen*. Yale University Press, 2019.
- Sarah T. Roberts. Your AI is a human. *Your Computer is on fire*, MIT Press, 2021.
- Arundhati Roy. *War talk*. South End Press, 2003.
- Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI magazine*, 36(4):105–114, 2015.
- Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, and Vinodkumar Prabhakaran. Re-imagining algorithmic fairness in India and beyond. In *Proceedings of the ACM conference on fairness, accountability, and transparency*, pages 315–328, 2021.
- F. Saracco, M. Viviani, et al. Overview of ROMCIR 2021: workshop on reducing online misinformation through credible information retrieval. In *ROMCIR 2021 CEUR Workshop Proceedings*, volume 2838, 2021.
- Deepak Saxena, P.J. Wall, and Dave Lewis. Artificial intelligence (AI) ethics: A critical realist emancipatory approach. In *2023 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–5. IEEE, 2023.

- Markus Schedl, Emilia Gómez, and Elisabeth Lex. Retrieval and recommendation systems at the crossroads of artificial intelligence, ethics, and regulation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3420–3424, 2022.
- Anuschka Schmitt, Thiemo Wambsganss, and Andreas Janson. Designing for conversational system trustworthiness: the impact of model transparency on trust and task performance. *ECIS 2022 Research Papers*, 2022.
- New Scientist. Covid-19: the story of a pandemic. *New Scientist*, 10, 2021.
- Chirag Shah and Emily M Bender. Situating search. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, pages 221–232, 2022.
- Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):1–42, 2019.
- Grace Shaw, Margot Brereton, and Paul Roe. Mobile phone use in Australian indigenous communities: future pathways for HCI4D. In *Proceedings of the 26th Australian computer-human interaction conference on designing futures: the future of design*, pages 480–483, 2014.
- Konnor Shetler. AI and consent: What the SAG-AFTRA and WGA agreements tell us about the future of generative AI. *Seton Hall University, Student Works*, 2024.
- Shreeti Shubham. Caste and the digital sphere. *Shuddhashar Youth Journal*, 2022.
- Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *KDD*, pages 2219–2228. ACM, 2018.
- Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5427–5437. Curran Associates, Inc., 2019.
- Jaspreet Singh and Avishek Anand. Posthoc interpretability of learning to rank models using secondary training data. *arXiv preprint arXiv:1806.11330*, 2018.
- Jaspreet Singh and Avishek Anand. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 770–773, 2019.
- Jessie J Smith and Lex Beattie. Recsys fairness metrics: Many to use but which one to choose? *arXiv preprint arXiv:2209.04011*, 2022.
- Thomas Smyth and Jill Dimond. Anti-oppressive design. *Interactions*, 21(6):68–71, 2014.
- Nick Srnicek. *Platform capitalism*. John Wiley & Sons, 2017.

- Nick Srnicek and Alex Williams. *Inventing the future: Postcapitalism and a world without work*. Verso Books, 2015.
- Susan Leigh Sta. The politics question in feminist science and technology projects: the queering of infrastructure. Talk presented at the Technology and Democracy – Comparative Perspectives Conference, University of Oslo, Norway, 1997. URL <http://www.drury.edu/faculty/Ess/Technology/starr.htm>.
- Cella M Sum, Franchesca Spektor, Rahaf Alharbi, Leya Breanna Baltaxe-Admony, Erika Devine, Hazel Anneke Dixon, Jared Duval, Tessa Eagle, Frank Elavsky, Kim Fernandes, et al. Challenging ableism: A critical turn toward disability justice in HCI. *XRDS: Crossroads, The ACM Magazine for Students*, 30(4):50–55, 2024.
- Huatong Sun. Critical design sensibility in postcolonial conditions. *AoIR Selected Papers of Internet Research*, 2013.
- Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.
- Ali Swenson and Christine Fernando. As social media guardrails fade and AI deepfakes go mainstream, experts warn of impact on elections, 2023. URL <https://apnews.com/article/election-2024-misinformation-ai-social-media-trump-6119ee6f498db10603b3664e9ad3e87e>.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.
- Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.
- Dana H Taplin and Hélène Clark. Theory of change basics: A primer on theory of change. *New York: ActKnowledge*, page 72, 2012.
- Jonathan Taplin. *Move fast and break things: How Facebook, Google, and Amazon have cornered culture and what it means for all of us*. Pan Macmillan, 2017.
- Adam Taylor. A historic rise in global conflict deaths suggests a violent new era. 2023. URL <https://www.washingtonpost.com/world/2023/06/29/conflict-war-deaths-global-peace-rise-casualty/>.
- Luke Taylor. Covid-19: True global death toll from pandemic is almost 15 million, says who. *BMJ: British Medical Journal (Online)*, 377:o1144, 2022.
- Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. In *Proc. SIGIR*, 2024.
- Rachel L. Thomas and David Uminsky. Reliance on metrics is a fundamental challenge for AI. *Patterns*, 3(5), 2022.

- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- Sophie Toupin. Shaping feminist artificial intelligence. *New Media & Society*, 26(1):580–595, 2024.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Kathie Treen, Hywel Williams, and Saffron O'Neill. Guest post: How climate change misinformation spreads online, *Carbon Brief*, 2020.
- Anna Lowenhaupt Tsing. *The mushroom at the end of the world: On the possibility of life in capitalist ruins*. Princeton University Press, 2015.
- Sherry Turkle. Computational reticence: Why women fear the intimate machine. In *Technology and women's voices*, pages 44–60. Routledge, 2004.
- J. Turrentine. Climate misinformation on social media is undermining climate action, 2022. URL <https://www.nrdc.org/stories/climate-misinformation-social-media-undermining-climate-action>.
- United Nations Meetings Coverage and Press Releases. With highest number of violent conflicts since second world war, united nations must rethink efforts to achieve, sustain peace, speakers tell security council. <https://press.un.org/en/2023/sc15184.doc.htm>, 2023.
- Aleksandra Urman and Mykola Makhortykh. The silence of the LLMs: Cross-lingual analysis of political bias and false information prevalence in Chatgpt, Google Bard, and Bing chat. *Telematics and Informatics* 96(C), 2023.
- Aleksandra Urman and Mykola Makhortykh. “foreign beauties want to meet you”: The sexualization of women in google’s organic and sponsored text search results. *New media & society*, 26(5):2932–2953, 2024.
- Palashi Vaghela, Steven J Jackson, and Phoebe Sengers. Interrupting merit, subverting legibility: Navigating caste in ‘casteless’ worlds of computing. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2022a.
- Palashi Vaghela, Ramaravind Kommiya Mothilal, Daniel Romero, and Joyojeet Pal. Caste capital on twitter: A formal network analysis of caste relations among indian politicians. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–29, 2022b.
- Joana Varon and Paz Peña. Artificial intelligence and consent: A feminist anti-colonial critique. *Internet Policy Review*, 10(4):1–25, 2021.

- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. of NeurIPS*, 2017.
- Suzan Verberne, Hussein Suleman, Luca Soldaini, and Avijit Ghosh. Report on the SIGIR 2023 session on diversity, equity and inclusivity. In *ACM SIGIR Forum*, volume 57, pages 1–2. ACM, 2024.
- Pieter Verdegem. Dismantling AI capitalism: the commons as an alternative to the power concentration of big tech. *AI & society*, pages 1–11, 2022.
- Manisha Verma and Debasis Ganguly. LIRME: locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1281–1284, 2019.
- Liisa von Hellens, Sue Nielsen, Kaylene Clayton, and Jenine Beekhuyzen. Conceptualising gender and it: Australians taking action in germany. *Proceedings of QualIT2007- Qualitative Research in IT & IT in Qualitative Research*, 2017.
- Ellen M Voorhees. Coopetition in ir research. In *ACM SIGIR Forum*, volume 54, pages 1–3. ACM, 2021.
- Ellen M Voorhees, Donna K Harman, et al. *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridge, 2005.
- Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. Recommenders with a mission: assessing diversity in news recommendations. In *Proceedings of the 2021 conference on human information interaction and retrieval*, pages 173–183, 2021.
- Judy Wajcman. *Feminism confronts technology*. Penn State Press, 1991.
- Judy Wajcman. *Technofeminism*. Cambridge: Polity. 2004.
- Judy Wajcman. Feminist theories of technology. *Cambridge journal of economics*, 34(1): 143–152, 2010.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.
- Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43, 2023.
- Tom Warren. Microsoft’s new Copilot pro brings AI-powered office features to the rest of us, 2024. URL <https://www.theverge.com/2024/1/15/24038711/microsoft-copilot-pro-office-ai-apps>.

- Carol H Weiss. Nothing as practical as good theory: Exploring theory-based evaluation for comprehensive community initiatives for children and families. *New approaches to evaluating community initiatives: Concepts, methods, and contexts*, 1:65–92, 1995.
- Galit Wellner and Tiran Rothman. Feminist AI: Can we expect our AI systems to become feminist? *Philosophy & Technology*, 33(2):191–205, 2020.
- Hannes Werthner, Carlo Ghezzi, Jeff Kramer, Julian Nida-Rümelin, Bashar Nuseibeh, Erich Prem, and Allison Stanger. *Introduction to Digital Humanism: A Textbook*. Springer Nature, 2024.
- Meredith Whittaker. The steep cost of capture. *Interactions*, 28(6):50–55, 2021.
- Dorothy Wickenden. A reckoning at Facebook, *The New Yorker*, 19 February 2018. URL <https://www.newyorker.com/podcast/political-scene/a-reckoning-at-facebook>.
- David Gray Widder, Derrick Zhen, Laura Dabbish, and James Herbsleb. It’s about power: What ethical concerns do software engineers have, and what do they (feel they can) do about them? In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 467–479, 2023.
- Wikipedia contributors. Theory of change — Wikipedia, the free encyclopedia, 2013. URL https://en.wikipedia.org/wiki/Theory_of_Change.
- Darcia Wilkinson, Michael Ekstrand, Janet A. Vertesi, and Alexandra Olteanu. Theories of change in responsible AI. Craft session at the ACM conference on fairness, accountability, and transparency, 2023. URL <https://facctconference.org/2023/acceptedcraft#theor>.
- Adrienne Williams, Milagros Miceli, and Timnit Gebru. The exploited labor behind artificial intelligence. *Noema Magazine*, 13, 2022. URL <https://www.noemamag.com/the-exploited-labor-behind-artificial-intelligence/>.
- Rua M Williams, Kathryn Ringland, Amelia Gibson, Mahender Mandala, Arne Maibaum, and Tiago Guerreiro. Articulations toward a crip HCI. *Interactions*, 28(3):28–37, 2021.
- Tom Williams and Kerstin Sophie Haring. No justice, no robots: From the dispositions of policing to an abolitionist robotics. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 566–575, 2023.
- Heike Winschiers-Theophilus and Nicola J Bidwell. Toward an Afro-centric indigenous HCI paradigm. *International Journal of Human-Computer Interaction*, 29(4):243–255, 2013.
- Pak-Hang Wong. Dao, harmony and personhood: Towards a confucian ethics of technology. *Philosophy & technology*, 25(1):67–86, 2012.
- Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022a.

- Haolun Wu, Bhaskar Mitra, Chen Ma, Fernando Diaz, and Xue Liu. Joint multisided exposure fairness for recommendation. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2022b.
- Haolun Wu, Bhaskar Mitra, and Nick Craswell. Towards group-aware search success. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 123–131, 2024.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.
- Meg Young, PM Krafft, and Michael A Katell. A call for scholar activism: A response to power and technology. *AI Activism*, 28:43, 2021.
- Peter K Yu. Bridging the digital divide: Equality in the information age. *Cardozo Arts & Ent. LJ*, 20:1, 2002.
- Brandy Zadrozny. Disinformation poses an unprecedented threat in 2024 — and the U.S. is less ready than ever, 2024. URL <https://www.nbcnews.com/tech/misinformation/disinformation-unprecedented-threat-2024-election-rcna134290>.
- Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. Retrieval-enhanced machine learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2875–2886, 2022.
- Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of the web conference 2020*, pages 2849–2855, 2020.
- Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021.
- Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking, part i: Score-based ranking. *ACM Computing Surveys*, 55(6):1–36, 2022a.
- Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking, part II: Learning-to-rank and recommender systems. *ACM Computing Surveys*, 55(6):1–41, 2022b.
- Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*, 14(1):1–101, 2020.
- Xinyi Zhou and Reza Zafarani. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*, 2, 2018.
- Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40, 2020.
- Honglei Zhuang, Xuanhui Wang, Michael Bendersky, Alexander Grushetsky, Yonghui Wu, Petr Mitrichev, Ethan Sterling, Nathan Bell, Walker Ravina, and Hai Qian. Interpretable learning-to-rank with generalized additive models. *arXiv preprint arXiv:2005.02553*, 2020.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Shoshana Zuboff. The age of surveillance capitalism. In *Social Theory Re-Wired*, pages 203–213. Routledge, 2023.

Supporting Evidence-Based Medicine by Finding Both Relevant and Significant Works

Sameh Frihat

*University of Duisburg-Essen
Duisburg, Germany*

SAMEH.FRIHAT@UNI-DUE.DE

Norbert Fuhr

*University of Duisburg-Essen
Duisburg, Germany*

NORBERT.FUHR@UNI-DUE.DE

Editor: Haiming Liu

Abstract

In this paper, we present a new approach to improving the relevance and reliability of medical information retrieval, which builds upon the concept of Level of Evidence (LoE). The LoE framework categorizes medical publications into seven distinct levels based on the underlying empirical evidence. Despite LoE framework's relevance in medical research and evidence-based practice, only few medical publications explicitly state their LoE. Therefore, we develop a classification model for automatically assigning LoE to medical publications, which successfully classifies over 26 million documents in MEDLINE database into LoE classes. The subsequent retrieval experiments on the TREC Precision Medicine datasets show substantial improvements in retrieval relevance, when LoE is used as a search filter.

Keywords: Medical Document Facade, Level of Evidence, Evidence-Based Medicine, Medical Search Engines

1 Introduction

In medical research and practice, where findings and decisions directly impact human lives, successful retrieval of relevant and reliable information from scientific literature is paramount. Relevant information includes findings that are directly applicable to a condition under study, whereas reliable means that the findings are consistent under similar conditions (Strage et al., 2023). These concepts contribute to identifying significant information, which implies that findings have a practical and meaningful impact that is not due to chance in terms of its effect on patient care or outcomes (Sathian et al., 2010).

Modern evidence-based medicine (EBM) relies on a systematic approach to guide medical decisions using scientific evidence (Burns et al., 2011; Patrick et al., 2004). A key component of EBM is the Level of Evidence (LoE) framework, which categorizes medical research papers into 7 main distinct levels based on the strength and reliability of evidence reported (Rosner, 2012; Desai et al., 2019; Van de Vliet et al., 2023). This stratification, exemplified by the OCEBM (Oxford Centre for Evidence-Based Medicine <https://www.cebm.net/>) framework (Howick, 2011), ranges from highly rigorous and reliable systematic reviews of randomized controlled trials (Level 1a) to case studies with limited evidential value (Level 4) (Borawski et al., 2007; Group et al., 2002).

Within this framework, each level holds unique significance, representing a specific study design and methodology (Borawski et al., 2007). The hierarchy includes the following Levels of Evidence (LoEs):

- **Level 1a: Systematic Reviews of Randomized Controlled Trials (RCTs).** At the apex of the LoE pyramid are systematic reviews and meta-analyses of well-conducted RCTs. Renowned for their comprehensive analysis of rigorous research, these reviews yield the most authoritative evidence.
- **Level 1b: Individual Randomized Controlled Trials (RCTs).** This level features individual RCTs that contribute crucial insights into causal relationships by evaluating interventions in controlled settings.
- **Level 2a: Systematic Reviews of Cohort Studies.** Systematic reviews of cohort studies provide valuable evidence regarding associations between interventions and outcomes in real-world settings.
- **Level 2b: Individual Cohort Studies.** Individual cohort studies at this level offer meaningful evidence about interventions’ effects within specific populations.
- **Level 3a: Systematic Reviews of Case-Control Studies.** Systematic reviews of case-control studies extend insight into the associations between interventions and outcomes, offering a broader perspective.
- **Level 3b: Individual Case-Control Studies.** Individual case-control studies contribute evidence by exploring the relationships between interventions and outcomes within well-defined contexts.
- **Level 4: Case Series.** At this level, case series provide preliminary evidence about interventions’ effects, although they are limited by their susceptibility to biases and confounding factors.

Although LoE is a crucial parameter for assessing a medical publication’s significance, it is often not explicitly stated in publications, creating a problem for medical information retrieval (IR), where the aim is to retrieve significant medical publications or their content.

Our work addresses the ‘Acquiring’ stage of the 5A’s model (Ask, Acquire, Appraise, Apply, and Assess) in EBM (Leung, 2001), which focuses on retrieving relevant literature to help users find the best available evidence. While LoE and the 5A’s model are distinct frameworks, enabling users to filter retrieved information based on LoE supports the ‘Acquiring’ stage. Future work could explore integrating automated evidence appraisal to complement our retrieval approach.

In this article, we propose an automatic approach to identifying and prioritizing significant works in medical research. First, we develop a classification method for automatically assigning LoE to medical publications, then we use the identified LoE as a search filter in an IR setting. We demonstrate on the TREC PM (Precision Medicine) 2017–2019 collections (Roberts et al., 2017) that using LoE as a filter when retrieving medical papers leads to improved retrieval results, and that the gain is highest for highly evidential medical papers.

2 Related Work

Recent advancements in Evidence-Based Medicine (EBM) have emphasized the role of automation in enhancing the classification and credibility assessment of Clinical Trials and RCTs. A key development in this area is the RobotReviewer system introduced by (Marshall et al., 2014, 2016), which automates the risk of bias assessment in RCTs and provides quality supporting text for bias assessments. This is vital for individual RCTs and also applicable to systematic reviews and meta-analyses of RCTs. The evaluation results indicate that RobotReviewer could match the performance of human reviewers in assessing the risk of bias (Marshall et al., 2016; Marshall and Wallace, 2019), which has been confirmed by several subsequent studies (Soboczenski et al., 2019; Hirt et al., 2021; Arno et al., 2022). Further, contributions from Hartling and Gates (2022) highlight the potential of such automation technologies to refine the quality and efficiency of systematic reviews, particularly in evaluating RCTs.

These advancements mark a significant shift in EBM, offering effective solutions for processing and categorizing extensive medical literature. However, these studies do not cover the full range of evidence levels of medical publications. Instead, they focus only on RCTs and their systematic reviews (Levels 1b and 1a in the LoE framework) and are possibly also applicable to levels 2b and 2a (cohort studies and their systematic reviews).

While large-scale metadata sources such as PubMed’s “publication type” field offer broader coverage (e.g., labeling studies as “Clinical Trial” or “Review”), they lack explicit evidence-hierarchy distinctions (e.g., differentiating high-quality RCTs from lower-quality observational studies) required for direct alignment with the LoE framework (Pasche et al., 2020). Several machine learning-based tools are developed and used for predicting the “publication type” field such as Anne O’Tate, RCT Tagger, Multi-Tagger, etc (Cohen et al., 2021).

No other automation effort to date has explicitly attempted to incorporate the LoE framework, despite its central place in EBM practice. This work’s main contribution is in providing a fully automatic retrieval system for medical publications by automatizing the EBM practice of assigning LoE to medical publications and then using LoE to decide on the relevance of a publication in a given context.

3 LoE Classifier

We view the problem of assigning LoE to medical publications as a classification task and explain in this section the training and the evaluation of the LoE classifier.

3.1 Data

We use a dataset derived from the Oncology Guidelines of the German Association of Scientific Medical Societies (Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften¹). This dataset is unique in that it explicitly mentions the LoE of various medical publications as per the OCEBM framework. It includes 2816 publication–LoE pairs, extracted from unstructured PDFs². The distribution of LoE levels in the dataset is

1. <https://www.awmf.org/>

2. A structured format of the dataset is available on <https://github.com/samehfrihat/LevelOfEvidence>

as follows: 14% in 1a, 18% in 1b, 10% in 2a, 24% in 2b, 12% in 3a, 7% in 3b, and 15% in 4. In Section 4.1, we compare this dataset distribution with the rest of the medical literature.

The Oncology Guidelines mention publications as citations, which include the authors names, publication year, and publication title. This information is not sufficient for automatic LoE classification, which additionally requires some of the methodology, interventions, and clinical outcomes. This information can only be found in publication abstracts or full texts. Therefore, we leverage the PubMed API³ to enrich the initial dataset with abstracts and PubMed IDs.

The average word count in the abstracts is 263 (SD=97), slightly above the typical range for medical articles (Andrade, 2011). The prevalence of longer abstracts can be attributed to the frequent use of structured abstract formats within the medical literature (Hartley, 2004). Notably, we observe a positive correlation between the abstract length and the LoE classification: publications with higher evidence levels tend to have longer abstracts (e.g. LoE 1a with a mean of 325 words (SD=163) than those with lower levels (LoE 3b and 4 with a mean of 233 words (SD=71)).

We split this data into a training dataset containing 1690 instances (60%) and a validation and testing dataset containing 563 instances (20%) each, ensuring a stratified representation across all classes.

3.2 Experimental Setup

For the task of LoE classification, we focus on fine-tuning PubMedBERT (Gu et al., 2021). PubMedBERT is a natural choice for this domain-specific classification task as it is a transformer-based model pre-trained using abstracts sourced directly from PubMed. Its efficacy has been well-established: It currently holds the top score on the Biomedical Language Understanding and Reasoning Benchmark (Gu et al., 2021), it excels in accurately interpreting the unique terminologies and context of biomedical texts, and it is proficient in handling the complexities of biomedical literature. The model is fine-tuned using the training set and hyperparameters are optimized using the validation set. We develop the following classifiers:

Random Forest (RF) RF serves as our baseline. It is trained on the training set for multi-class classification. We use TF-IDF vectorization and chi-squared feature selection, and K-Fold cross-validation using the validation dataset, evaluating its performance with the macro-F1 score.

Multi-Class-PubMedBERT This classifier is directly fine-tuned on the training set to classify texts into specific LoE classes, with the macro-F1 score as the evaluation matrix.

Reg-PubMedBERT This is a regression approach, which assigns numeric values to LoE classes. PubMedBERT is fine-tuned to predict these values, by mapping different LoEs (1a, 1b, 2a, 2b, 3a, 3b, 4) to the numeric values (0, 1, 2, 3, 4, 5, 6). We used root-mean-square error (RMSE) for evaluation. To align the model’s predictions with the original LoE classes and to facilitate comparison with other classifiers using the F1 matrix, we mapped the predicted value to the nearest integer value and then used the same map to get predictions back to their corresponding LoE classes.

3. <https://pubmed.ncbi.nlm.nih.gov/>

Multi-Label-PubMedBERT This classifier incorporates the multi-label classification approach, i.e. we transform the LoE categorization into a set of binary labels. Each label corresponds to a specific LoE class, effectively converting the problem into a multi-label classification task. This version enabled PubMedBERT to predict multiple labels simultaneously, accommodating the scenario where only one of the labels should be true while others are false. By modelling the LoE classification as a multi-label task, we aim to capture potential overlap between LoE classes and assess the model’s capacity to handle such nuances by looking at the prediction list that might contain multiple levels of evidence. For proper evaluation, we assigned the highest confidence value when multiple positive predictions.

Ensemble Majority Vote Ensemble methods are a well-established technique in classification that capitalizes on the strengths of diverse classifiers to enhance prediction accuracy and generalization (Polikar, 2012). We employed an Ensemble Majority Vote strategy, combining the strengths of the three PubMedBERT models (Multi-Class, Reg, and Multi-Label). This approach used majority voting to aggregate predictions from each model, enhancing the overall classification accuracy and robustness (Zhou and Zhou, 2021; Dang et al., 2020).

3.3 Classifier Evaluation

We evaluate our LoE document classifiers using Macro F1 score, RMSE, and Confusion matrices. RMSE treats the LoE classification as a regression task by mapping each LoE category to a corresponding integer. The RMSE score reflects the average squared difference between the predicted and true LoE values, providing important insight into how closely the model captures the ordinal nature of the LoE hierarchy. Lower RMSE values indicate that the model’s predictions are closer to the true LoE, particularly emphasizing the reduced impact of misclassifications to adjacent levels.

3.3.1 INDIVIDUAL CLASSIFIERS PERFORMANCE

Table 1 summarizes the performance of each classifier on the test dataset.

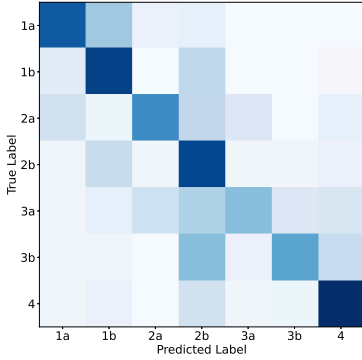
Model	F1 score	RMSE
Random Forest (RF)	0.59	1.30
Multi-Class-PubMedBERT	0.78	0.90
Reg-PubMedBERT	0.74	0.69
Multi-Label-PubMedBERT	0.79	0.90 [*]
Majority voting	0.83	0.65

Table 1: Level of Evidence Classifiers Performance on our test set. Macro F1 Score.

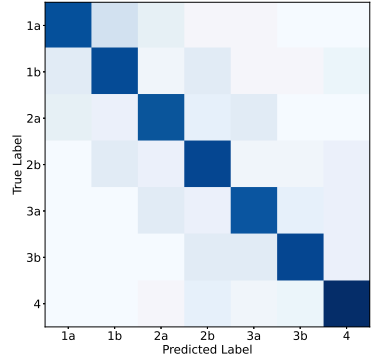
^{*} By considering the label of the highest confidence score as predicted class

RF Baseline The RF model’s performance with a macro-F1 score of 0.59 and an RMSE of 1.30 did not surpass the deep learning models’ results. Nevertheless, the RF model shows robustness in effectively handling the challenges of multi-class LoE classification. Analyzing the confusion matrix in Figure 1a, we see that the misclassifications are rather scattered, they are not clustered in any particular class or in neighboring classes.

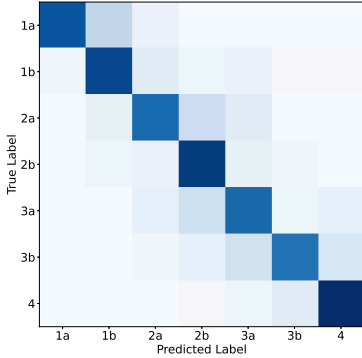
Multi-Class-PubMedBERT Multi-Class-PubMedBERT scored 0.78 in F1 (+0.19 compared with baseline) and 0.90 in RMSE, showing effectiveness in multi-class categorization. However, after we analysed misclassification in Figure 1b, we found that the model has some difficulties distinguishing closely related LoE classes. This suggests considering the problem as a regression task, since misclassification with neighbor classes is not as bad as assigning distant classes such as replacing 1a with 4.



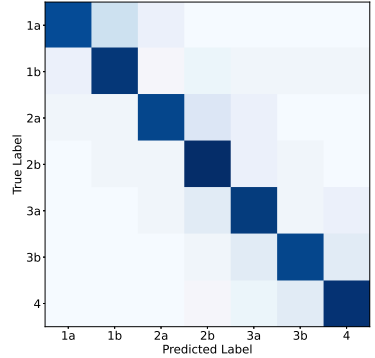
(a) Random Forest as a baseline



(b) Multi-Class-PubMedBERT



(c) Reg-PubMedBERT



(d) Majority voting

Figure 1: Confusion Matrices using the test set per model.

Reg-PubMedBERT exhibited strengths in capturing the ordered nature of LoE with an F1 score of 0.74 and the second-best RMSE of 0.69, indicating proficiency in differentiating between levels. This makes misclassified documents closer to the true labels, which is reflected in the smaller RMSE and highlighted in Figure 1c. This makes misclassification to neighboring classes less harmful than assigning far classes.

Multi-Label-PubMedBERT The model performed best among individual classifiers with an F1 score of 0.79, adeptly handling documents with multiple LoE categories. A closer qualitative examination of this model’s performance revealed that some documents were assigned into multiple LoE classes. This is a well known phenomenon, which was explored in the work of Murad et al. (2016) bringing into question the clear demarcation between the evidence levels of the EBM pyramid. Instead, a nuanced perspective on LoEs has been proposed to align with the flexibility of multi-label classification as demonstrated by Multi-Label-PubMedBERT.

3.3.2 ENSEMBLE MAJORITY VOTE PERFORMANCE

The Ensemble Majority Vote method combines the predictions of all three PubMedBERT models and demonstrates the best performance. It scores highest in F1 (0.83) and achieves an RMSE of 0.65, indicating its effectiveness in accurately categorizing medical literature by LoE. This result emphasizes the significant role of collaborative intelligence in enhancing classification outcomes. It also benefited from the power of the regression model, where misclassification resulted in neighboring classes as shown in Figure 1d and the smallest RMSE.

3.3.3 STATISTICAL SIGNIFICANCE ANALYSIS

We performed a statistical significance analysis on our machine learning models using a paired t-test. After applying Bonferroni correction ($\alpha = 0.05/10$), we found that all deep learning models significantly outperformed the Random Forest baseline, indicating their effectiveness in LoE classification. However, no significant performance differences were observed among the deep learning models themselves, highlighting their comparable efficacy in evidence-based classification.

3.3.4 IDENTIFYING SIGNIFICANT TERMS

We utilized the LIME (Local Interpretable Model-Agnostic Explanations) explainer (Ribeiro et al., 2016) to identify key terms influencing our model’s predictions for different Levels of Evidence (LoE) categories. This method provides insights by aggregating term scores, helping us to determine significant terms for each LoE level. Such an approach enhanced the interpretability and transparency of our model, highlighting LoE-specific terms in the analyzed documents.

Table 2 presents the top 10 contributing terms across the LoE levels in the test set. The results highlighted that our model was able to identify discriminating terms for each class. Moreover, we discovered common terms shared across multiple levels, such as “systematic review” in 1a (systematic reviews of RCTs), 2a (systematic reviews of cohort studies), and 3a (systematic reviews of case-control studies), and “RCT” in 1a and 1b (individual RCTs). Additionally, some less expected terms, like “risk” in 2a, 2b (individual cohort studies), 3a, and 3b (individual case-control studies), and “accuracy study” in 1a, 2a, and 3a (pertaining to Diagnostic Test Accuracy studies), emerged as significant classifiers. Interestingly, a specific therapy (“acupuncture”) only occurs among the terms of level 4, possibly indicating the lack of stronger evidence for this method.

1a		1b		2a		2b	
term	score	term	score	term	score	term	score
accur predict	2.11	achiev complet	1.92	cohort studi	1.30	cohort studi	1.62
accur stage	1.85	achiev patient	1.91	accuraci detect	1.14	accrual	1.42
accuraci respect	1.72	activ control	1.58	systemat review	1.09	acquisit	1.14
rct	1.42	activ intervent	1.56	meta analysi	1.02	accept	1.11
meta analysi	1.31	activ surveil	1.25	exposur	0.98	access	1.08
systemat review	1.30	rct	1.21	longitudin	0.95	accru	1.01
accuraci studi	1.17	control set	1.12	access	0.74	longitudin	0.89
accuraci clinic	1.16	acut delay	0.98	accur stage	0.73	risk	0.61
achiev	1.15	acut	0.79	accuraci studi	0.64	administr	0.21
activ treatment	1.02	adjuv	0.71	risk	0.59	affect patient	0.14

3a		3b		4	
term	score	term	score	term	score
systemat review	1.24	case control	1.60	small sampl	1.69
epidemiolog	1.21	case definit	1.41	preliminari evid	1.32
case definit	1.17	exposur	1.02	exploratori research	0.99
abnorm	1.12	risk	0.49	uncontrol studi	0.98
exposur	1.11	advers reaction	0.31	acupunctur treatment	0.68
absent	0.98	affect patient	0.30	patient characterist	0.60
accuraci respect	0.88	age	0.29	acupunctur effect	0.51
accuraci studi	0.71	age diagnosi	0.23	analisi reveal	0.22
risk	0.64	advers effect	0.19	analisi identifi	0.22
accur stage	0.51	affect surviv	0.10	affect	0.13

Table 2: Significant Terms in the Level of Evidence Classifier.

4 Levels of Evidence as a filter in medical IR

For the retrieval experiments, the 7-class LoE model was simplified into a 4-class setup by grouping related evidence levels. This decision reflects real-world usage patterns where users often prioritize broader evidence categories, such as high-quality studies (e.g., systematic reviews and RCTs) or intermediate-level evidence (e.g., cohort and case-control studies). This reduction not only simplifies classification, but also improves retrieval effectiveness without compromising performance.

In this experiment, we investigate the benefit of LoE classification for the IR of medical publications using TREC Precision Medicine (PM) datasets from 2017 to 2019 (Roberts et al., 2017, 2018, 2019).

4.1 Data

The TREC PM datasets, sourced from the Medline collection⁴, consist of over 26 million research article abstracts accessible via PubMed and designed to enhance biomedical IR. Topics/queries were constructed based on disease and gene fields from the dataset, omitting demographic data to focus specifically on abstract retrieval. Relevance judgements were

4. https://www.nlm.nih.gov/medline/medline_overview.html

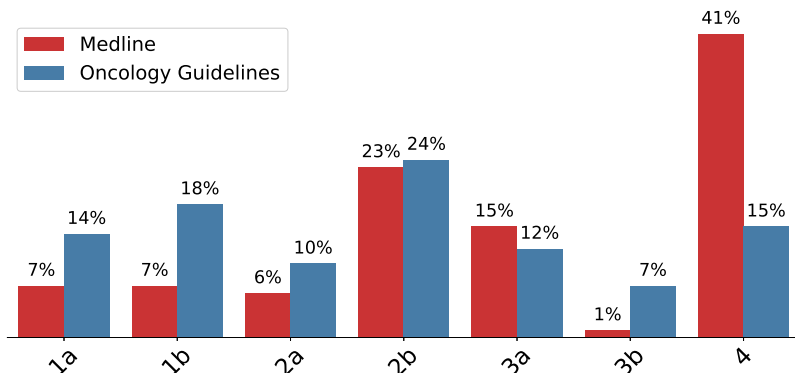


Figure 2: The distribution of LoE Classes in the Medline Dataset and Oncology Guidelines (Classifier Dataset).

performed by expert assessors on a scale of ‘not relevant (0)’, ‘partially relevant (1)’, and ‘definitely relevant (2)’, based on alignment with a given topics (Roberts et al., 2017). The criteria for relevance did not include the LoE of the documents.

We categorize each abstract in the Medline collection into its respective LoE category using our ensemble classifier. Figure 2 shows the distribution of LoE classes in Medline data. Most frequent are Level 4 documents (41% of the collection), which require the smallest empirical basis. The highest LoE 1a and 1b each represent only 7% of the documents. This unbalanced distribution reflects the inherent nature of the biomedical literature, where expert opinion and hypothesis-generating studies far outnumber high-evidence clinical research, which is usually conducted after several studies have confirmed the same observations.

In contrast, the oncology guidelines exhibit a reversed pattern, with only 15% of low evidence documents, suggesting a higher balance ratio. This difference is due to the selection process during the formulation of new clinical guidelines, where publications with higher evidence are prioritized.

4.2 Experimental Setup

In this subsection, we present the IR methods used for performing this task and the evaluation metric. As a core retrieval algorithm, we use BM25 (Robertson et al., 2009), which is widely used in IR for scoring and ranking documents based on their relevance to a user query. It is a probabilistic-based approach that builds on the classic TF-IDF (Term Frequency-Inverse Document Frequency) model, refining it by incorporating factors like term frequency saturation and document length normalization. The algorithm calculates a score for each document by considering how frequently the query terms appear in the document, adjusting for the overall document length and the rarity of the terms across the entire corpus. BM25 is particularly valued for its ability to effectively balance term frequency and inverse

document frequency, making it one of the most robust and popular methods for ranking search results.

In the retrieval experiment, we first indexed the entire Medline collection (Sec. 4.1) alongside their assigned LoE classes. The BM25 algorithm, parameterized with $K_1 = 1.2$ and $b = 0.75$ as recommended in (Connelly, 2019), was then applied to retrieve and rank documents based exclusively on textual relevance (abstracts and titles) to the query, without integrating LoE into the ranking process. This methodology ensures a fair comparison with the baseline.

4.2.1 RETRIEVAL METHODS

Our experiment utilizes the BM25 retrieval method applied to documents of all LoE classes (‘All’) as a baseline for our IR process. The impact of LoE classification is tested by filtering the documents based on their LoE as follows:

- *LoE3+*: LoE categories 3b to 1a, i.e. case-control studies or higher LoE.
- *LoE2+*: LoE categories 2b to 1a, i.e. cohort studies or higher LoE.
- *LoE1*: LoE categories 1a and 1b, i.e. RCTs only.

4.2.2 EVALUATION METRIC

The performance of each model’s effectiveness was assessed using infNDCG , R-Prec , and P@10 matrices, as these are the official matrices used to report on the datasets. Also, we report the “Normalized discounted cumulative gain @10” (NDCG@10) metric as our core metric (Järvelin and Kekäläinen, 2000). This measure allows for considering relevance grades $0 \dots n$, where, in our case, irrelevant documents receive a score of 0, while partially relevant and definitely relevant documents receive scores of 1 and 2, respectively. For a ranked document list, let r_j denote the relevance grade of the document at rank j . Then the (unnormalized) discounted cumulative gain for a ranked list of length k is defined as:

$$DCG(k) = \sum_{j=1}^k \frac{r_j}{\max(1, \log_b(j))}.$$

With the denominator in the summation elements, DCG simulates a stochastic user stopping behavior, where not every user checks all documents up to the final rank k , but some users might stop at earlier ranks; the fraction of users reaching a certain rank is controlled by the logarithm base b (usually chosen as $b = 2$). As the DCG values for a query depend heavily on the number of relevant documents in the collection, they are normalized by comparing them with the value $DCG_{opt}(k)$ of the optimum retrieval result (i.e. ranking documents by decreasing relevance grades), thus arriving at the normalized discounted cumulative gain:

$$NDCG@k = DCG(k) / DCG_{opt}(k).$$

Besides incorporating a fairly realistic user stopping behavior and being one of the few retrieval metrics considering different relevance grades, NDCG also has a nice theoretic property: (Ferrante et al., 2021) showed that NDCG comes closest to an interval scale

(which is a requirement for computing means and effect sizes), while other popular measures with stochastic stopping behavior (like average precision or rank-biased precision) clearly violate this property.

4.3 Results

As shown Table 3 using LoE to filter out document set to be searched improves the retrieval effectiveness as measured by NDCG@10 score. The retrieval of RCT documents with highest LoEs is the most successful. Moreover, there is a clear trend in improving NDCG when the minimum LoE is increased. For all three collections, the strictest filter (LoE1 with only 14% of the collection) outperformed all other methods, with substantial NDCG improvements (0.08 ... 0.11) over the baseline. As we are re-using a test collection, performing statistical tests here would contradict statistical testing theory (Fuhr, 2017). Instead, we give the effect sizes, which indicate substantial improvements over the baseline.

Moreover, as shown in Table 4, our LoE1 model improved the performance of the baseline on all matrices. It also outperformed each of the best-reported runs on infNDCG matrix and provided comparable results on R-Prec⁵. In addition, the retrieval quality of our method is accompanied with the guarantee of returning only documents of the highest evidence. On the other hand, as the results for P@10 show, LoE seems to be too strict when the user is looking at all 10 top-ranking documents.

Exp./Year	size*	2017	2018	2019
<i>All</i>	100%	0.46	0.59	0.54
<i>LoE3+</i>	59%	0.48 (0.02)**	0.60 (0.01)	0.57 (0.03)
<i>LoE2+</i>	43%	0.49 (0.03)	0.64 (0.05)	0.58 (0.04)
<i>LoE1</i>	14%	0.54 (0.08)	0.69 (0.10)	0.65 (0.11)

Table 3: Models’ NDCG@10 performance on TREC PM datasets

* Size denotes the percentage of the collection that was considered in retrieval.

** Numbers in parentheses show the effect size when comparing with the baseline “All”.

Exp./Year	2017	2018	2019
<i>All</i>	0.43 / 0.27 / 0.52	0.50 / 0.32 / 0.58	0.47 / 0.30 / 0.57
<i>LoE3+</i>	0.45 / 0.28 / 0.54	0.52 / 0.34 / 0.60	0.50 / 0.31 / 0.58
<i>LoE2+</i>	0.47 / 0.28 / 0.54	0.55 / 0.36 / 0.61	0.52 / 0.31 / 0.61
<i>LoE1</i>	0.52 / 0.30 / 0.55	0.57 / 0.38 / 0.61	0.58 / 0.34 / 0.61
<i>Top run</i>	0.46 / 0.30 / 0.64	0.56 / 0.37 / 0.71	0.58 / 0.36 / 0.65

Table 4: Models’ InfNDCG/R-Prec/P@10 performance on TREC PM datasets.*

* Best reported runs per matrix, meaning the model performing best on P@10 is not the same as the model performing best on infNDCG.

5. Note that these are pessimistic estimates, as unjudged documents only retrieved by our method are treated as irrelevant

5 Discussion

In this paper, we have effectively demonstrated the automated application of the LoE framework for improving the retrieval of relevant medical publications. Our approach, leveraging fine-tuned PubMedBERT models, has proven adept at classifying medical publications based on their LoE with a high degree of accuracy (macro F1 = 0.83). This advancement addresses a significant gap in existing literature, where previous studies have largely focused on specific evidence levels, particularly RCTs and their systematic reviews. The higher transparency of our approach gives users full control over the LoE of the documents returned. Moreover, the method investigated here could be directly integrated into the existing PubMed search engine, by simply adding estimated LoE as an additional document attribute that can be referred to in the query.

A key finding of our work is the effect of LoE filtering in directing attention towards the most reliable 14% of documents, while enhancing retrieval quality at the same time. This aspect is particularly crucial in the medical domain, where accessing accurate and high-quality information rapidly can make a pivotal difference in patient care and medical research. On the other hand, LoE2 or LoE3 papers may also be searched for in case there are no relevant answers in the top level, e.g. when the user is interested in more recent methods for which higher level studies are not available yet. Therefore, we acknowledge that clinical decision-making often requires synthesizing multiple sources across different LoE levels.

In our study, the LoE1 model outperformed the best-reported runs on the three datasets (Roberts et al., 2017, 2018, 2019) in terms of infNDCG and provided comparable results in R-Prec matrix. This demonstrated the effectiveness of using LoE as a filter in medical IR, improving the relevance and reliability of retrieved documents. These improvements over integrating the LoE filter in the BM25 baseline suggest that these benefits could extend to the other stronger baselines.

Although our study shows the potential of using LoE in Medline, one limitation that needs to be considered is the potential bias from using the oncology guideline dataset for training the classifiers. Medline collection contains publications where LoE can not be applied, such as bioinformatics. To apply it in real-world applications, we could introduce a new class, "others", where the model confidence score is below the seine threshold or when multiple positive labels are in the multi-label classifier.

Moverover, the LoE framework prioritizes study design rigor but does not assess study quality (e.g., risk of bias). Future work should integrate tools like GRADE (Guyatt, 2009) or Cochrane’s risk of bias assessment to enhance reliability. This requires expanding the research article and analyzing the full text rather than the title and abstract, which are enough for assigning LoE.

In our recently published user study (Frihat et al., 2024), we present findings from an evaluation with medical professionals testing a clinical search engine that integrates LoE classification with biomedical concepts as a semantic layer (Frihat and Fuhr, 2025).

The results demonstrated strong user engagement with LoE: 93% of participants reported prior familiarity with LoE frameworks, and 85% actively filtered search results based on high LoE levels, noting that this feature facilitated their ability to prioritize high-quality

evidence. Their feedback also highlighted the added value of biomedical concept extraction (e.g., gene-disease relationships) in contextualizing evidence.

6 Conclusion

Our research addresses the challenge faced by current search engines in identifying significant, evidence-backed medical publications. Although relevant and widely used in evidence-based medical practice, the LoE framework has not yet been fully automatised and tested for medical IR. We introduce a classification model for tagging medical research abstracts with LoE levels and demonstrate that a vast number of medical publications without LoE tags can be successfully and fully automatically enriched with this crucial information. Our retrieval results confirm that LoE is an effective filter that improves results in a fully automatic retrieval scenario. These results suggest that our LoE based approach to medical IR is a viable and robust tool to evidence-based medical practice, which can facilitate and improve medical decision-making, leading to better patient care. However, effective decision-making often requires synthesizing multiple studies and integrating clinical practice guidelines, which remains an important area for future work.

Acknowledgments and Disclosure of Funding

This work was funded by a PhD grant from the DFG Research Training Group 2535 Knowledge- and data-based personalization of medicine at the point of care (WisPerMed), University of Duisburg-Essen, Germany. We also acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

References

- Chittaranjan Andrade. How to write a good abstract for a scientific paper or conference presentation. *Indian Journal of Psychiatry*, 53(2):172, 2011.
- Anneliese Arno, James Thomas, Byron Wallace, Iain J Marshall, Joanne E McKenzie, and Julian H Elliott. Accuracy and efficiency of machine learning-assisted risk-of-bias assessments in “real-world” systematic reviews: A noninferiority randomized controlled trial. *Annals of Internal Medicine*, 175(7):1001–1009, 2022.
- Kristy M Borawski, Regina D Norris, Susan F Fesperman, Johannes Vieweg, Glenn M Preminger, and Philipp Dahm. Levels of evidence in the urological literature. *The Journal of Urology*, 178(4):1429–1433, 2007.
- Patricia B Burns, Rod J Rohrich, and Kevin C.Chung. The levels of evidence and their role in evidence-based medicine. *Plastic and Reconstructive Surgery*, 128(1):305, 2011.
- Aaron M Cohen, Jodi Schneider, Yuanxi Fu, Marian S McDonagh, Prerna Das, Arthur W Holt, and Neil R Smalheiser. Fifty ways to tag your pubtypes: Multi-tagger, a set of probabilistic publication type and study design taggers to support biomedical indexing and evidence-based medicine. *medRxiv*, pages 2021–07, 2021.

- Shane Connelly. Practical BM25 – Part 3: Considerations for picking b and k1 in Elasticsearch, 2019. URL <https://www.elastic.co/blog/practical-bm25-part-3-considerations-for-picking-b-and-k1-in-elasticsearch>.
- Huong Dang, Kahyun Lee, Sam Henry, and Özlem Uzuner. Ensemble BERT for classifying medication-mentioning tweets. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 37–41, Barcelona, Spain, 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.smm4h-1.5>.
- Vishal S Desai, Christopher L Camp, and Aaron J Krych. What is the hierarchy of clinical evidence? *Basic Methods Handbook for Clinical Orthopaedic Research: A Practical Guide and Case Based Research Approach*, Springer, pages 11–22, 2019.
- Marco Ferrante, Nicola Ferro, and Norbert Fuhr. Towards meaningful statements in IR evaluation: Mapping evaluation measures to interval scales. *IEEE Access*, 9:136182–136216, 2021. doi: 10.1109/ACCESS.2021.3116857.
- Sameh Frihat and Norbert Fuhr. Integration of biomedical concepts for enhanced medical literature retrieval. *International Journal of Data Science and Analytics*, pages 1–24, 2025.
- Sameh Frihat, Papernmeier, and Norbert Fuhr. Enhancing biomedical literature retrieval with level of evidence and bio-concepts: A comparative user study. The ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2024.
- Norbert Fuhr. Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum*, 51(3):0 32–41, 2017. URL <http://sigir.org/wp-content/uploads/2018/01/p032.pdf>.
- Evidence-Based Medicine Working Group, Gordon Guyatt, Drummond Rennie, et al. *Users’ guides to the medical literature: a manual for evidence-based clinical practice*. AMA Press, 2002.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.
- GH Guyatt. Grade: an emerging consensus on rating quality of evidence and strength of recommendations. *Chinese Journal of Evidence-Based Medine*, 9:8, 2009.
- James Hartley. Current findings from research on structured abstracts. *Journal of the Medical Library Association*, 92(3):368, 2004.
- Lisa Hartling and Allison Gates. Friend or foe? the role of robots in systematic reviews. *Annals of Internal Medicine*, 175(7):1045–1046, 2022.
- Julian Hirt, Jasmin Meichlinger, Petra Schumacher, and Gerhard Mueller. Agreement in risk of bias assessment between robotreviewer and human reviewers: An evaluation study

on randomised controlled trials in nursing-related cochrane reviews. *Journal of Nursing Scholarship*, 53(2):246–254, 2021.

Jeremy Howick. The Oxford 2011 levels of evidence. *Centre for Evidence-Based Medicine*, 2011. URL <https://www.cebm.ox.ac.uk/resources/levels-of-evidence/ocebml-evels-of-evidence>

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In Emmanuel J. Yannakoudakis, Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong, editors, *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 41–48, ACM, 2000. doi: 10.1145/345508.345545.

Gabriel M Leung. Evidence-based practice revisited. *Asia Pacific Journal of Public Health*, 13(2):116–121, 2001.

Iain J Marshall and Byron C Wallace. Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. *Systematic reviews*, 8:1–10, 2019.

Iain J Marshall, Joël Kuiper, and Byron C Wallace. Automating risk of bias assessment for clinical trials. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 88–95, 2014.

Iain J Marshall, Joël Kuiper, and Byron C Wallace. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, 23(1):193–201, 2016.

M Hassan Murad, Noor Asi, Mouaz Alsawas, and Fares Alahdab. New evidence pyramid. *BMJ Evidence-Based Medicine*, 21(4):125–127, 2016.

Emilie Pasche, Déborah Caucheteur, Luc Mottin, Anaïs Mottaz, Julien Gobeill, and Patrick Ruch. SIB text mining at TREC precision medicine 2020. In *Proceedings of the 29th Text REtrieval Conference (TREC 2020)*. 16–20, 2020.

Timothy B Patrick, George Demiris, Lillian C Folk, David E Moxley, Joyce A Mitchell, and Donghua Tao. Evidence-based retrieval in evidence-based medicine. *Journal of the Medical Library Association*, 92(2):196, 2004.

Robi Polikar. Ensemble learning. *Ensemble machine learning: Methods and applications*, pages 1–34, 2012.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why should I trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, Alexander J Lazar, and Shubham Pant. Overview of the TREC 2017 precision medicine track. In *Proceedings of the 28th Text REtrieval Conference (TREC)*, 2017.

- Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, and Alexander J Lazar. Overview of the TREC 2018 precision medicine track. In *Proceedings of the 29th Text REtrieval Conference (TREC)*, 2018.
- Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, Alexander J Lazar, Shubham Pant, and Funda Meric-Bernstam. Overview of the TREC 2019 precision medicine track. In *Proceedings of the 30th Text REtrieval Conference (TREC)*, 2019.
- Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
- Anthony L Rosner. Evidence-based medicine: revisiting the pyramid of priorities. *Journal of Bodywork and Movement Therapies*, 16(1):42–49, 2012.
- Brijesh Sathian, Jayadevan Sreedharan, Suresh N Baboo, Krishna Sharan, ES Abhilash, and E Rajesh. Relevance of sample size determination in medical research. *Nepal Journal of Epidemiology*, 1(1):4–10, 2010.
- Frank Soboczenski, Thomas A Trikalinos, Joël Kuiper, Randolph G Bias, Byron C Wallace, and Iain J Marshall. Machine learning to help researchers evaluate biases in clinical trials: a prospective, randomized user study. *BMC Medical Informatics and Decision Making*, 19:1–12, 2019.
- Katya Strage, Stephen Stacey, Cyril Mauffrey, and Joshua A Parry. The interobserver reliability of clinical relevance in medical research. *Injury*, 54:S66–S68, 2023.
- Peter Van de Vliet, Tobias Sprenger, Linde FC Kampers, Jennifer Makalowski, Volker Schirmacher, Wilfried Stücker, and Stefaan W Van Gool. The application of evidence-based medicine in individualized medicine. *Biomedicines*, 11(7):1793, 2023.
- Zhi-Hua Zhou and Zhi-Hua Zhou. *Ensemble learning*. Springer, 2021.

Annotative Indexing

Charles L. A. Clarke

*University of Waterloo
Canada*

CLACLARK@GMAIL.COM

Editor: Ismail Sengor Altingovde

Abstract

This paper introduces annotative indexing, a novel framework that unifies and generalizes traditional inverted indexes, column stores, object stores, and graph databases. As a result, annotative indexing can provide the underlying indexing framework for databases that support retrieval augmented generation, knowledge graphs, entity retrieval, semi-structured data, and ranked retrieval. While we primarily focus on human language data in the form of text, annotative indexing is sufficiently general to support a range of other data types, and we provide examples of SQL-like queries over a JSON store that includes numbers and dates. Taking advantage of the flexibility of annotative indexing, we also demonstrate a fully dynamic annotative index incorporating support for ACID properties of transactions with hundreds of multiple concurrent readers and writers.

Keywords: Search, Indexing, Inverted Indexes, Minimal-interval Semantics

1 Introduction

Until recently, and with few exceptions, an inverted index provided the foundational file structure for an information retrieval system. Over the years, research progress on file structures for information retrieval was primarily driven by the need to make traditional first-stage sparse retrieval methods (e.g., BM25) as fast as possible, while minimizing storage and memory requirements, motivating the development of specialized processing methods (e.g., WAND) and compression methods (e.g., vByte). To a large extent, this research views an inverted index as single-purpose file structure, with the sole task of delivering the top- k items from a large collection to a second-stage re-ranker with high throughput and low latency. More recently, vector databases supporting dense retrieval have begun to replace inverted indexes, but the focus remains on the efficiency and effectiveness of first-stage retrieval.

Managing large collections of human language data requires more than just a single-minded focus on first-stage retrieval. For example, guidelines for the TREC 2024 RAG Track¹ describe the preparation of a segmented version of the MS MACRO V2 passage corpus for use by track participants. Processing steps include the identification and elimination of duplicate passages to avoid holes and inconsistencies in evaluation. The original corpus was segmented with “a sliding window size of 10 sentences and a stride of 5 sentences” to make it “more manageable for users and baselines.” The original corpus and

1. <https://trec-rag.github.io/>

its de-duplicated/segmented version are distributed as two independent sets of compressed JSONL files, linked to each other only by a naming convention for document identifiers.

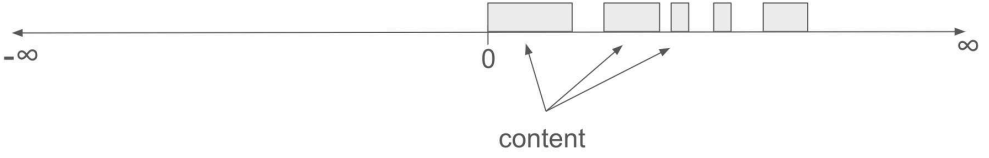
In general, collections of human language data employ a variety of text formats, including JSON, JSONL, TSV, CSV, HTML, CBOR, LaTeX, Word, and PDF. Even source code, such as Python and C++, can be considered as a form of human language data. Processing text collections involves transformations such as tokenization, sentence/word splitting, de-duplication, tagging, and entity linking, as well as generating and storing weights for sparse retrieval and vectors for dense retrieval. Tools for these tasks range from record-at-a-time processing in notebooks to storage in a variety of database systems, including relational databases, search engines, object stores, and knowledge graphs. No single tool allows us to flexibly store, transform, and search multi-format heterogeneous collections of unstructured and semi-structured human language data.

This paper introduces *annotative indexing*, a novel framework that unifies and generalizes traditional inverted indexes, column stores, object stores, and graph databases. As a result, annotative indexing can provide the underlying indexing framework for databases that support retrieval augmented generation, knowledge graphs, entity retrieval, semi-structured data, and ranked retrieval. While we primarily focus on human language data in the form of text, annotative indexing is sufficiently general to support a range of other data types. Annotative indexing facilitates dynamic update, which in turn facilitates text processing pipelines that perform de-duplication, segmentation, and similar operations, expressing these operations by annotating the source text, rather than generating new text.

The next section (Section 2) presents the fundamentals of annotative indexing, providing a foundation for the remainder of the paper. Section 3 places annotative indexing in the context of prior work. Section 4 then presents the overall organization of an annotative index. As a proof of concept, the section also describes the architecture of our reference implementation, called Cottontail². All experimental results in the paper were generated with this reference implementation. The design of the reference implementation reflects the relative simplicity of an annotative index, with a small number of generic components that can be specialized and combined to support different applications.

Section 5 discusses query processing, including an example of a JSON store built on Cottontail, which supports structural containment, Boolean expressions, and similar operations, along with numbers and unified support for dates in differing formats. Section 6 discusses support for dynamic update and transactions, including support for ACID properties. As an example, Section 6 presents a dynamically evolving collection that recapitulates the early years of TREC experiments, with dozens of concurrent writers and hundreds of concurrent readers. Annotative indexing and dynamic update complement and support each other. Dynamic update of traditional inverted indexes is generally limited to adding and deleting entire documents, with no or limited support for concurrent update and transactions. Annotative indexing provides the ability to annotate content after it has been added, enabling richer and more flexible update operations — which, in turn, requires transactional support to ensure concurrency among multiple readers and writers.

2. Code for the reference implementation is available at <https://github.com/claclark/Cottontail>. Following past practice in the information retrieval community, the reference implementation is named after an animal, in this case the eastern cottontail, which is the most common species of rabbit in North America. The author often encounters them out and about near the University of Waterloo.



$\mathcal{X}(592856130, 592856138) \Rightarrow$ To be or not to be, | that is the

$\mathcal{X}(17905274055, 17905274393) \Rightarrow$

```
{ "docid": "msmarco_v2.1_doc_29_677149#3_1637632" , "end_char": 3061 ,
  "headings": "Aeolian Vibration of Transmission Conductors Aeolian
  Vibration of Transmission Conductors What is Aeolian Vibration?
  Wind causes a variety of motions on transmission line conductors.
  Important among them are How Aeolian Vibration Occurs? Theory/Mechanism...
  ...that creates an alternating pressure imbalance causing the conductor to
  move up and down at a ninety-degree angle to the flow direction." ,
  "start_char": 1806 , "title": "Aeolian Vibration of Transmission Conductors",
  "url": "https://studyelectrical.com/2019/07/aeolian-vibration-..." }
```

Figure 1: The content of an associative index is situated in an address space, which may contain gaps, where content has been deleted. A translation function $\mathcal{X}(p, q)$ maps an interval in the address space to its associated content. The figure shows examples from an index containing the segmented version of the MS MARCO V2.1 Document Corpus as used by the TREC 2023 RAG Track. In this example, tokenization is word based. At the content level, a JSON object is represented as a sequence of tokens, with special tokens representing JSON structural tokens (" , : , etc.). Annotations on top of the content define structural elements. For example, the annotation $\langle \text{:title:}, (17905274368, 17905274374) \rangle$ indicates the interval containing the title in this object.

2 Fundamentals of Annotative Indexing

An annotative index stores human language data as its *content* plus a set of *annotations* describing that content. The content is represented by a sequence of *tokens*, where each token is assigned an integer location in an *address space*, as illustrated in Figure 1. If content has been deleted, gaps are possible. By convention, our reference implementation appends content at increasing addresses, starting at zero. However, negative addresses are supported for mathematical simplicity and consistency (see example below). As shown in Figure 1, a translation function $\mathcal{X}(p, q)$ maps an interval in the address space to the associated content. $\mathcal{X}(p, q)$ is undefined if (p, q) contains a gap. For content addressing purposes, tokenization can be flexibly defined at the word or character level. Separate and distinct tokenization and stemming can be also employed for specific applications, e.g. ranking.

Annotations provide information about intervals over the content. An annotation is a triple $\langle f, (p, q), v \rangle$, where f is a *feature*, (p, q) is the interval over which the annotation applies, and v is the value of the feature over that interval, which defaults to 0. For

convenience we define:

$$\langle f, p, v \rangle = \langle f, (p, p), v \rangle \quad (1)$$

$$\langle f, (p, q) \rangle = \langle f, (p, q), 0 \rangle \quad (2)$$

$$\langle f, p \rangle = \langle f, (p, p), 0 \rangle \quad (3)$$

For example, the annotation:

$$\langle \text{start_char:}, (17905274359, 17905274362), 1806 \rangle$$

indicates that over the interval $(17905274359, 17905274362)$ the feature `start_char:` has the value 1806, as shown in Figure 1. The annotation:

$$\langle \text{tf:porter:aeolian}, 17905274055, 17 \rangle$$

indicates that the Porter-stemmed term “aeolian” appears 17 times in the JSON object starting at address 17905274055. The annotation:

$$\langle :, (17905274055, 17905274393) \rangle$$

indicates that the interval $(17905274055, 17905274393)$ contains a JSON object, as represented by the feature “:”. The annotation:

$$\langle \text{aeolian}, 17905274369 \rangle$$

indicates that the word “aeolian” appears at that address. We can use annotations like these to implement BM25 ranking on a JSON store, but the annotative index itself merely stores the content and its associated annotations. The interpretation of the annotations as term and document statistics is left to the ranking algorithm.

Annotations are indexed by feature, with two access methods (τ and ρ) that both take an address k in the address space and return the first annotation greater than or equal to k , according to the start or end address of the interval.

$$f.\tau(k) = \{ \langle (p, q), v \rangle, \text{ where } \langle f, (p, q), v \rangle \text{ is the annotation for } f \text{ with minimal } p \geq k \} \quad (4)$$

$$f.\rho(k) = \{ \langle (p, q), v \rangle, \text{ where } \langle f, (p, q), v \rangle \text{ is the annotation for } f \text{ with minimal } q \geq k \} \quad (5)$$

To simplify index organization and facilitate index processing, the set of annotations for a feature must follow *minimal-interval semantics* as defined in prior work, including Boldi and Vigna (2016, 2018); Clarke and Cormack (2000), Clarke (1996), and Clarke et al. (1995a). Minimal interval semantics requires that no annotation for the same feature can be contained in another, but they can overlap. If $\langle f, (p, q), v \rangle$ and $\langle f, (p', q'), v' \rangle$ are annotations for feature f , then either $p < p'$ and $q < q'$, or $p > p'$ and $q > q'$. The annotations for f are thus totally ordered — in the same order — by their start and end addresses. For mathematical simplicity and consistency, we consider every feature f to have the annotations $\langle f, (-\infty, -\infty), 0 \rangle$ and $\langle f, (\infty, \infty), 0 \rangle$. Boldi and Vigna (2018) describe a set of intervals under minimal-interval semantics as an element of a “Clarke–Cormack–Burkowski lattice”. Clarke, Cormack, and Burkowski (1995) themselves call it a “generalized concordance list”. In this paper, the term “annotation list” implies an ordered set of annotations under minimal-interval semantics.

Prior work on minimal-interval semantics demonstrated their practical value as a method for expressing queries over heterogeneous collections of semi-structured data, providing efficient support for containment, boolean, merge, proximity, ordering, and other structural operators. This paper extends this prior work in two ways, which together substantially increase expressive power. First, while prior work treated singleton intervals as the only atomic unit for indexing purposes, we index intervals of any length. For example, by indexing intervals we can run a sentence splitter over the content and add annotations to the index indicating sentence boundaries. Second, we associate a value with each interval, which is preserved by containment and merge operations. For example, we can compute terms statistics over the content and add annotations to support ranked retrieval. In this work, the atomic unit for indexing is an annotation, comprising a feature, an interval and a value. Operators combine annotations lists to produce annotation lists.

At its core, an annotative index is just a set of features with the τ and ρ methods of Equations 4 and 5, restricted by minimal interval semantics and as supported by the translation function \mathcal{X} . The generality of annotative indexing lies in the flexibility of what the features represent, which can include term statistics, document structure, and links. Since implementation details are hidden behind τ , ρ , and \mathcal{X} , an annotative index can be implemented with a wide variety of data structures. For example, Cottontail provides both static and dynamic index formats that employ (nearly) distinct storage structures.

Clarke et al. (1995a) and Clarke (1996) contain many examples illustrating queries and query processing. Suppose we want to find all objects (“:”) with “aeolian” in their “:title:”. the τ and ρ methods work together to provide an efficient solution.

```

1   $k \leftarrow 0$ 
2   $\mathcal{S} \leftarrow \{ \}$ 
3  while  $k \neq \infty$ :
4       $\langle (p, q), v \rangle \leftarrow \text{:title:}.\rho(k)$ 
5       $\langle (p', q'), v' \rangle \leftarrow \text{aeolian}.\tau(p)$ 
6      if  $q' \neq \infty$  and  $q' \leq q$ :
7           $\mathcal{S} \leftarrow \mathcal{S} \cup \{ \text{:}.\rho(q) \}$ 
8           $k \leftarrow q + 1$ 
9      else:
10          $k \leftarrow q'$ 

```

After execution, the set \mathcal{S} contains a set of annotations corresponding to the required objects, which is itself an annotation list. Lines 4 and 5 generate the next candidate title and occurrence of “aeolian”. Line 6 determines if the candidate title contains “aeolian”, and if so, we add the associated object to \mathcal{S} . Line 10 is the key to efficiency, setting k to skip titles that can’t contain the next candidate “aeolian”. If “aeolian” is relatively rare, and clustered in a relatively small number of objects, we may be able to avoid considering most titles. For simplicity, this solution assumes that all titles are contained in an object, which may not be true if the context mixes different types of text. As shown later, we would not normally materialize \mathcal{S} as a set, but rather would provide “lazy” access through $\mathcal{S}.\tau$ and $\mathcal{S}.\rho$ implemented in terms of τ and ρ for the “:”, “aeolian”, and “:title:” features.

As another example, we can define access methods for a window of $n > 0$ tokens as $\#n.\tau(k) \equiv \langle (k, k + n - 1), 0 \rangle$ and $\#n.\rho(k) \equiv \langle (k - n + 1, k), 0 \rangle$. In the example above, replacing “aeolian” with the feature #12 generates the set of objects with titles at least 12 tokens long. While titles can no longer be skipped, windows are generated “as needed” to test against each title. Note that $\#12.\rho(0) \equiv \langle (-11, 0), 0 \rangle$, providing a simple example of a negative address. Section 3.3 provides additional background on minimal interval semantics, including additional discussion regarding operators and query processing

3 Comparison with Prior Work

3.1 Static and Dynamic Inverted Indexes

Annotative indexes generalize inverted indexes. Büttcher et al. (2010) provides a review of inverted index file structures and associated query processing methods that remains reasonably current. They describe the core techniques that are still widely employed, along with experimental comparisons against competing techniques. A generic inverted index maps each term in a *vocabulary* — maintained in by a *dictionary* — to a *postings list* of *document identifiers* where the term appears. Postings lists often include term frequencies to support ranking formulae and term offsets to support phrase searching. Postings lists are typically gap-encoded and compressed with a method such as vByte (Williams and Zobel, 1999), which usually provides an acceptable trade off between compression ratio and decompression speed. Since query processing methods can often skip documents, synchronization points may be included in the compressed posting lists to improve performance and reduce the need for decompression (Moffat and Zobel, 1996).

Academic research on inverted indexes often views them through the lens of a static file structure, built once from a collection and never changed (e.g., Arroyuelo et al. (2018); Mallia et al. (2019); Mackenzie and Moffat (2020)). If the collection changes, the index is re-built from scratch. For example, if a researcher wants to remove near-duplicates from a collection because they are causing problems with their retrieval experiments, the researcher first filters the collection and then builds a new index for the filtered collection. A complete index re-build can be slow, even for a relatively small collection. At the very least, a complete re-build requires an end-to-end read of the collection to construct postings lists, so that the build time grows linearly with the size of the collection.

Prior research has considered a variety of dynamic update models for inverted indexes (Büttcher et al., 2010). The simplest model provides for *batch updates*, which build index structures for new documents and merge them into the original index without requiring a complete rebuild. During the merge, the index also deletes any unneeded documents. Ideally, the overall process is managed as a transaction, so that a failure during the update process does not corrupt the index structures. The batch update model supports only one transaction at a time. Starting a second update during a transaction either produces an error or blocks until the current transaction completes. If the index is queried during a transaction, atomicity should guarantee that a result over the original index is returned. While a batch update avoids some of the work required by a full rebuild, the index cannot evolve quickly. Depending on the final size of the index, it might take minutes or hours for a change to become visible to queries.

Under the *immediate-access* dynamic update model, changes become visible as soon as they are made (Büttcher et al., 2010). Social media search provides an important use case for immediate-access dynamic update. Asadi et al. (2013) describe index update in the EarlyBird search engine, developed for Twitter. EarlyBird search was designed for a single, high-volume update stream, with many concurrent readers and a strong temporal ranking signal, placing a high priority on recent tweets. Once a tweet is indexed, its indexing does not change. Moffat and Mackenzie (2023) explore trade-offs between insertion speed, query speed, and index size in an immediate-access dynamic index. They describe in-memory indexing structures that supports a stream of interleaved queries and document insertions. Document deletion is not supported, so that index grows with each insertion. To maintain a consistent view of the index, they assume that “all postings associated with each ingested document are processed into the index before the next query operation is permitted”, effectively requiring a read-lock on the index during each insertion. Eades et al. (2022) describe index structures for dynamic update that uses fixed volume of memory, with older documents expiring from the index.

Prior research on immediate-access dynamic update does not satisfy the requirements of annotative indexing. In particular, prior work assumes that indexing for a document happens all at once; no additional indexing for a document can be added at a later time (Büttcher et al., 2010). In contrast, annotative indexing enables novel use cases that require additional indexing. For example, imagine an annotative index supporting a document ingestion pipeline for a retrieval augmented generation (RAG) system that includes de-duplication, segmentation, and indexing stages. Each stage reads its input from the index and records its output by adding annotations. For an annotative index to fully support a document processing pipeline, the output from a stage must be immediately visible as soon as the stage finishes. However, each stage must see a complete and consistent view of the output from the previous stages. When stages are independent of each other, it should be possible for them to run concurrently. Since some stages may require considerable processing, it is important for updates to be durable, allowing the pipeline to recover quickly from a failure. To satisfy this scenario and realize the full benefits of annotative indexing, an annotative index must support concurrent access and ensure ACID properties of transactions, requirements that are not met by prior research.

3.2 First-Stage Retrieval

Over 30 years after its invention, the BM25 formula remains the touchstone for unsupervised first-stage retrieval (Robertson and Walker, 1994; Robertson et al., 1994). When compared to other unsupervised retrieval formulae from the 1990s and early 2000s — which may provide as-good-or-better retrieval effectiveness — BM25 exhibits term saturation properties that can be exploited to substantially improve query performance through WAND query processing (Broder et al., 2003; Petri et al., 2013; Turtle and Flood, 1995) and Block-Max WAND processing (Dimopoulos et al., 2013; Ding and Suel, 2011). Term saturation guarantees an upper bound on the weight given to any single query term, allowing us to skip documents whose score cannot exceed a threshold defined by scores of the current top- k documents.

In their description of standard WAND processing, Petri et al. (2013) assume posting lists will be accessed through three functions: 1) A *first* function, which creates an iterator for the list, 2) a *next* function, which advances the iterator by one posting, and 3) a *seek(d)* function, which advances iterator to the first document identifier greater than or equal to d . The τ and ρ operations in Equation 4 and Equation 5 generalize the *seek* function to intervals. When wrapped in an appropriate iterator, and with appropriate annotations, they directly support WAND processing over annotative indexes.

Using the τ and ρ operations, Cottontail can efficiently implement WAND processing. As a demonstration of performance we use the 6980 “dev small” queries and passages from the original MS MARCO test collection (Bajaj et al., 2018). The corpus comprises 8,841,823 passages with an uncompressed size of 2.85GB. Full cottontail indexing with BM25 annotations using the static index implementation described in Section 4 gives a compressed index of 4.4GB. In addition to the annotations required for BM25 ranking, this index includes token-level annotations, to implement phrase search and structural queries, along with the full text of the collection, to implement the \mathcal{X} translation function.

With BM25 parameters $b = 0.68$ and $k_1 = 0.82$ we achieve an MRR@10 of 0.185. These BM25 parameters are recommended by the Anserini onboarding guide, which uses this collection as an example for teaching indexing and retrieval³. MMR@10 is the standard precision metric for this test collection. Running with two threads per physical core, i.e., 46 concurrent queries, it requires less than 20 seconds to rank all queries to a depth of 10, giving a throughput of over 350 queries/second⁴. Running one query at a time gives an average query latency of 65ms. In comparison, using a system based on the Lucene search library, Lin et al. (2020) report a BM25 latency of 55ms and MMR@10 of 0.184 on the same collection. While Lin et al. (2020) do not indicate the hardware used for their measurements, nor do they report query throughput, this comparison suggests that our general indexing framework can be reasonably competitive with a specialized index developed through years of engineering effort.

In recent years, neural retrieval methods have eclipsed traditional unsupervised methods for first-stage retrieval. Neural first-stage retrieval methods fall into two camps: *sparse vector retrieval* and *dense vector retrieval*. Sparse vector retrieval methods represent queries and documents in a high-dimensional space, where each dimension corresponds to a token (Lin and Ma, 2021; Song et al., 2021) and most weights are zero, especially in query vectors. Sparsity allows these vectors to be stored in an inverted index; ranking requires only a dot product between the query and document vectors. Successful approaches to sparse retrieval include DeepCT (Dai and Callan, 2019), HDCT (Dai and Callan, 2020), uniCOIL (Lin and Ma, 2021) and SPLADE (Formal et al., 2021). In particular, SPLADE is widely recognized for its retrieval effectiveness (Lassance et al., 2024; Mallia et al., 2024; Bruch et al., 2024). Despite a few proposals for unsupervised neural sparse methods (e.g., Ma et al. (2023)), neural sparse methods are often called “learned sparse retrieval” methods to distinguish them from traditional unsupervised sparse methods, such as BM25.

Annotative indexing trivially supports learned sparse retrieval by creating an annotation for each element of a sparse vector. It is also trivial to support multiple sparse retrieval

3. <https://github.com/castorini/anserini/blob/master/docs/experiments-msmarco-passage.md>

4. All experiments reported in this paper were conducted on a Intel(R) Xeon(R) Gold 5120 CPU with 256GB of memory.

methods (e.g. BM25 and SPLADE) in the same index, or to use different ranking approaches at different structural levels (e.g. BM25 at the document level and SPLADE at the passage level). Unfortunately, learned weights do not provide the distributional properties that algorithms like WAND exploit to improve query performance. Score-at-a-time ranking approaches can partly address this problem (Mackenzie et al., 2021). It may also be possible to adapt block pruning and other methods to annotative indexes, for example, by adding additional annotations summarizing weights over blocks of documents (Mallia et al., 2024; Bruch et al., 2024; Mallia et al., 2017; Ding and Suel, 2011).

Dense vector retrieval is currently the focus of intense research, with multiple recent surveys available (Pan et al., 2024; Zhao et al., 2024). The simplest form of dense retrieval, *bi-encoder retrieval*, represents queries and documents in a low-dimensional space (e.g., 768 dimensions) where the values in most dimensions are non-zero. Ranking requires only a dot product between the query and document vectors (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Zhan et al., 2020). Various approximate k-nearest neighbor search methods can speed the ranking process. For example, Hierarchical Navigable Small World (HNSW) graphs arrange vectors in a hierarchy of proximity graphs that can be traversed in approximately logarithmic time (Malkov and Yashunin, 2020).

While each dimension could be represented as an annotation list, retrieval would be inefficient because of the relatively large number of non-zero values in a dense vector. To support dense vectors, we would need to extend annotative indexing with a vector store, which might map locations in the address space to vectors. Annotative indexes can also help support hybrid approaches that combine sparse and dense retrieval (Leonhardt et al., 2022). It may also be possible to encode and efficiently search HNSW graphs encoded as annotations.

3.3 Minimal Interval Semantics

Minimal-interval semantics were invented by the author for his Ph.D thesis nearly 30 years ago (Clarke, 1996). If we view the result of a text search over a string as a set of substrings that satisfy the requirements of the search, minimal-interval semantics provide a simple and natural way to linearize the set, as well as enabling fast and flexible algorithms for combining and filtering search results. If we specify the set of substrings S as a set of intervals (p, q) , minimal interval semantics allows these intervals to overlap but not to nest.

An interval (p, q) *overlaps* an interval (p', q') if either $p' \leq p \leq q' \leq q$ or $p' \leq q \leq q' \leq p$, but not both. An interval (p, q) is *nested* in an interval (p', q') if $(p, q) \neq (p', q')$ and $p' \leq p \leq q \leq q'$. If $a = (p, q)$ and $b = (p', q')$ are intervals, the notation $a \sqsubset b$ indicates that a nests in b ; the notation $a \sqsubseteq b$ indicates that a is *contained in* b : that either a and b are equal or that a nests in b . Intervals form a partial order under \sqsubseteq .

We formalize the reduction of a set of intervals S to a *generalized concordance list* as a function $\mathcal{G}(S)$:

$$\mathcal{G}(S) = \{a \mid a \in S \text{ and } \nexists b \in S \text{ such that } b \sqsubset a\}$$

A set S is a generalized concordance list if and only if $S = \mathcal{G}(S)$. Each interval in a generalized concordance list acts as a “witness” to the satisfiability of the requirements of the search (Boldi and Vigna, 2018). As a simple example, consider the query:

"peanut butter" \triangle "jelly doughnut",

Containment Operators

Contained In:

$$A \triangleleft B = \{a \mid a \in A \text{ and } \exists b \in B \text{ such that } a \sqsubseteq b\}$$

Containing:

$$A \triangleright B = \{a \mid a \in A \text{ and } \exists b \in B \text{ such that } b \sqsubseteq a\}$$

Not Contained In:

$$A \ntriangleleft B = \{a \mid a \in A \text{ and } \nexists b \in B \text{ such that } a \sqsubseteq b\}$$

Not Containing:

$$A \ntriangleright B = \{a \mid a \in A \text{ and } \nexists b \in B \text{ such that } b \sqsubseteq a\}$$

Combination Operators

Both Of:

$$A \triangle B = \mathcal{G}(\{c \mid \exists a \in A \text{ such that } a \sqsubseteq c \text{ and } \exists b \in B \text{ such that } b \sqsubseteq c\})$$

One Of:

$$A \nabla B = \mathcal{G}(\{c \mid \exists a \in A \text{ such that } a \sqsubseteq c \text{ or } \exists b \in B \text{ such that } b \sqsubseteq c\})$$

Follows:

$$A \diamond B = \mathcal{G}(\{c \mid \exists (p, q) \in A \text{ and } \exists (p', q') \in B \text{ where } q < p' \text{ and } (p, q') \sqsubseteq c\})$$

Figure 2: Fundamental operators for expressing structural relationships over generalized concordance lists, which underlie annotation lists. A and B can be any generalized concordance lists, including subqueries built from these operators.

where “ \triangle ” indicates Boolean conjunction. If we view the set of intervals satisfying the query “**peanut butter**” as the set of all intervals containing that string, of any length, then $\mathcal{G}(\text{“peanut butter”})$ is just the set of intervals corresponding to the string itself. If we view the set of intervals satisfying the conjunction as the set of intervals that contain both strings, then the set of minimal intervals that contain both strings is $\mathcal{G}(\text{“peanut butter”} \triangle \text{“jelly doughnut”})$, which may overlap but not nest. For example, the sentence:

Peanut butter on a jelly doughnut is better than a peanut butter sandwich.

contains two overlapping intervals which satisfy the conjunction under minimal interval semantics.

Figure 2 summarizes fundamental operators from Clarke (1996), where A and B are generalized concordance lists. The operators fall into two groups, *containment* and *combination*, which together support a wide range of queries specifying structural relationships. For example, the query for all objects (“:”) with “aeolian” in their “:title:” is:

$$: \triangleright (: \text{title:} \triangleright \text{aeolian})$$

Generalized concordance lists have the same access methods as annotation lists, but without associated values. They are just ordered sets of intervals under minimal interval semantics.

If S is a generalized concordance list, then:

$$S.\tau(k) = \{(p, q), \text{ where } (p, q) \text{ is the interval in } S \text{ with minimal } p \geq k\} \quad (6)$$

$$S.\rho(k) = \{(p, q), \text{ where } (p, q) \text{ is the interval in } S \text{ with minimal } q \geq k\} \quad (7)$$

A key observation of Clarke (1996) — echoed by Boldi and Vigna (2016) — is that evaluation can be “lazy”. Much like WAND processing, lazy evaluation allows us to skip solutions to subqueries that cannot lead to a solution for the overall query. If A and B are generalized concordance lists, we can implement τ and ρ access methods for each operator of Figure 2 in terms of τ and ρ for A and B , which in turn can subqueries built from these operators. For example, the ρ operator for $A \triangleright B$ can be written as:

```

1   $(A \triangleright B).\rho(k) \equiv$ 
2     $(p, q) \leftarrow A.\rho(k)$ 
3     $(p', q') \leftarrow B.\tau(p)$ 
4    if  $q' \leq q$ :
5      return  $(p, q)$ 
6    else:
7      return  $(A \triangleright B).\rho(q')$ 
```

If A corresponds to “:title:” and B corresponds to “aeolian”, we obtain a ρ access method for titles containing “aeolian”, which can in turn be combined with the generalized concordance list for objects to give a ρ access method for objects containing these titles. Compare this definition to the algorithm on page 113.

Unfortunately, there is no single summary of key theorems and algorithms for operations under minimal interval semantics. Clarke et al. (1995b) contains an overview with many examples. Clarke et al. (1995a) contains additional examples and algorithms for implementing the operators, which are extended with proofs by Clarke (1996). While τ and ρ are sufficient to implement the operators of Figure 2, Clarke (1996) also defines “backwards” version of these access methods that facilitate solutions that start with the last interval, which can be valuable in finding the most-recent solutions to queries over a growing index (Asadi et al., 2013). More recently, Boldi and Vigna (2016, 2018) present a mathematical foundation for minimal-interval semantics based on lattices.

Clarke and Cormack (2000) present an improved implementation framework for the combinational operators, including Boolean operators. Under this framework, finding all solutions to a combinational query with n terms requires no more than $O(n \cdot \mathcal{A})$ calls to access methods for the terms, where \mathcal{A} is the number of **solutions** to the query. Using *galloping search* (Büttcher et al., 2010, pp. 42–44) to implement the access methods for the terms gives an overall time complexity of $O(n \cdot \mathcal{A} \cdot \log(L/\mathcal{A}))$, where L is the length of the longest posting list for a term. The overall time complexity is nearly linear in the number of solutions, rather than the length of any postings list, as might be expected. If there are few solutions, most of the postings lists might be skipped. The reference implementation for annotative indexing, Cottontail, captures most of the cumulative insights from research on minimal interval semantics⁵.

5. <https://github.com/claclark/Cottontail/blob/main/src/gcl.cc>

3.4 Column Stores

A column store is a physical design strategy for relational database systems that partitions data primarily by column, rather than by row. As opposed to a traditional row-oriented strategy for physical database design, a column-oriented strategy is known to provide better performance on data analytics and other read-intensive workloads. Among other properties, grouping together values from a single column can improve compression because values within the same column are often similar or repetitive. This homogeneity makes it easier to apply compression techniques that exploit redundancy. The popularity of column stores grew from the success of systems such as C-Store (Stonebraker et al., 2005) and MonetDB (Idreos et al., 2012). Currently, open formats such as ORC and Parquet enable support for columnar storage on most platforms for data analytics.

Inverted indexes are close cousins of column stores (Mühleisen et al., 2014). If we consider the terms in the vocabulary as columns of a table whose rows represent documents, then we can imagine the table as containing term weights, perhaps with a special column containing document identifiers. Since most terms appear only in a relatively small number of documents, the table is sparse. Most of the entries are NULL. Computing a retrieval formula requires an aggregation over the columns corresponding to terms in the query. Since an inverted index organizes this table by column, only query columns need to be accessed to compute the retrieval formula.

Annotative indexes generalize inverted indexes to something close to a column store. If we store rows of a table as the content of the annotative index and treat the features as columns, then each annotation represents an entry in the table:

$$\langle column, (start, end), value \rangle,$$

where $(start, end)$ is the location of the value in the row. Data can be accessed by column through annotation lists, and by row through the translation function $\mathcal{X}(p, q)$.

3.5 Graph Structures

Many application areas now require database support to store and process large graphs structures (Sahu et al., 2023). For example, Facebook developed the Unicorn search engine to store and search social graph information at worldwide scale (Curtiss et al., 2013). Unicorn’s core file structures extend inverted lists with additional information, similar to annotative indexing. However, it does not support minimal-interval semantics. In an information retrieval context, graph data often takes the form of a *knowledge graph*, with Bast et al. (2025) providing a current survey. Knowledge graphs often encode relationships as subject-predicate-object triples: For example the triple:

$$\langle \text{Meryl_Streep} \rangle - \langle \text{won_award} \rangle - \langle \text{Best_Actress} \rangle$$

indicates that Meryl Streep won an Oscar for Best Actress.

An annotation list can encode a directed graph by storing a location in the address space as the value of annotation, so that the annotation $\langle G, p, v \rangle$ is interpreted as a link from an object containing the location p to an object containing the location v . For example, consider the trivial friend graph:

```

{"name": "Alice", "friends": ["Bob", "Carol", "Dave"]}
{"name": "Bob", "friends": ["Alice", "Dave"]}
{"name": "Carol", "friends": ["Alice"]}
{"name": "Dave", "friends": ["Bob", "Alice"]}
    
```

If the Alice object is stored at (0, 26) and the Bob object is stored at (27, 49), the annotation $\langle @friend, 7, 27 \rangle$ indicates a link from Alice’s friends array to Bob, where 7 is the address of the token “Bob” the occurs in Alice’s friend array. Using a similar approach, annotations can encode subject-predicate-object triples.

$\langle predicate, subject, object \rangle$.

To encode a triple indicating that Meryl Streep won an award for best actress, the *predicate* would be encoded as the feature `won_award`, the *subject* would be an address in the record associated with Streep, and the *object* would be an address in the record associated with the best actress award.

4 Organization of an Annotative Index

In this section, we consider the organization and construction of an annotative index, using our reference implementation, Cottontail, as an example. Cottontail provides two distinct implementations of the index structures, a *static index* and a *fully dynamic index*. The static index supports larger collections, where it may not be possible to maintain the entire collection in memory. The static index reads annotation lists from storage only for query processing and index update; it supports only a single update transaction at a time under the batch update model. The dynamic index maintains all active index structures in memory, while still durably committing transactions to storage. In this section, we focus on the basic index construction process, which applies to both static and dynamic indexes. Section 6 extends this material with details for the fully dynamic index, including support for immediate update and multiple concurrent readers and writers.

An annotative index extends and generalizes an inverted index, as outlined in Section 3.1, annotations are indexed by feature, with annotations ordered by the start address (and equivalently the end address) of their intervals. If the annotations for feature f are

$$a_0 = \langle f, (p_0, q_0), v_0 \rangle, a_1 = \langle f, (p_1, q_1), v_1 \rangle, a_2 = \langle f, (p_2, q_2), v_2 \rangle, \dots$$

then $\forall i, p_i < p_{i+1}$ and $q_i < q_{i+1}$. Since they strictly increase, successive start (and end) addresses can be gap-encoded and compressed with vByte, or other methods developed for compressing postings lists. For a given f , if $\forall i, p_i = q_i$, then its end addresses can be compressed away. Similar to column stores, values will tend to share distributional properties that can be exploited to improve compression. For a given f , if $\forall i, v_i = 0$, its values can be compressed away.

Figure 3 provides an overview of the major components of Cottontail. The various components of an annotative index are grouped into a **Warren**, which manages transactions and simplifies common operations that interact with multiple components⁶. Apart from a

6. The author is aware that eastern cottontail rabbits are solitary and don’t live in warrens.

Warren: Groups the following components and manages transactions.

Operations: `clone`, `start`, `end`, `transaction`, `ready`, `commit`, `abort`

Tokenizer: Facilitates content addressability (Section 4).

Operations: `tokenize`, `split`, `skip`

Featurizer: Maps a feature (expressed as a string) to a 64-bit value (Section 4).

Operations: `featurize`

Annotator: Inserts and deletes annotations (Section 4).

Operations: `annotate`, `erase`

Appender: Appends text to the content (Section 4).

Operation: `append`

Idx: Provides read access to annotations (Section 5).

Operation: `hopper(f)` — create a cursor (called a **Hopper**) for the feature `f`

Txt: Provides read access to content (Section 5).

Operation: `translate(p, q)` — return content associated with the interval $(\mathcal{T}(p, q))$

Figure 3: Major components of Cottontail, the reference implementation for annotative indexing. A **Warren** object contains and manages one instance of each of the other components (**Tokenizer**, **Featurizer**, etc.). Cottontail provides multiple versions of each component, each specialized for a different purpose. Section numbers indicate where the component is discussed. The transaction model for a **Warren** is discussed in Section 6.

Warren, each component implements no more than three operations. Cottontail provides multiple versions of each component, each specialized for a different purpose, which can be mixed and matched in a **Warren**.

A **Tokenizer** facilitates content addressability by splitting strings into tokens, computing token boundaries, and skipping tokens. Support for ASCII content with HTML-style tags is provided by **AsciiTokenizer**, which is intended for use with older TREC collections. Generic support for Unicode is provided by **Utf8Tokenizer**, which is intended for use with JSON and other modern content. The role of a **Tokenizer** in a **Warren** is limited to facilitating content addressability. Other tokenization (e.g., language specific or WordPiece) can be used by features in annotations to support ranking and other applications.

Internally, cottontail represents an annotation as four 64-bit values, using a **Featurizer** to map a feature expressed as a string to a 64-bit value. **HashingFeaturizer** maps strings to 64-bit values with a MurmurHash function. **HashingFeaturizer** can be wrapped by other **Featurizer** classes to record vocabulary items and to exclude selected features

from indexing. By convention, features mapped to 0 are not indexed. For example, the `JsonFeaturizer` wraps any `Featurizer`, and maps to 0 those tokens that represent JSON structural elements, such as the curly braces surrounding objects.

An `Appender` and an `Annotator` work together for index construction and update. Both support two-phase commit protocols, with the overall transaction managed by the `Warren`. An `Appender` appends data to the content through its `append` operation:

$$\text{append}(\text{content}) \rightarrow (p, q)$$

The `append` operation returns the interval where the appended content is located. An `Annotator` adds an annotation to the index through the `annotate` operation:

$$\text{annotate}(f, v, p, q)$$

which adds the annotation $\langle f, (p, q), v \rangle$, where the value v is optional.

Figure 4 shows a partial trace of `append` and `annotate` operations, while adding a nested JSON object to an annotative index. With the help of a fast JSON parser⁷, support for a general JSON store requires less than 500 lines of C++ beyond the core generic annotative indexing code. The example object is taken from a set of open source examples available on Adobe’s website⁸. The order that JSON key-value pairs are added differs from the textual order in the object because the object is first parsed into a C++ map and then traversed to add the object to the annotative store⁹.

In the figure, the operation `append("batters":)` appends four tokens: `"`, `"batters"`, `"`, and `“:”`, returning the interval (1, 4). Tokens marking structural elements of the JSON object (`{`, `}`, `"`, `“:”`, etc.) are encoded as special tokens using Unicode noncharacters, which are permanently reserved for the internal use by systems that store and transmit text. With this encoding, the `translate` operation of a `Txt` component, which implements $\mathcal{X}(p, q)$, can return any interval of the content and recognize the difference between a `“:”` separating a JSON key-value pair and a `“:”` that happens to appear in a string.

For conciseness, the trace omits `annotate` operations that add annotations for single tokens, which are automatically performed as part of an `append` operation. For example, as part of the `append("Regular")` operation, the annotation $\langle \text{regular}, 35 \rangle$ is automatically added. As previously mentioned, `JsonFeaturizer` returns 0 for tokens marking structural elements, suppressing automatic annotation to avoid unnecessary indexing.

All structure and nesting is retained in the features. For example, the annotation $\langle \text{:batters:batter:[0]:type:}, (24, 26) \rangle$ indicates the `“type”` property of the first element of the `“batter”` array of the `“batters”` property. A JSON object is not “flattened” in any sense. The content (i.e., $\mathcal{X}(0, 254)$) contains the full JSON object, which can be accessed by the `translate` operation of the `txt` component.

By convention, the feature `“:”` is used as the root of the object, as seen in the annotation $\langle \text{:}, (0, 254) \rangle$. Individual objects in a collection of JSON objects, e.g. a JSONL file, can be accessed through this `“:”` feature. In the annotation $\langle \text{:batters:batter:}, (10, 84), 4 \rangle$ the value 4 gives the length of the array. In a later example, we apply the convention of storing

7. <https://github.com/nlohmann/json>

8. https://opensource.adobe.com/Spry/samples/data_region/JSONDataSetSample.html

9. <https://github.com/clacklark/Cottontail/blob/main/src/json.cc>

```

{
  "id": "0001",
  "type": "donut",
  "name": "Cake",
  "ppu": 0.55,
  "batters":
  {
    "batter":
    [
      { "id": "1001",
        "type": "Regular"},
      { "id": "1002",
        "type": "Chocolate"},
      { "id": "1003",
        "type": "Blueberry"},
      { "id": "1004",
        "type": "Devil's Food"}
    ]
  },
  "topping":
  [
    { "id": "5001",
      "type": "None"},
    { "id": "5002",
      "type": "Glazed"},
    { "id": "5005",
      "type": "Sugar"},
    { "id": "5007",
      "type": "Powdered Sugar"},
    { "id": "5006",
      "type": "Chocolate with Sprinkles" },
    { "id": "5003",
      "type": "Chocolate" },
    { "id": "5004",
      "type": "Maple" }
  ]
}

```

```

transaction()
append({} → (0, 0)
append("batters:") → (1, 4)
append({} → (5, 5)
append("batter:") → (6, 9)
append([] → (10, 10)
append({} → (11, 11)
append("id:") → (12, 15)
append("1001") → (16, 18)
annotate(:batters:batter:[0]:id:, 16, 18)
append(,) → (19, 19)
append("type:") → (20, 23)
append("Regular") → (24, 26)
append({} → (27, 27)
annotate(:batters:batter:[0]:type:, 24, 26)
annotate(:batters:batter:[0]:, 11, 27)
...
annotate (:batters:batter:, 10, 84, 4)
...
append("name:") → (95, 98)
append("Cake") → (99, 101)
annotate(:name:, 99, 101)
append(,) → (102, 102)
append("ppu:") → (103, 106)
append("0.5500") → (107, 110)
annotate(:ppu:, 107, 110, 0.55)
...
annotate(:, 0, 254)
ready()
commit()

```

Figure 4: Constructing an annotative index. The inset on the right shows a partial trace of `append` and `annotate` operations during the addition of the JSON object on the left.

the array length as the value for the array feature to step through (“explode”) arrays of different lengths in different objects. These conventions, as well as other conventions used to support a JSON store, are independent of the underlying associative index.

5 Query Processing

The τ and ρ access methods, as defined by Equations 4 and 5, provide the foundation for query processing. In Cottontail, the `hopper(f)` operation of the `Idx` component creates a `Hopper` object for the 64-byte feature value `f`. A `Hopper` object acts as a cursor, supporting the τ and ρ access methods over the feature and caching the most recent result from each access method. All accesses to the underlying index structures are abstracted by τ and ρ , which we are then free to implement in any suitable way. For example, the index structures might include synchronization points to allow the `Hopper` to skip annotations (Moffat and Zobel, 1996). The current version of Cottontail represents annotation lists as arrays, compressed until active, and skips annotations with galloping search. However, since the index structures are known only to the `Idx` component, it could employ any file structures and storage strategies able to efficiently support the τ and ρ access methods¹⁰.

The translation function $\mathcal{X}(p, q)$ is implemented by the `translate(p, q)` operation of the `Txt` component. A typical query processing loop for a structural query expressing containment relationships might start with a query Q expressed by the operators of Figure 2. Calls to τ or ρ generate successive solutions, with the content translated and the results aggregated as needed.

```

1  Solve( $Q$ )  $\equiv$ 
2     $\langle (p, q), v \rangle \leftarrow Q.\tau(0)$ 
3    while  $p \neq \infty$ :
4      Translate/Aggregate  $\langle (p, q), v \rangle$ 
5       $\langle (p, q), v \rangle \leftarrow Q.\tau(p + 1)$ 

```

The access methods return $\langle (\infty, \infty), 0 \rangle$ to indicate the end of the list. As the solutions are generated, the τ and ρ operators allow solutions to subqueries to be skipped when they cannot lead to a solution for the overall query. The specific translation (\mathcal{X}) and aggregation operations required on line 4 depend on the problem at hand. Aggregations include the standard SQL aggregations (MAX, MIN, COUNT, etc.), which need to be preformed in memory.

To provide more concrete examples, we use the heterogeneous collection of JSON objects presented in Figure 5. We base our examples on this collection due to its level of heterogeneity and its independence from this work. The collection was originally created as a resource for exploring and learning MongoDB. Compared to standard benchmarking tools (Belloni et al., 2022) it provides a reasonable source of clear and simple examples, with an emphasis on heterogeneity. Single-thread build time for this collection is just over 4 minutes for a static index and just over 3 minutes for a dynamic index.

10. The name “Cottontail” was inspired by the ability of the τ and ρ access methods to efficiently “hop” around the index.

Data Set	Description	Records	Size
books	Descriptions of technical books	431	524K
city_inspections	Results of NYC business inspections	81,047	23M
companies	Overviews of tech companies	18,801	74M
countries-big	Country names by language	21,640	2291K
covers	Book ratings	5,071	470K
grades	Grades for homework assignments	280	91K
products	Phone and cable products	11	2K
profiles	Update log records	1,515	454K
restaurant	Restaurant addresses and ratings	2,548	666K
students	Student grades	200	34K
trades	Stock trades	1,000,001	231M
zips	NYC zip codes	29,353	3107K
Total		1,160,898	337M

Figure 5: Curated collection of heterogeneous JSON objects compiled by Özler as a resource for exploring MongoDB (<https://github.com/ozlerhakan/mongodb-json-files>).

Figure 6 presents these examples. The Cottontail repo contains associated source code¹¹. For each query, we give a description in English, a description in an SQL-like notation, and a query in the structural query notation of Figure 2. The source code should be consulted for full details. The figure includes query execution times for both static and dynamic indexes.

Examples 1-3 follow the general pattern above, i.e. a single query with different types of aggregation. Example 4 involves exploding an array containing author names. In the figure, the structural query for this example returns each array of author names as a whole, while the example code in the repo illustrates the use of array indexes to access individual elements one at a time. Example 5 requires roughly a second on both indexes. Processing this query requires over 80,000 accesses to the content, corresponding to an average access time of $20\mu s$ on the static index. Even with various caching methods in place, there are limits on random access to compressed text. As much as possible, query processing should take place over the annotations.

The “FROM *” notation in Examples 7 is not valid SQL. If it were, it would imply a Cartesian product of all tables. Here, it suggests the ability to run queries that span objects with different schema. Examples 8 and 9 provide a more substantial example of annotative indexing that enables unified queries over objects with different schema. The objects in many of the subcollections include properties indicating their creation date. For example, in the `city_inspections` subcollection, dates are specified in a human readable format (e.g. `{"date": "Feb 20 2015"}`). In the `companies` subcollection, some dates are specified as UNIX timestamps in milliseconds (e.g. `"created_at" : { "$date" : 1180075887000 }`). With annotative indexing, we can annotate the objects to provide consistent date annotations, allowing Example 9 to count the objects created on a specific date across all subcollections.

11. <https://github.com/claclark/Cottontail/blob/main/apps/json-examples.cc>

	Static	Dynamic
Example 1: Statistics for restaurant ratings SELECT MIN(rating), AVG(rating), MAX(rating) FROM restaurant :rating: \triangleleft Files/restaurant.json	14 ms	<1 ms
Example 2: How many zip codes does New York have? SELECT COUNT(*) FROM zips WHERE CITY = "NEW YORK" (:city: \triangleright "New York") \triangleleft Files/zips.json	23 ms	2 ms
Example 3: Names of nanotech companies SELECT name FROM companies WHERE category_code CONTAINS "nanotech" :name: \triangleleft (: \triangleright (nanotech \triangleleft (:category_code: \triangleleft Files/companies.json)))	133 ms	3 ms
Example 4: Titles and authors of books SELECT title, EXPLODE(authors) AS author FROM books (:title: ∇ :authors:) \triangleleft Files/books.json	95 ms	21 ms
Example 5: How many stock trades? SELECT COUNT(*) FROM trades : \triangleleft Files/trades.json	70 ms	71 ms
Example 6: Outcomes from city inspections SELECT result, COUNT(result) FROM city_inspections GROUP BY result :result: \triangleleft Files/city_inspections.json	1,686 ms	939 ms
Example 7: How many objects in the database? SELECT COUNT(*) FROM * :	< 1 ms	< 1 ms
Example 8: Titles of books published in 2008 SELECT title FROM books WHERE created \geq '2008-01-01' AND created \leq '2008-12-31' :title: \triangleleft (Files/books.json \triangleright year=2008)	13 ms	9 ms
Example 9: Count objects created on December 1, 2008. SELECT COUNT(*) FROM * WHERE created = '2008-12-01' : \triangleright (year=2008 \triangle month=12 \triangle Day=01)	13 ms	4 ms

Figure 6: Illustrative examples of containment and other operations over the JSON collection from Figure 5, with query processing times over static and dynamic index structures. Examples 8 and 9 depend on additional date annotations not present in the original JSON. The SQL queries are provided for explanatory purpose; they cannot be directly executed by the reference implementation. The structural queries describe index access only; additional processing is required to complete query processing, including aggregations.

6 Dynamic Update

Annotative indexing fosters a dynamic view of the content it stores. After we append text to the content, we can annotate it in different ways and for different purposes. For example, the transformations applied to the MS MARCO corpus described in the introduction, including tagging and segmentation into passages, could be achieved through annotations. For ranking purposes, term frequency values at the document level can be combined with sparse learned weights at the passage level to support hybrid search. Fields in heterogeneous collections of objects can be unified and related objects can be linked.

This section outlines an approach to dynamic update of an annotative index that maximizes flexibility, including support for multiple simultaneous readers and writers. Updates are grouped into transactions. At the start of a transaction, a snapshot is taken of the index state, which remains active until the transaction is committed or aborted. Both content and annotations in this snapshot can be accessed on read-only basis until the transaction ends. For example, during the transaction we might read the content to identify sentence boundaries in passages, or to compute term statistics. During the transaction, we can append to the content and add annotations, but these changes will not be immediately visible in the snapshot. We can also erase content and annotations. Once the update is complete, we follow a two-phase protocol to commit or abort the update, allowing us to support transactions that span independent annotative indices. After the transaction is complete, the updated content and annotations become visible.

While the Cottontail’s static index supports only one transaction at a time, its dynamic index supports multiple concurrent transactions. Each transaction is managed by a **Warren** (see Figure 3). The **clone** operation allows a **Warren** to be copied for the purpose of supporting concurrent transactions, with each clone managing one transaction at a time. For example, in a multi-threaded application each thread could **clone** a copy for its own use. The **start** operation captures the read-only snapshot of the index, while the **end** operation releases this snapshot. Any accesses to the **Warren**, even read-only access, must be bracketed by a **start/end** pair. The **transaction** operation starts a write transaction, at which point the **Appender** and **Annotator** may be used. In addition, to the **annotate** operator, the **Annotator** supports an **erase** operation that removes the content and its annotations over a specified interval by annotating the interval with the reserved feature 0. **Text** and **Hopper** objects skip these intervals until the associated content and annotations can be garbage collected. The remaining operations — **ready**, **commit**, and **abort** — complete the two-phase commit protocol. The update is not visible to the **Warren** until after the **end** operation, followed by another **start**.

Internally, each committed transaction creates a special update **Warren** object that contains only the newly added content and annotations¹². After a commit, an update **Warren** object is immutable. At the start of the ready phase of the two-phase commit, the index assigns an update **Warren** a sequence number. A vector of **Warren** objects in sequence order provides the snapshot used for read access. In the background **Warren** objects are merged and garbage collected, with a merged **Warren** representing a subindex of the full index, corresponding to a range of updates in sequence order. Once a **Warren** is merged into a larger range and is released from all active snapshots, it is deleted.

12. In a dynamic index, warrens multiply like rabbits.

During an update, content and annotations are assembled in a separate address space. At the start of the ready phase, when the index knows the final length of the appended content, it assigns a permanent address interval to the content and maps newly added annotations to this interval. During the ready phase the update is also logged durably to storage. If the commit is aborted after the ready phase, the assigned address interval becomes a gap, and the update is garbage collected from the log. During the update process, a global lock is held only for brief periods, such as when a snapshot is taken or when sequence numbers and address intervals are assigned.

Cottontail supports ACID properties of transactions. Transactions are fully atomic, with newly added content and annotations remaining invisible to `Text` and `Idx` operations until the transaction is committed. Cottontail guarantees consistency in that updates to annotations preserve minimal interval semantics. However, to maximize concurrency, Cottontail provides limited support for isolation. If concurrent transactions add annotations for the same feature that nest, the index retains only the innermost. If concurrent transactions add annotations with the same start and end addresses, the index retains only the value from the one with the largest sequence number. A failure before the start of a final commit phase of a two-phase commit, guarantees that the transaction is aborted, with no changes. A failure after a commit guarantees that the update is durably recorded. A failure during commit processing will leave the index in a consistent state, with the transaction either committed or aborted.

Figure 7 proves an illustration of dynamic update with multiple concurrent readers and writers¹³. The figure recapitulates four years of older TREC experiments when the test collection changed substantially from year to year (Voorhees and Harman, 1998). Documents for the test collection were distributed on five disks, encoded in an HTML-like format and organized into 4,905 files. TREC-4 used disks 2 and 3; TREC-5 used 2 and 4; TREC-6 used 4 and 5; TREC-7 dropped the low quality CR subcollection from disk 4. 50 new queries were introduced each year, but one query was excluded from TREC-4, leaving 199 queries in total. Each query was judged for relevance over the collection from the year it was introduced, with an average of 1,866 judgments/query. The figure was generated by hundreds of threads concurrently reading and writing a Cottontail dynamic index, including:

1. 28 appending threads, one for each core. Together they append the entire collection, a file at a time. Each file is appended as a separate transaction. After each append is committed, the thread re-reads the documents from the index, computes term statistics for them, and writes the statistics to the index as a second transaction. Finally, if there are documents in the file that are relevant to any of the queries, annotations reflecting these relevance judgments are written to the index as annotations in a third transaction.
2. 199 querying threads, one for each query. Each repeatedly starts a read access, runs its query with BM25, expands the query using pseudo-relevance feedback over the top 20 documents, runs the expanded query to return the top 1000 documents, reads relevance judgments from the index, computes average precision, and reports it on output where it is captured for later summarization on a per-year basis.

13. <https://github.com/claclark/Cottontail/blob/main/apps/trec-example.cc>

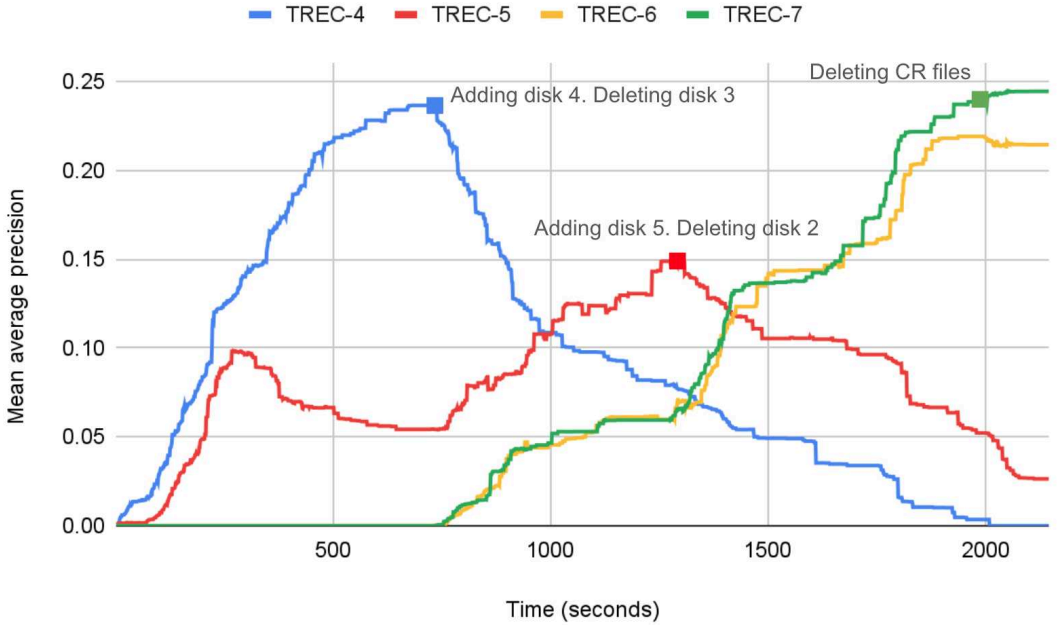


Figure 7: Example of transaction processing in cottontail. The example recapitulates four years of early TREC experiments, when the collection was changing significantly from year to year. The example was generated by 28 appending threads — one for each processor core — one deletion thread, and 199 querying threads — one for each query in the TREC-4 to TREC-7 test collections. The appending threads append each of the 4,905 files in the TREC collection as a separate transaction. They then add ranking statistics and relevance information as separate transactions. The deletion thread removes documents, so that collection evolves from year to year. The querying threads run continuously, each executing a BM25 query with pseudo-relevance feedback and then computing mean average precision using relevance information from the index. The lines in the figure plot MAP values as they change over the course of the experiment.

3. One deletion thread. It erases documents, a file at a time, so that the collection evolves over time. Each file is erased as a separate transaction. The squares in the figure indicate points where the deletion thread synchronizes with the other threads so that all queries are executed at least once on the entire collection for a given year.

In addition to these application-level threads, maintenance threads work throughout the experiment to merge and garbage collect the index. The experiment requires 16,442 update transactions in total. By the end of the experiment, these have been merged into 12 subindexes, each corresponding to a thousand or so sequence numbers. Throughout the experiment, processor utilization essentially remains at 100% on all cores.

As documents are added to the index for a given year, the MAP value for that year increases until it hits a synchronization point. It then drops as documents are deleted. The

BM25 parameters are tuned for more recent collections. The peak MAP values represent good performance on TREC-6 and TREC-7, and reasonable performance on TREC-4 and TREC-5.

7 Conclusion

This paper introduces and explores annotative indexing, a novel and flexible indexing framework, which unifies and generalizes inverted indexes, column stores, object stores, and graph databases. A particular feature of annotative indexing is its ability to manage heterogeneous collections of semi-structured data, unifying common elements across diverse formats. Text in any format can simply be appended to the content, with annotations added at a later time for a variety of purposes, such as sentence segmentation, tagging, or indexing for ranked retrieval.

Integrating annotative indexing into a retrieval augmented generation (RAG) system (Gao et al., 2024) forms a primary focus for current and future work. Given a few examples, a large language model (LLM) can generate structural queries using the operators of Figure 2, allowing natural language queries to be translated into structured queries over heterogeneous content. For example, imagine a life-logging application supported by a RAG system that integrates an annotative index. Messages, mail, conversations, and other experience could be poured into the index as content for ongoing tagging, linking, indexing, and other annotation. From the perspective of a person using the application, querying their past experience (“I really liked the movie I saw on the plane last weekend. What are similar movies I haven’t seen yet?”) happens in natural language, but internally this query could be handled by a combination of ranked retrieval and structured queries to a knowledge graph linked with the experiences.

Extending annotative indexing to support dense retrieval provides another immediate goal. While a 64-bit value in an annotation cannot store a dense vector, it can store a vector identifier. However, to better support a fully dynamic index, the author plans to mimic the approach taken for the content translation function $\mathcal{X}(p, q)$ by associating vectors with positions in the address space. A vector mapping function $\mathcal{V}(p)$ would return the vector associated with a location in the address space, presumably the location where the corresponding content appears. In this way, dense vectors can be garbage collected as intervals in the address space are erased.

To provide further support for dense retrieval, current work also includes an exploration of methods for encoding HNSW graphs as annotations. Graph structures can be represented in two ways by an associative index. First, as suggested in Section 3.5, we can store an address as the value in an annotation, so that $\langle G, p, v \rangle$ indicates a directed edge from p to v in the graph G . However, unless we are careful with updates, this representation can create “dangling references” to deleted content. An alternative representation stores a feature representing a list of out edges as the value in an annotation. Under this representation, the value in the annotation $\langle G, p, E_p \rangle$ is a feature indicating outlinks from the content at p in the graph G . An annotation for E of the form $\langle E_p, p' \rangle$ indicates a directed edge from p to p' . While the details are left for a future paper, this second representation should allow for the representation and traversal of HNSW graphs through annotations.

At the time of writing, the largest collection indexed by Cottontail is the 350GB C4 corpus,¹⁴ which the author routinely uses for cross-collection pseudo-relevance feedback. Ongoing work includes scaling Cottontail to handle larger collections, as well as fully distributed collections. Cottontail has indexed Wikidata as eight shards directly from its JSON dump,¹⁵ and the author is exploring support for knowledge graph queries over this collection.

At the time of writing, Cottontail remains an experimental system. Compilation requires the Bazel build system and the Boost C++ library, both of which require some effort to install. The system has not been ported to Mac or Windows, and it currently runs only on Ubuntu. In the near future, I plan a single-file release in the spirit of SQLite, along with examples illustrating common use cases. Wider use of Cottontail also requires a Python wrapper.

Current performance on basic BM25 ranking does not quite match that of Lucene. In the immediate future, I plan to focus attention on improving performance of the lowest level code for the **Hopper** operations, which should improve general performance, including ranking. Finally, the only delete operation supported by Cottontail is to erase all content and annotations from an interval of the address space. Additional delete operations might delete specific annotations or all annotations for a given feature.

Acknowledgments and Disclosure of Funding

The reference implementation for annotative indexing has its roots as a pandemic project. While I do not exactly thank the pandemic, I appreciate it gave me time to do some things that I would not otherwise have time to do. I made a final push to complete this work while was on a six-month sabbatical May to October 2024. During May and June, Yiqun Liu, Min Zhang, and Qingyao Ai kindly hosted me for a visit to Tsinghua University in Beijing. During September and October, Craig Macdonald and Iadh Ounis kindly hosted me for a visit to the University of Glasgow. Mark Smucker and Negar Arabzadeh read earlier versions of this paper and provided helpful feedback.

References

- Diego Arroyuelo, Mauricio Oyarzún, Senén González, and Víctor Sepúlveda. Hybrid compression of inverted lists for reordered document collections. *Information Processing & Management*, 54(6):1308–1324, 2018.
- Nima Asadi, Jimmy Lin, and Michael Busch. Dynamic memory allocation policies for postings in real-time Twitter search. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1186–1194, 2013.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated machine reading comprehension dataset, 2018. URL <https://arxiv.org/abs/1611.09268>.

14. <https://huggingface.co/datasets/allenai/c4>

15. https://www.wikidata.org/wiki/Wikidata:Database_download

- Hannah Bast, Johannes Kalmbach, Theresa Klumpp, and Claudius Korzen. Knowledge graphs. In Omar Alonso and Ricardo Baeza-Yates, editors, *Information Retrieval: Advanced Topics and Techniques*, chapter 2. ACM Press, 2025.
- Stefano Belloni, Daniel Ritter, Marco Schröder, and Nils Rörup. Deepbench: Benchmarking JSON document stores. In *9th International Workshop on Testing Database Systems*, page 1–9, 2022.
- Paolo Boldi and Sebastiano Vigna. Efficient optimally lazy algorithms for minimal-interval semantics. *Theoretical Computer Science*, 648:8–25, 2016.
- Paolo Boldi and Sebastiano Vigna. On the lattice of antichains of finite intervals. *Order*, 35(1):57–81, 2018.
- Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. Efficient query evaluation using a two-level retrieval process. In *12th International Conference on Information and Knowledge Management*, page 426–434, 2003.
- Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. Efficient inverted indexes for approximate retrieval over learned sparse representations. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 152–162, 2024.
- Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2010.
- Charles L. A. Clarke. *An Algebra for Structured Text Search*. PhD thesis, University of Waterloo, 1996. URL <https://plg.uwaterloo.ca/~claclark/phd.pdf>.
- Charles L. A. Clarke and Gordon V. Cormack. Shortest-substring retrieval and ranking. *ACM Transactions on Information Systems*, 18(1):44–78, 2000.
- Charles L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38(1):43–56, 1995a.
- Charles L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. Schema-independent retrieval from heterogeneous structured text. In *4th Annual Symposium on Document Analysis and Information Retrieval*, pages 279–289, Las Vegas, Nevada, 1995b.
- Michael Curtiss, Iain Becker, Tudor Bosman, Sergey Doroshenko, Lucian Grijincu, Tom Jackson, Sandhya Kunnatur, Soren Lassen, Philip Pronin, Sriram Sankar, Guanghao Shen, Gintaras Woss, Chao Yang, and Ning Zhang. Unicorn: A system for searching the social graph. *VLDB Journal*, 6(11):1150–1161, 2013.
- Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval, *ArXiv preprint arXiv:1910.10687*, 2019. URL <https://arxiv.org/abs/1910.10687>.

- Zhuyun Dai and Jamie Callan. Context-aware document term weighting for ad-hoc search. In *The Web Conference*, page 1897–1907, 2020.
- Constantinos Dimopoulos, Sergey Nepomnyachiy, and Torsten Suel. Optimizing top-k document retrieval strategies for block-max indexes. In *6th ACM International Conference on Web Search and Data Mining*, page 113–122, 2013.
- Shuai Ding and Torsten Suel. Faster top-k document retrieval using block-max indexes. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 993–1002, 2011.
- Patrick Eades, Anthony Wirth, and Justin Zobel. Immediate text search on streams using apoptotic indexes. In *44th European Conference on IR Research*, page 157–169, 2022.
- Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2288–2292, 2021.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, *ArXiv preprint arXiv:2312.10997*, 2024. URL <https://arxiv.org/abs/2312.10997>.
- Stratos Idreos, Fabian Groffen, Niels Nes, Stefan Manegold, Sjoerd Mullender, and Martin Kersten. MonetDB: Two decades of research in column-oriented database architectures. *IEEE Data Engineering Bulletin*, 35(1):40–45, 2012.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2020.
- Carlos Lassance, Hervé Dejean, Stéphane Clinchant, and Nicola Tonellotto. Two-step SPLADE: Simple, efficient and effective approximation of splade. In *46th European Conference on Information Retrieval*, page 349–363, 2024.
- Jurek Leonhardt, Koustav Rudra, Megha Khosla, Abhijit Anand, and Avishek Anand. Efficient neural ranking using forward indexes. In *ACM Web Conference*, page 266–276, 2022.
- Jimmy Lin and Xueguang Ma. A few brief notes on DeepImpact, COIL, and a conceptual framework for information retrieval techniques, *ArXiv preprint arXiv:2010.11386*, 2021. URL <https://arxiv.org/abs/2106.14807>.
- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Distilling dense representations for ranking using tightly-coupled teachers. *ArXiv preprint arXiv:2010.11386*, 2020. URL <https://arxiv.org/abs/2010.11386>.
- Xueguang Ma, Hengxin Fun, Xusen Yin, Antonio Mallia, and Jimmy Lin. Enhancing sparse retrieval via unsupervised learning. In *1st Annual International ACM SIGIR Conference*

- on *Research and Development in Information Retrieval in the Asia Pacific Region*, page 150–157, 2023.
- Joel Mackenzie and Alistair Moffat. Examining the additivity of top-k query processing innovations. In *29th ACM International Conference on Information & Knowledge Management*, page 1085–1094, 2020.
- Joel Mackenzie, Andrew Trotman, and Jimmy Lin. Wacky weights in learned sparse representations and the revenge of score-at-a-time query evaluation, *ArXiv preprint arXiv:2110.11540*, 2021. URL <https://arxiv.org/abs/2110.11540>.
- Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 824–836, 2020.
- Antonio Mallia, Giuseppe Ottaviano, Elia Porciani, Nicola Tonellotto, and Rossano Venturini. Faster BlockMax WAND with variable-sized blocks. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 625–634, 2017.
- Antonio Mallia, Michał Siedlaczek, and Torsten Suel. An experimental study of index compression and DAAT query processing methods. In *41st European Conference on IR Research*, page 353–368, 2019.
- Antonio Mallia, Torsten Suel, and Nicola Tonellotto. Faster learned sparse retrieval with block-max pruning. In *47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2411–2415, 2024.
- Alistair Moffat and Joel Mackenzie. Efficient immediate-access dynamic indexing. *Information Processing & Management*, 60(3), 2023.
- Alistair Moffat and Justin Zobel. Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*, 14(4):349–379, 1996.
- Hannes Mühleisen, Thaer Samar, Jimmy Lin, and Arjen de Vries. Old dogs are great at new tricks: Column stores for IR prototyping. In *37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 863–866, 2014.
- James Jie Pan, Jianguo Wang, and Guoliang Li. Survey of vector database management systems. *The VLDB Journal*, 33(5):1591–1615, 2024.
- Matthias Petri, J. Shane Culpepper, and Alistair Moffat. Exploring the magic of WAND. In *18th Australasian Document Computing Symposium*, page 58–65, 2013.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.
- S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, 1994.

- S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *3rd Text REtrieval Conference*, 1994.
- Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, and Tamer Özsu. The ubiquity of large graphs and surprising challenges of graph processing. *VLDB Journal*, 11(4), 2023.
- Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast Word-Piece tokenization, In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. <https://aclanthology.org/2021.emnlp-main.160/>.
- Michael Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Samuel Madden, Elizabeth J. O’Neil, Patrick E. O’Neil, Alex Rasin, Nga Tran, and Stanley B. Zdonik. C-store: A column-oriented DBMS. In *31st International Conference on Very Large Data*, pages 553–564, 2005.
- Howard Turtle and James Flood. Query evaluation: Strategies and optimizations. *Information Processing & Management*, 31(6):831–850, 1995.
- Ellen M. Voorhees and Donna Harman. Overview of the seventh Text REtrieval Conference (TREC-7). In *7th Text REtrieval Conference*, 1998.
- Hugh E. Williams and Justin Zobel. Compressing integers for fast file access. *The Computer Journal*, 42(3):193–201, 1999.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. RepBERT: Contextualized text embeddings for first-stage retrieval, *ArXiv preprint arXiv:2006.15498*, 2020. URL <https://arxiv.org/abs/2006.15498>.
- Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4), 2024.

Graph Embeddings to Empower Entity Retrieval

Emma J. Gerritse

*Radboud University
Nijmegen, The Netherlands*

EMMA.GERRITSE@RU.NL

Faegheh Hasibi

*Radboud University
Nijmegen, The Netherlands*

FAEGHEH.HASIBI@RU.NL

Arjen P. de Vries

*Radboud University
Nijmegen, The Netherlands*

ARJEN.DEVRIES@RU.NL

Editor: Negin Rahimi, Makoto Kato

Abstract

In this research, we investigate methods for entity retrieval using graph embeddings. While various methods have been proposed over the years, most utilize a single graph embedding and entity linking approach. This hinders our understanding of how different graph embedding and entity linking methods impact entity retrieval. To address this gap, we investigate the effects of three different categories of graph embedding techniques and five different entity linking methods. We perform a reranking of entities using the distance between the embeddings of annotated entities and the entities we wish to rerank. We conclude that the selection of both graph embeddings and entity linkers significantly impacts the effectiveness of entity retrieval. For graph embeddings, methods that incorporate both graph structure and textual descriptions of entities are the most effective. For entity linking, both precision and recall concerning concepts are important for optimal retrieval performance. Additionally, it is essential for the graph to encompass as many entities as possible.

Keywords: Entity retrieval, Knowledge Graph Embeddings, Word Embeddings

1 Introduction

A significant portion of user queries in web search and question answering explicitly or implicitly reference entities, with users either asking for some entity-related facts or looking for the page of a specific entity (Balog, 2018; Meij et al., 2014). Entities are a vital part of information retrieval research, and this has led to the development of datasets (Hasibi et al., 2017b) and methods for effective entity retrieval (Arabzadeh et al., 2024; Gerritse et al., 2022; Chatterjee and Dietz, 2022; Gerritse et al., 2020; Dietz, 2019; Garigliotti et al., 2019). Entities are commonly stored in a Knowledge Graph (KG), which connects them through various relationships. The entire knowledge graph can then be represented using graph embeddings, which capture the rich and human-curated information in low-dimensional vector spaces, representing the similarity and relations between different entities. Popularized by methods such as Trans-E (Bordes et al., 2013) and Wikipedia2Vec (Yamada et al., 2020), graph embeddings have proven to be useful for tasks like link prediction (Bordes et al., 2013) and text classification (Yamada and Shindo, 2019).

This study investigates the use of KG embeddings to enhance lexical entity retrieval models. While recent transformer-based entity retrieval models have demonstrated superior performance compared to KG-augmented lexical models (Chatterjee and Dietz, 2022; Gerritse et al., 2022), they still struggle with long-tail entities (Gerritse et al., 2022). Similarly, dense retrievers face challenges in generalizing to the entity retrieval task (Kamalloo et al., 2024), performing worse than or only on par with strong lexical retrieval models such as BM25F or GEEER (Gerritse et al., 2020). These findings are in line with recent studies that show LLMs struggle to answer factual questions about long tail entities, and Retrieval Augmented Generation (RAG) needs to be employed to fill this gap (Mallen et al., 2023; Soudani et al., 2024).

Various methods have been proposed to include graph embeddings for entity retrieval. Gerritse et al. (2020) and Nikolaev and Kotov (2020) introduce methods using the similarity between the graph embedding vectors to compute relevance between queries and entities. These fairly simple methods improve over the previously established state of the art. However, existing approaches for utilizing graph embeddings for entity retrieval typically employ a single entity linker and embedding methods. Therefore, the effect of different entity linking and graph embedding methods on overall entity retrieval performance has remained unexplored.

In this study, we aim to fill this gap and explore various aspects of utilizing knowledge graph embeddings for entity retrieval. We base our approach on the method proposed by Gerritse et al. (2020), an early work on the use of graph embeddings for entity retrieval that has been reproduced and extended in various setups (Oza and Dietz, 2023; Chatterjee and Dietz, 2022; Jafarzadeh et al., 2022; Daza et al., 2021). Additionally, it provides a simple framework for the fusion of graph embeddings and retrieval models, enabling us to understand the effect of different entity linking and embedding methods on the entity retrieval task.

We tackle the following research questions in this work:

RQ 1. *How do different entity linking methods and their specific properties affect graph embedding-empowered entity retrieval?* Different entity-linking methods have been introduced throughout the years. Some take into account all different scenarios that might be meant by a query, for example, considering all other meanings of homonyms and polysemes, while others consider the most probable interpretation of a query and map each entity mention to a single entity. Additionally, entity linking methods such as REL (Van Hulst et al., 2020) and Nordlys (Hasibi et al., 2017a) provide high precision of named entities like people, location, and organizations. In contrast, methods like SMAPH (Cornolti et al., 2018) and TagMe (Ferragina and Scaiella, 2010) are designed to annotate both named entities and general concepts, such as DEMOCRACY. We hypothesize that when using graph embeddings for entity retrieval, it is beneficial to include as many entity embeddings relevant to the query as possible, encompassing both concepts and name entities. To assess this hypothesis, we manually annotate queries of the DBpedia-Entity collection (Hasibi et al., 2017b), and compare different entity linkers using this ground truth. We find that, indeed, the best entity retrieval performance is achieved with entity linkers that annotate both concepts and name entities.

RQ 2. *Which graph embedding techniques work best for entity retrieval methods empowered by graph embeddings?* In addition to diverse approaches proposed for entity linking, numerous algorithms have been introduced for graph embeddings. This work classifies them into three distinct groups, namely skip-gram-based (Yamada et al., 2016), transition-based (Trouillon et al., 2016), and random-walk-based (Ristoski and Paulheim, 2016). Often, research in the field of information retrieval only utilizes one type of graph embedding without considering the other classes of methods. In this work, we compare these various methods. We find that Wikipedia2Vec, as a skip-gram-based method, has the highest performance of all methods, provided that much effort is made to correct missing entities. We also see a positive impact from including as many entities as possible in the graph embedding for entity retrieval, even if it significantly increases the graph size during embedding training.

RQ 3. *How does the structural information captured by the skip-gram-based graph embedding approach, Wikipedia2Vec, contribute to entity retrieval effectiveness?* To address this research question, we train two versions of Wikipedia2Vec embeddings, with and without link graph, and compare the obtained embeddings and retrieval results. Utilizing the cluster hypothesis (Rijsbergen, 1979), we show that a representation of the graph structure in the embeddings leads to better clusters and higher effectiveness of retrieval results. We further see that queries with correctly linked entities (by an entity linker) are helped the most, while queries with wrongly linked entities are helped the least.

This work is an extension of (Gerritse et al., 2020), which makes the following new contributions: (i) We provide a comprehensive investigation of different entity linking and graph embedding methods for entity retrieval, (ii) We provide a new set of entity annotations of the widely-used DBpedia-Entity collection that includes both concepts and named entities. All the resources developed in the course of this study are made publicly available at: <https://github.com/informagi/GEEER>.

2 Related Work

2.1 Word and Entity Embeddings

Distributional representations of language have been the object of study for many years in natural language processing (NLP), because of their promise to represent words not in isolation, but semantically, with their immediate context. Algorithms like Word2Vec (Mikolov et al., 2013b) and Glove (Pennington et al., 2014) construct a vector space of word domains where similar words are mapped together (based on their linguistic context). Word2Vec embeddings are extracted from neural networks that predict words based on their context (continuous bag of words) or that predict the context for a given word (skip-gram). These word-embedding representations have proven to be highly effective in various Information Retrieval (IR) and NLP tasks.

Word embeddings have been shown to improve effectiveness in document retrieval (Dehghani et al., 2017; Diaz et al., 2016). In (Diaz et al., 2016), locally trained word embeddings are used for query expansion. Here, queries are expanded with terms highly similar to the query, and it is shown that this method beats several other neural methods. In (Dehghani et al., 2017), embeddings are used to train a neural ranking model using weak supervision.

The authors use query embeddings and document embeddings to predict relevance between queries and documents when given BM25 scores as labels, outperforming BM25.

Word embeddings capture the immediate linguistic context of word occurrences. Going beyond the text itself, researchers across various research communities have proposed a vast number of Knowledge Graph (KG) embedding methods. A KG is a graph where the nodes represent entities, and the edges represent the relations between them. One of the earliest and most well-known KG embedding methods is TransE (Bordes et al., 2013), which falls under the transition-based category. The TransE method considers graph edges as (*head*, *label*, *tail*) triples, where *label* is the value of the edge. Under the objective that adding graph embedding vectors of the *head* and the *label* should result in the vector of the *tail*, these embeddings are learned by gradient descent. TransE has been influential but has proved to be ineffective for anti-symmetric relations present in the knowledge graph. This resulted in various proposals for KG embeddings that extend this method with additional objectives, such as DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and SimpleE (Kazemi and Poole, 2018).

ComplEx (Trouillon et al., 2016) is a robust and widely used graph embedding, which utilizes complex vectors instead of real vectors and is better suited to represent anti-symmetric relations. Several studies (Ruffinelli et al., 2020; Chekalina et al., 2022; Kochsiek et al., 2023) have demonstrated that ComplEx achieves competitive performance on the Wikidata5M dataset (Wang et al., 2021). In particular, Ruffinelli et al. (2020) performed parameter tuning of ComplEx on the Wikidata5M dataset¹, and showed that when training conditions are standardized, models such as TransE, ComplEx, and DistMult perform similarly.

Another class of KG embeddings that has been widely applied in downstream tasks (Oza and Dietz, 2023; Gerritse et al., 2022), without direct comparison with translation-based embeddings in the literature, is skip-gram-based approaches, such as Deepwalk (Perozzi et al., 2014) and RDF2Vec (Ristoski and Paulheim, 2016). Deepwalk (Perozzi et al., 2014) is a graph embedding method that expects non-labeled edges. It first randomly samples vertices from the graph as starting points, and then performs a random walk from each of these starting points. The vertices walked in these random walks can then be represented as sentences. After having created these sentences from the graph, Deepwalk uses these as input for a Word2Vec-based approach using the skip-gram method. RDF2Vec (Ristoski and Paulheim, 2016) extends Deepwalk to work on knowledge graphs, by not only using the vertices, but also the labels of the edges for creating the random walks.

Wikipedia2Vec (Yamada et al., 2016) is a skip-gram-based approach that applies graph embeddings to Wikipedia, creating embeddings that jointly capture link structure and text. The Wikipedia knowledge graph is a natural resource for training graph embeddings, considering that it represents entities in a graph of interlinked Wikipedia pages and their text. The method proposed in (Yamada et al., 2016) embeds words and entities in the same vector space using word and graph contexts. The word-word context is modeled using the Word2Vec approach, entity-entity context considers neighboring entities in the link graph, and word-entity context takes the words in the context of the anchor text that links to an

1. highlighted in the GitHub repository <https://github.com/uma-pi1/kge>, last accessed 20-05-2025

entity. The authors of Wikipedia2Vec demonstrate performance improvements on various NLP tasks, although they did not consider entity retrieval in their work.

Recent transformer-based methods utilize transformers to create vector representations of entities. KGT5 (Saxena et al., 2022) treats link prediction as a sequence to sequence (seq2seq) task using a T5 architecture. It uses the prediction of sequences in the form of “predict tail: **subject mention** | **relation mention** | ”, and is trained on facts in the KG with the objective of generating the true answer using teacher forcing. The method achieves competitive scores on link prediction on large datasets like Wikidata5M. This work is improved upon in KGT5-context (Kochsiek et al., 2023), by adding extra context in the sequences, yielding even higher scores on link prediction on Wikidata5M. Li et al. (2024) introduces MoCoKGC, which utilizes three different transformer-based encoders to create its embedding. It separately trains an entity-relation encoder, an entity encoder, and a momentum-entity encoder, which provides more negative samples and allows the gradual updating of entity encodings. In this work, we study three widely used and competitive KG embedding models with reasonable computational demands: ComplEx, RDF2Vec, and Wikipedia2Vec.

2.2 Entity Linking

Methods incorporating entity information in NLP or information retrieval rely on entity linkers to identify entity mentions in the query or document. Entity linking refers to the process of detecting all possible mentions of entities and linking them to the corresponding identifier in a certain knowledge base (Balog, 2018). In this study, we employ the following five entity linking methods to annotate queries:

In Hasibi et al. (2015), the difference between entity linking in queries compared to full texts is discussed. Entity linking for queries is divided into two different tasks: The first is semantic mapping, which is finding a list of ranked entities similar to the queries. The second is interpretation finding, which finds sets of linked entities, representing possible interpretations, which can then be used for machine-understanding of the query. This work then also discusses evaluation methods for both tasks.

TagMe (Ferragina and Scaiella, 2010) is an on-the-fly entity linker specializing in short texts. It works in a three-step pipeline, parsing the query to make a candidate list of entities, then disambiguating. Afterward, it prunes entities with a low link probability/coherence to the other candidates. The REL entity linker (Van Hulst et al., 2020; Joko and Hasibi, 2022) is a popular open-source scalable entity linking toolkit (Kamphuis et al., 2022) for annotating various types of texts (e.g., documents and conversations) with entities. REL detects mentions using Flair, an NLP library that supports Named Entity Recognition (NER). REL performs candidate selection based on Wikipedia2Vec embeddings, and entity disambiguation based on latent relations between entity mentions in the text.

A number of entity linkers are designed for annotating queries. Nordlys (Hasibi et al., 2017a) uses the method created by Hasibi et al. (2015), which employs a learning-to-rank (LTR) model with various textual and semantic similarity features. SMAPH (Cornolti et al., 2018) utilizes a so-called piggybacking approach that uses information from a web search engine to link entities. It does this by first using the text, which needs to be linked as a query, and building a set of candidate entities, which are then filtered by linking the candidate entities to the terms in the input query. ELQ (Li et al., 2020) is a fast end-to-end

entity linking model specialized for questions using bi-encoders. It performs both mention detection and entity linking in one pass. Its architecture encodes queries and entities, then scores candidate entities through the inner product with the entity vectors.

2.3 Entity Retrieval

Knowledge Graphs like Wikipedia enrich the representation of entities by modeling the relations between them. Methods for ad-hoc document retrieval, such as BM25, have been applied successfully to retrieve entity information from knowledge graphs. However, since knowledge bases are semi-structured resources, this structural information may be used as well, for example, by viewing entities as fielded documents extracted from the knowledge graph. BM25F (Robertson et al., 2009) is a fielded retrieval model, where term frequencies between different fields in documents are normalized to the length of each field. Another effective model for entity retrieval is the Fielded Sequential Dependence Model (FSDM) (Zhiltsov et al., 2015), which estimates the probability of relevance using information from single terms and bi-grams, normalized per field.

Linking entities mentioned in the query to the knowledge graph allows relationships encoded in the knowledge graph to improve the estimation of the relevance of candidate entities. Previous work has shown that entity linking can indeed help increase effectiveness of entity retrieval. In (Hasibi et al., 2016), for example, entity retrieval has been combined with entity linking to improve retrieval effectiveness over state-of-the-art methods like FSDM.

Liu et al. (2019) is one of the early works applying graph embeddings to entity retrieval, which demonstrates consistent, albeit modest, improvements. KEWER (Nikolaev and Kotov, 2020) is another work that introduces a method for retrieving entities based on graph embeddings. It learns embeddings for entities and words based on TransE and annotates queries using SMAPH to re-rank entities, which improved on the previous state of the art. Similarly, Gerritse et al. (2020) first annotate queries using TagMe, and then use Wikipedia2Vec embeddings to compute the similarity between queries and documents. This method also improves on the previous state of the art.

Following the popularity of transformer-based methods, various works have introduced combinations of transformers with graph embeddings. In (Oza and Dietz, 2023), the work of Gerritse et al. (2020) is extended to include transformer-based entity embeddings. Daza et al. (2021) introduces a BERT architecture combined with TransE graph embeddings to re-rank entities. After encoding the queries and entities, it uses their similarity as a query score. In (Gerritse et al., 2022), a cross-encoder method is introduced, based on E-BERT (Poerner et al., 2020). It introduces entity tokens in the input layer, of which the encodings are based on Wikipedia2Vec embeddings. In (Tran and Yates, 2022), the BERT encoding of a query is combined with the Wikipedia2Vec embeddings of the annotated entities, which are aggregated using clusters. Last, Chatterjee and Dietz (2022) construct a method for entities without entity embeddings, by identifying the most relevant top-level sections from a Wikipedia page, depending on the query. These sections are then used to train a BERT model to represent the entities, which, in turn, are used as features in an LTR model for entity retrieval. While existing approaches typically examine only a single graph embedding and entity linking method, this work explores the effect of various graph embeddings and entity linkers on entity retrieval.

3 Embedding Based Entity Retrieval

In this section, we describe the graph embedding and entity retrieval approaches used in this paper. We utilize three different graph embedding methods that are representative of the three types of graph embeddings. The first is Wikipedia2Vec, which combines graph-based and text-based structures and learns embeddings using skip-gram algorithms. Next is RDF2Vec, which is based on random walks. Finally, we use ComplEx, which represents transition-based graph embeddings. We conclude by describing the methodology of our graph embedding based on the re-ranking algorithm.

3.1 Wikipedia2Vec

Taking a knowledge graph as the input, Wikipedia2Vec (Yamada et al., 2016, 2020) extends the skip-gram variant of Word2Vec (Mikolov et al., 2013b,a) and learns word and entity embeddings jointly for the Wikipedia knowledge graph. The objective function of this model is composed of three components. The first component infers optimal embeddings for words W in the corpus. Given a sequence of words $w_1 w_2 \dots w_T$ and a context window of size c , the word-based objective function is:

$$\mathcal{L}_w = \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log \frac{\exp(\mathbf{V}_{w_t}^T \mathbf{U}_{w_{t+j}})}{\sum_{w \in W} \exp(\mathbf{V}_{w_t}^T \mathbf{U}_w)}, \quad (1)$$

where matrices \mathbf{U} and \mathbf{V} represent the input and output vector representations, deriving the final embeddings from matrix \mathbf{V} .

The two other components of the objective function take the knowledge graph into account. The first considers a link-based measure estimated from the knowledge graph (i.e., Wikipedia). This measure captures the relatedness between entities in the knowledge graph, based on the similarity between their incoming links:

$$\mathcal{L}_e = \sum_{e_i \in \mathcal{E}} \sum_{e_o \in C_{e_i}, e_i \neq e_o} \log \frac{\exp(\mathbf{V}_{e_i}^T \mathbf{U}_{e_o})}{\sum_{e \in \mathcal{E}} \exp(\mathbf{V}_{e_i}^T \mathbf{U}_e)}, \quad (2)$$

where C_e denotes entities linked to an entity e , and \mathcal{E} represents all entities in the knowledge graph. The last addition to the objective function places similar entities and words near each other by considering the context of the anchor text. The intuition is the same as in Word2Vec, but here, words in the vicinity of the anchor text are used to predict the entity. Considering a knowledge graph with hyperlinks A and an entity e , the goal is to predict the context words of the entity:

$$\mathcal{L}_a = \sum_{e_i \in A} \sum_{w_o \in a(e_i)} \log \frac{\exp(\mathbf{V}_{e_i}^T \mathbf{U}_{w_o})}{\sum_{w \in W} \exp(\mathbf{V}_{e_i}^T \mathbf{U}_w)}, \quad (3)$$

where $a(e)$ gives the previous and next c words of the referent entity e .

These three components (word context, link structure, and anchor context) are then combined linearly into the following objective function:

$$\mathcal{L} = \mathcal{L}_w + \mathcal{L}_e + \mathcal{L}_a. \quad (4)$$

3.2 RDF2Vec

RDF2Vec (Ristoski and Paulheim, 2016) is a graph embedding method that takes a collection of triples represented in the Resource Description Framework (RDF) format and generates entity-relation sentences using random walks throughout the graph. These sentences are then used to compute Word2Vec-based embeddings. Suppose we have an RDF graph $G = (R, \mathcal{E})$ where R is a set of relations and \mathcal{E} is a set of entities. RDF2Vec generates all paths P_e in the RDF of depth d for all entities $e \in \mathcal{E}$. These walks are generated using a breadth-first algorithm. First, for a starting entity e_s , the algorithm explores the direct outgoing relations $R(e_s)$. Of these edges, a path $e_s \rightarrow r_i$ is randomly selected, where $r_i \in R(e_s)$. Then, for each previously explored node, the algorithm will visit the connected vertices. This generates a path $e_s \rightarrow r_i \rightarrow e_i$. This will continue until d iterations are reached, in which d is a hyperparameter. After this, all paths $\cup_{e \in \mathcal{E}} P_e$ are entered as sentences into the SkipGram model, as is seen in equation 1. This will result in an embedding for all $e \in \mathcal{E}$, thus leading to an embedding space for G .

3.3 ComplEx

The TransE triple-based graph embeddings have been introduced in (Bordes et al., 2013). The intuition behind TransE is to construct embeddings for head-relation-tail triples. Given triples $\langle e_h, r, e_t \rangle$, where $e_h, e_t \in \mathcal{E}$, $r \in R$, it minimalizes the vectors $\vec{e}_h, \vec{r}, \vec{e}_t \in \mathbb{R}^d$ with respect to the function $s(e_h, r, e_t) = \|\vec{e}_h + \vec{r} - \vec{e}_t\|$. This results in an embedding space where the tail of the triple can be found using vector addition, i.e., $\vec{e}_h + \vec{r} = \vec{e}_t$. However, TransE cannot embed one-to-many and many-to-many relations properly. Multiple algorithms expanding on TransE have been introduced, one of which is ComplEx (Trouillon et al., 2016), where entities are embedded in the complex field. The intuition for utilizing complex vectors is to capture the many anti-symmetric relations in knowledge graphs more effectively than TransE, and this makes the computations arguably simpler since they use only the Hermitian dot product.

ComplEx first initializes random embeddings $\vec{e}_h, \vec{r}, \vec{e}_t \in \mathbb{C}^d$ for $e_h, e_t \in \mathcal{E}$, $r \in R$. Then, for all training triples, the following scoring function $S(\vec{e}_h, \vec{r}, \vec{e}_t)$ is minimized:

$$S(e_h, r, e_t) = \text{Re}(\vec{e}_h^T \text{diag}(\vec{r}) \vec{e}_t)$$

where $\text{Re}()$ is the function that maps a complex vector to its real part, and $\text{diag}()$ the function that maps a vector of size d to a matrix with dimensions $d \times d$, that has the input vector as diagonal and all other elements are 0.

We use the setup for sampling and loss computation described in (Ruffinelli et al., 2020). During training time, both positive and negative examples are presented, where the negative samples are obtained by randomly perturbing one of the head, relation, or tail entities. The loss over these positive and negative samples is computed as follows: first, apply the sigmoid function on $S(e_h, r, e_t)$, then compute cross-entropy loss over the resulting value and the label of that triple.

3.4 Re-ranking Entities

In this section, we discuss the method of using these graph embeddings in the setting of entity retrieval. We propose a two-stage ranking model, where we first produce a ranking

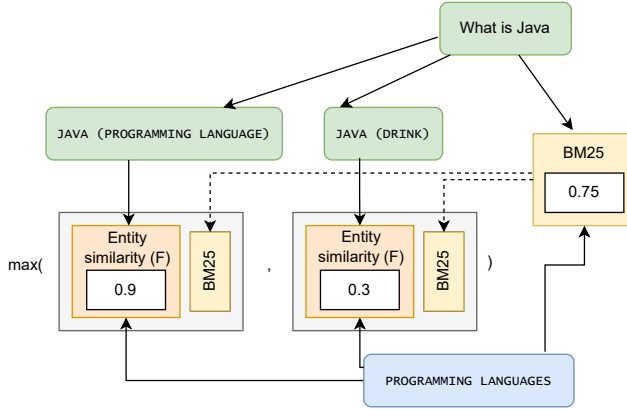


Figure 1: A diagram illustrating our re-ranking approach for an example query with multiple entity linking interpretations. The re-ranking method computes the similarity score (Eq. 5) between target entity (Programming Language) and linked entities in each query interpretation. These scores will then be combined with the score of the initial retriever (BM25 in this example). The maximum of these scores is considered as the final re-ranking score.

of candidate entities using a high-performing baseline entity retrieval model (see Section 2.3), and then use the graph embeddings to reorder these entities based on their similarity to the entities mentioned in the query, as measured in the derived graph embedding space.

Following the related work discussed in Section 2.2, we use the selected entity linkers to identify the entities mentioned in the query. Given input query q , we obtain a set of linked entities E_q and a confidence score $s(e_q)$ for each entity $e_q \in E_q$, which represents the strength of the relationship between the query and the linked entity. We then compute an embedding-based score for the linked entities of query q and entity e :

$$F(e, E_q) = \sum_{e_q \in E_q} s(e_q) \cdot \cos(\vec{e}, \vec{e_q}), \quad (5)$$

where $\vec{e}, \vec{e_q}$ denote the embeddings vectors for entities e and e_q , respectively.

The rationale for the scoring in Equation (5) is the hypothesis that relevant entities for a given query are situated close (in graph embedding space) to the query entities identified by the entity linker. Consider, for example, the query “Who is the daughter of Bill Clinton married to”, which is linked to three entities by entity linker: BILL CLINTON, DAUGHTER, and SAME-SEX MARRIAGE with a confidence of scores of 0.66, 0.13, and 0.21, respectively; i.e., $E_q = \{\text{BILL CLINTON, DAUGHTER, SAME-SEX MARRIAGE}\}$. The relevant entities for this query (according to the DBpedia-Entity V2 collection) are CHELSEA CLINTON, who is Bill Clinton’s daughter, and CLINTON FAMILY. Highly ranked entities thus exhibit strong similarity to these linked entities, with similarity to BILL CLINTON contributing more to the score than similarity to DAUGHTER or SAME-SEX MARRIAGE, due to the higher confidence score associated with BILL CLINTON. Given their close semantic and relational ties, it is

reasonable to expect relevant entities to be located near the linked entities in the embedding space, which confirms our intuition.

To produce our final score, we interpolate the embedding-based score computed using Equation (5) with the score of the baseline entity retrieval model $score_{other}$ used to produce the candidate entities in stage one:

$$score_{total}(e, q) = (1 - \lambda) \cdot score_{other}(e, q) + \lambda \cdot F(e, E_q) \quad \lambda \in [0, 1]. \quad (6)$$

When considering entity linking methods that return multiple interpretations of the query (i.e., multiple sets of E_q), we compute Equation (6) for every entity linking interpretation E_q^i and choose the interpretation that generates the highest score:

$$score_{total}(e, q) = \max_{E_q^i} \left((1 - \lambda) \cdot score_{other}(e, q) + \lambda \cdot F(e, E_q^i) \right). \quad (7)$$

Equation 7 extends Equation 6 in the following way: It first computes $score_{total}(e, q)$ for every entity linking interpretation of the query and then chooses the highest score as the final score. We select the highest similarity score, rather than other aggregation methods, because a specific entity is typically relevant to only one interpretation of a query. For example, consider the query “*What is Java?*”, which has three entity linking interpretations: $E_q^1 = \{\text{JAVA (ISLAND)}\}$, $E_q^2 = \{\text{JAVA (DRINK)}\}$, and $E_q^3 = \{\text{JAVA (PROGRAMMING LANGUAGE)}\}$. The similarity between the entity PROGRAMMING LANGUAGE and the third interpretation $E_q^3 = \{\text{JAVA (PROGRAMMING LANGUAGE)}\}$ would be high, while its similarity to the other two interpretations would be low. Since Programming Language is highly relevant to one specific sense of the query, taking the maximum score rather than an average is more appropriate. A visual representation of this can be seen in Figure 1.

4 Experimental Setup

4.1 Test collection

In our experiments, we use the standard entity retrieval collection, DBpedia-Entity V2 (Hasibi et al., 2017b). The collection consists of 467 queries and relevance assessments for 49280 query-entity pairs, where the entities are drawn from the DBpedia 2015-10 dump. The relevance assessments are graded values of 2, 1, and 0 for highly relevant, relevant, and non-relevant entities, respectively. The queries are categorized into 4 different groups: **SemSearch ES** consisting of short and ambiguous keyword queries (e.g., “*Nokia E73*”), **INEX-LD** containing IR-Style keyword queries (e.g., “*guitar chord minor*”), **ListSearch** consisting of queries seeking for a list of entities (e.g., “*States that border Oklahoma*”), and **QALD-2** containing entity-bearing natural language queries (e.g., “*Which country does the creator of Miffy come from*”). Following the baseline runs curated with the DBpedia-Entity V2 collection, we use the stopped version of queries provided by the dataset maintainers, where stop patterns like “which” and “who” are removed from the queries.

DBpedia-Entity V2 comprises queries from six benchmark evaluation campaigns, providing a diverse and heterogeneous set of queries. We point the reader to Oza and Dietz (2023), which shows that our entity ranking method also generalizes to the TREC CAR dataset (Dietz and Foley, 2019) expanded with automatic entity annotations.

4.2 Entity Annotation

We use two sets of human-annotated queries as ground truth, Webis annotations provided by (Kasturia et al., 2022), and Radboud annotations, a set created in this work.

Webis annotations. The annotations created in (Kasturia et al., 2022) present multiple scenario entity linking interpretations of the queries. For example, a query like “*java*” may have the island, the coffee, or the programming language as relevant documents, resulting in a set of interpretations where each interpretation relates to a different set of entities. Following (Hasibi et al., 2015), Webis annotations include only named entities (NEs).

Radboud annotations. To compare the influence of concepts to that of named entities (NE), we construct a new set of annotations of DBpedia-Entity queries, which we refer to as Radboud annotations. We ask expert annotators to annotate all queries with any entity that would help solve the information needs of the queries. This leads to annotations of *Named entities*, such as NOKIA and *Concepts* such as MOVIE PRODUCER in query “*who produced the film the ritual.*” Annotators use the INCEpTION annotation platform (Klie et al., 2018) to link mention spans to WikiData instances. We let one expert annotate the entire dataset and let 2 additional users annotate 50 randomly selected queries, similar to the setup in (Kasturia et al., 2022).

We compute the F-measure and Cohen’s Kappa for agreement, following (Deleger et al., 2012; Cui et al., 2022). Cohen’s Kappa score is computed on the token level (i.e., each word in a query is treated as a separate data point), a common strategy for handling multiple annotations per query. We found Kappa scores of 0.54 and 0.59, and F-scores of 0.51 and 0.57, which is sufficient for agreement (Cohen, 1960). An explanation for not having higher scores is that annotators might interpret a query differently. Since only one interpretation is asked, these annotations will then not agree. For example, the query “*sri lanka government gazette*” could be linked to the two entities SRI LANKA and THE SRI LANKA GAZETTE, but also to the non-overlapping entities GOVERNMENT OF SRI LANKA and GOVERNMENT GAZETTE, which have high similarity in the embedding space (and are thus likely to only have small differences when using the differences to these entity embeddings), but still have an overlap of 0.

4.3 Entity Linking

We employ the following entity linking methods: TagMe (Ferragina and Scaiella, 2010), REL (Van Hulst et al., 2020), Nordlys (Hasibi et al., 2017a), SMAPH (Cornolti et al., 2018), and ELQ (Li et al., 2020). To make a fair comparison, we use the default settings of all tools available, using the API if available.

- *TagME and REL*: We use their APIs with the default settings.
- *Nordlys*: Following (Nikolaev and Kotov, 2020), we use the Nordlys toolkit API with the Learning to Rank option.
- *SMAPH*: Following (Nikolaev and Kotov, 2020), we annotate using the *d4science* API, with the Google API as search engine.

- *ELQ*: We utilize the example script as listed on the official GitHub repository, using the default settings and preprocessing, most importantly casting all input to lowercase and using the *elq wiki large* model.

4.4 Embedding Training

Wikipedia2Vec. Wikipedia2Vec provides pre-trained embeddings. These embeddings, however, are not available for all entities in Wikipedia; e.g., 25% of the assessed entities in DBpedia-Entity V2 collection have no pre-trained embedding. The reasons for these missing embeddings are two-fold: (i) “rare” entities were excluded from the training data, and, (ii) entity identifiers evolve over time, resulting in entity mismatches with those in the DBpedia-Entity collection.

For training new graph embeddings, we used the Wikipedia 2019-07 dump, which is also compatible with recent entity linkers. We address the entity mismatch problem by identifying the entities that have been renamed in the new Wikipedia dump. Some of these entities were obtained using the redirect API of Wikipedia.² Others were found by matching the Wikipedia page IDs of the two Wikipedia dumps. The page IDs of Wikipedia 2019-07 were available on the Wikipedia website. For the dump where DBpedia-Entity is based on, however, these IDs are not available anymore; we obtained them from the Nordlys package Hasibi et al. (2017a).

To avoid excluding rare entities and generate embeddings for a wide range of entities, we changed several Wikipedia2Vec settings. The two settings that resulted in the highest coverage of entities are: (i) minimum number of times an entity appears as a link in Wikipedia, (ii) whether to include or exclude disambiguation pages. Table 1 shows the effect of these settings on the number of missing entities; specifically the number of entities that are assessed in the DBpedia-Entity collection, but have missing embeddings. We categorize these missing entities into two groups:

- *No-page*: Entities without any pages, i.e., although they have an identifier, there is no actual Wikipedia page with that exact identifier name. These entities were neither found by the Wikipedia redirect API nor could they be matched by their page IDs.
- *No-emb*: Entities that could be found by their identifiers, but were not included in the Wikipedia2Vec embeddings.

The first line in Table 1 corresponds to the default setting of Wikipedia2Vec, which covers only 75% of assessed entities in the DBpedia-Entity collection. When considering all entities in the knowledge graph, this setting discards an even larger number of entities, which is not an ideal setup for entity ranking. By choosing the right settings (the last line of Table 1), we increased the coverage of entities to 97.6%.

To make the comparison fair to the other graph embedding methods, we also use the Wikipedia 2015-10 dump, the version on which DBpedia 2015-10 was based.

RDF2vec. RDF2Vec and ComplEx are trained using the DBpedia link graph as input. DBpedia consists of over 20 possible input files, some containing more relevant information

2. <https://wikipedia.readthedocs.io/en/latest/>

Settings	No-emb	No-page	Total
min-entity-count = 5, disambiguation = False	9640	608	10248
min-entity-count = 1, disambiguation = False	1220	398	1618
min-entity-count = 1, disambiguation = True	1220	377	1597
min-entity-count = 0, disambiguation = False	724	380	1104
min-entity-count = 0, disambiguation = True	724	333	1057

Table 1: Missing entities with different settings.

for retrieval-oriented graph embeddings than others. For training RDF2vec and ComplEx, we use the following files from DBpedia:

- Disambiguations: Links extracted from Wikipedia Disambiguation pages, which are the Wikipedia pages that redirect a user when there are multiple entities with the same name.
- Infobox properties: Information which was extracted from the Wikipedia Infoboxes.
- Instance types: Triples of the form object, RDF type, and class from the mapping-based extraction.
- Instance types transitive: Transitive RDF type-class based on the ontology.
- Mapping-based objects: Mapping-based statements with object values.
- Transitive redirects: Transitively resolved redirects between articles in Wikipedia.
- Pagelinks: Contains internal links between entities on DBpedia, created based on the internal links between Wikipedia pages.

In the original RDF2Vec paper, the files used for finetuning are as listed above, except for the ‘Pagelinks’ file. Including this file increases the graph size from 11GB to 35GB, which in turn increases the training time and the final embedding output size. However, excluding this file results in missing a substantial number of entities, as seen in Table 2. Since several triples in the Pagelinks files appear to include alternatives for semantically similar entities, incorporating these nodes and edges could potentially dilute the effect of fine-tuning. We hypothesize that this occurs because entities seen sparsely during fine-tuning tend to have lower-quality representations in the embedding space. Distributing such an entity across multiple semantically similar entities may further degrade the quality of embeddings. As a resolution, we include results with and without the Pagelinks file.

For finetuning RDF2Vec, we use the jRDF2VEC package (Ristoski and Paulheim, 2016). We finetune using the default settings of the package, which is *walk depth* of 4, *numberOfWalks* of 100 per entity, *walkGenerationMode* as Random, *Dimension size* of 200 and *Number of Epochs* of 5.

Embedding	#Missing entities
Wikipedia2Vec 2019	36326
Wikipedia2Vec 2015	14124
ComplEX	25512
ComplEX Pagelinks	58
RDF2Vec	31782
RDF2Vec Pagelinks	75

Table 2: Number of missing entities with different settings of graph embeddings.

ComplEX. For ComplEX, we use the same training files as for RDF2Vec, reusing the same setup with and without the Pagelinks file. We use the LibKGE (Broscheit et al., 2020) package and the same configuration as used with the Wikipedia5M dataset, which has similar properties to DBpedia.

Table 2 shows the number of missing entity embeddings per embedding type. This is after using the DBpedia redirect file to solve redirects. We can see that, even when using DBpedia 2015-10 and Wikipedia 2015-10, many files are still missing in the eventual embedding space, which is bound to hurt results for re-ranking.

4.5 Evaluation metrics

Entity Linking. Given that Webis annotations contain multiple entity linking scenarios and Radboud annotations contain only one scenario, we need an evaluation method to accommodate both formats. One could choose, for example, evaluating exclusively the best possible scenario, the average across all scenarios, or the union of all scenarios as a ground truth. However, using these strategies, comparing queries with a difference in the number of scenarios presents a challenging task. We, therefore, use the lean evaluation metrics introduced by Hasibi et al. (2015).

Suppose $\hat{I} = \{\hat{E}_1, \dots, \hat{E}_m\}$ denote the query interpretations for the ground truth, and $I = \{E_1, \dots, E_n\}$ the interpretation returned by the system. Here, each \hat{E}_i , E_i is a set of entities, and thus \hat{I}, I are sets of sets of entities. Let $\hat{E} = \bigcup \hat{E}_i$ and $E = \bigcup E_i$. We first define the precision and recall based on the different interpretations, which we call P_{int} and R_{int} :

$$P_{int} = \begin{cases} \frac{|\hat{I} \cap I|}{|I|}, & I \neq \emptyset \\ 1, & I = \emptyset, \hat{I} = \emptyset \\ 0, & I = \emptyset, \hat{I} \neq \emptyset \end{cases} \quad R_{int} = \begin{cases} \frac{|\hat{I} \cap I|}{|\hat{I}|}, & \hat{I} \neq \emptyset \\ 1, & \hat{I} = \emptyset, I = \emptyset \\ 0, & \hat{I} = \emptyset, I \neq \emptyset \end{cases}$$

We then define the precision and recall based on all the entities linked, which we refer to as P_{ent} and R_{ent} :

$$P_{ent} = \begin{cases} \frac{|\hat{E} \cap E|}{|\hat{E}|}, & E \neq \emptyset \\ 1, & E = \emptyset, \hat{E} = \emptyset \\ 0, & E = \emptyset, \hat{E} \neq \emptyset \end{cases} \quad R_{ent} = \begin{cases} \frac{|\hat{E} \cap E|}{|E|}, & \hat{E} \neq \emptyset \\ 1, & \hat{E} = \emptyset, E = \emptyset \\ 0, & \hat{E} = \emptyset, E \neq \emptyset. \end{cases}$$

Lean precision and recall are then the combination between these two scores, being:

$$P_{lean} = \frac{P_{int} + P_{ent}}{2}, \quad R_{lean} = \frac{R_{int} + R_{ent}}{2}.$$

When given only one scenario for both system annotations and ground truth, we see that $P = P_{lean} = P_{int} = P_{ent}$ and $R = R_{lean} = R_{int} = R_{ent}$. Thus, lean evaluation can be utilized for Webis annotations, which encompass multiple scenarios, as well as Radboud annotations, which do not include multiple scenarios. Lean evaluation can be interpreted as the standard precision and recall for the latter.

Entity retrieval. We evaluate each combination of all methods and queries using the method used in (Hasibi et al., 2017b), which is the Normalized Discounted Cumulative Gain (NDCG) at ranks 10 and 100. We report on statistical significance for NDCG@10 and NDCG@100 using a two-sided t-test with p-value < 0.05 .

5 Results and Analysis

5.1 Entity Linking Results

To compare all different entity linking methods, we compute precision, recall, and F-measure for all entity linkers in Table 3, using lean evaluation described in Section 4.5. The line ‘combined’ indicates the score for the combination of all linked entities, which is the union of TagMe, REL, Nordlys, SMAPH, and ELQ. With Webis as ground truth, Nordlys has the highest recall and precision. Using our concept annotations as ground truth, ELQ has the highest precision, and TagMe has the highest recall. As seen in Table 3, there is a difference in what each of these entity linkers excels in, resulting in no single best method.

5.2 Entity Retrieval Results

Table 4 depicts entity retrieval results using different automatic entity linkers and human annotations; the breakdown of results by query type can be found in Table 8 of the Appendix. We use the Wikipedia2Vec embeddings from (Gerritse et al., 2020) and perform re-ranking with BM25F-CA. When using human annotations, Radboud annotations receive better results than the multiple scenarios used in the Webis annotations. Especially in the SemSearch, INEX, and ListSearch categories, we see an increase in both NDCG@10 and NDCG@100 using Radboud annotations. Since the Radboud annotations focuses on adding concept annotations, this indicates that concept entities are, in fact, an essential factor for entity retrieval using entity embeddings.

	#Linked entities	Webis (NEs)			Radboud (Concepts+NEs)		
		P_{lean}	R_{lean}	F_{lean}	P	R	F
TagMe	1186	0.479	0.598	0.532	0.704	0.814	0.755
REL	363	0.699	0.618	0.656	0.534	0.416	0.467
Nordlys	450	0.722	0.642	0.680	0.553	0.439	0.490
SMAPH	797	0.526	0.550	0.538	0.758	0.720	0.739
ELQ	647	0.606	0.587	0.597	0.787	0.701	0.741
Combined	1795	0.437	0.686	0.534	0.621	0.911	0.738

Table 3: Performance of different entity linkers against Webis (only NE) and Radboud (both NEs and concepts) annotations. Lean evaluation is used for multiple interpretations (Webis), and regular precision/recall is used for single interpretations (Radboud). ‘Combined’ refers to the union of all entities returned by all linkers.

	NDCG@10	NDCG@100
BM25F-CA	0.461	0.551
+ Wikipedia2Vec (w/ TagMe)	0.484 ^b	0.570 ^b
+ Wikipedia2Vec (w/ SMAPH)	0.483 ^b	0.570 ^b
+ Wikipedia2Vec (w/ Nordlys)	0.477 ^b	0.563 ^{bt}
+ Wikipedia2Vec (w/ REL)	0.472 ^{bt}	0.561 ^{bt}
+ Wikipedia2Vec (w/ ELQ)	0.489 ^{bnr}	0.573 ^{bnr}
+ Wikipedia2Vec (w/ Combined)	0.498 ^{bt snre}	0.58 ^{bt snre}
+ Wikipedia2Vec (w/ Webis)	0.475 ^{bea}	0.563 ^{bt sea}
+ Wikipedia2Vec (w/ Radboud)	0.492 ^{bsnrw}	0.577 ^{bnrw}

Table 4: Entity retrieval results on DBpedia-Entity V2 collection using different entity linkers and gold annotations. Wikipedia2Vec 2019 is used for re-ranking of BM25F-CA results. Superscripts denote statistically significant differences (better or worse) corresponding to the beginning letter of entity linkers’ names, BM25F-CA, and Webis.

The entity retrieval methods using TagMe, SMAPH and ELQ as entity linkers obtain the best results, with no overall significant differences between each other. These three entity linkers obtain the highest F-measure when using the Radboud annotations as ground truth. This answers our first research question **RQ1**: Entity-linking methods with a high F-measure with respect to both concepts and named entities are best suited for the entity retrieval method discussed in this paper.

Table 5 shows the results for different embedding algorithms using TagMe and Radboud annotations; the full table can be found in Table 9 of the Appendix. In the first block, denoted as ‘base’, we only rank using the entity similarity score per embedding type; i.e., using Equation 5 only, using TagMe as an entity linker. Next, we re-rank the BM25F-CA results, both using TagMe (as used in (Gerritse et al., 2020)) and Radboud annotations. We see that Wikipedia2Vec embeddings outperform ComplEx and RDF2Vec in base form, but not in the base form where many entities are missing. We also see that RDF2Vec and

	NDCG@10	NDCG@100
<i>Base</i>		
Wikipedia2Vec 2019	0.262 ¹	0.335 ¹
Wikipedia2Vec 2015	0.184 ¹²	0.278 ¹²
ComplEx	0.182 ¹²	0.216 ¹²³
ComplEx Pagelinks	0.204 ¹²³⁴	0.255 ¹²³⁴
RDF2Vec	0.188 ¹²⁵	0.23 ¹²³⁴⁵
RDF2Vec Pagelinks	0.195 ¹²	0.262 ¹²³⁴⁶
<i>TagMe</i>		
BM25F-CA	0.461	0.551
+ Wikipedia2Vec 2019	0.484 ¹	0.57 ¹
+ Wikipedia2Vec 2015	0.466 ²	0.559 ¹²
+ ComplEx	0.468 ¹²	0.558 ¹²
+ ComplEx Pagelinks	0.474 ¹²³⁴	0.564 ¹²³⁴
+ RDF2Vec	0.461 ²⁴⁵	0.554 ²³⁵
+ RDF2Vec Pagelinks	0.467 ²⁵⁶	0.56 ¹²⁶
<i>Radboud annotations</i>		
BM25F-CA	0.461	0.551
+ Wikipedia2Vec 2019	0.492 ¹	0.576 ¹
+ Wikipedia2Vec 2015	0.473 ¹²	0.564 ¹²
+ ComplEx	0.467 ¹²	0.558 ¹²³
+ ComplEx Pagelinks	0.476 ¹²⁴	0.564 ¹²⁴
+ RDF2Vec	0.463 ²³⁵	0.556 ¹²³⁵
+ RDF2Vec Pagelinks	0.471 ¹²⁶	0.566 ¹²⁴⁶

Table 5: Entity retrieval results on DBpedia-Entity V2 collection using different graph embeddings. Superscripts denote statistically significant differences (better or worse) corresponding to that line of the table.

ComplEx perform significantly better when using the Pagelink pages, indicating that they are essential when re-ranking with entity retrieval. Last, we see that Radboud annotations yield higher scores than the TagMe annotations.

5.3 Embedding Analysis

To judge how suitable each graph embedding is for retrieval, we compare the embeddings of documents relevant to the same query, based on the cluster hypothesis (Rijsbergen, 1979). This states that documents relevant to the same query should cluster together in a higher dimensional space. Here, we consider the entity embedding of a document as its representation. Using this hypothesis, we compute each query’s coherence score as defined in (He, 2011), which measures the similarity between all pairs of documents relevant to the same query and returns the percentage of items with a similarity score higher than a

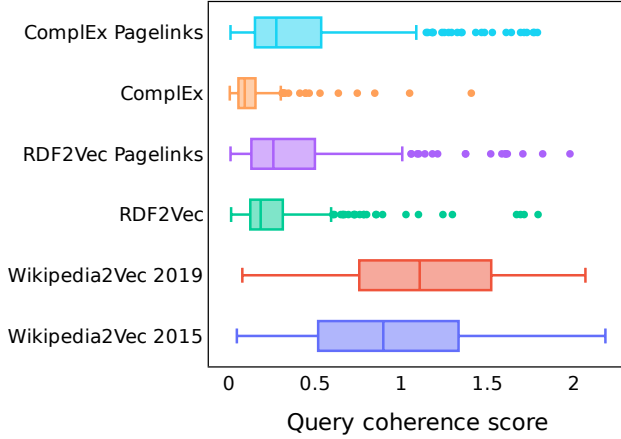


Figure 2: Query coherence score of three different graph embedding algorithms. Higher coherence score is better. Wikipedia2Vec has higher coherence scores compared to RDF2Vec and ComplEx.

threshold. Formally, given a document set D , the coherence score is computed as:

$$Co(D) = \frac{\sum_{i \neq j \in 1, \dots, M} \delta(d_i, d_j)}{\frac{1}{2}M(M-1)}, \quad (8)$$

where M is total number of documents and the δ function for each document pair d_i and d_j is defined as:

$$\delta(d_i, d_j) = \begin{cases} 1, & \text{if } sim(d_i, d_j) \geq \tau \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In Figure 2, we see coherence scores for the different entity embedding methods used in this paper, with a threshold of $\tau = 0.7$. This threshold is selected via grid search as the highest value at which none of the box plots had a first quartile equal to 0. For RDF2Vec and ComplEx, we include both versions with and without pagelinks. We compute the coherence score on the 295 queries with at least ten relevant entities. The higher the query coherence scores are, the better the distance between embeddings should be able to represent relevance for documents to the same query. We see that Wikipedia2Vec leads to the highest coherence scores. Interestingly enough, we see that the 2019 version of Wikipedia has a higher average coherence score than the 2015 version, even though more entities seem to be missing. A reason for this could be that the quality of Wikipedia has improved over the years, with more new pages and additional text added, resulting in a better coherence score. Besides, we see that for both ComplEx and RDF2Vec, the additional page links improve the coherence scores.

We now answer our second research question **RQ2**: Wikipedia2Vec leads to the best results for the entity retrieval method discussed in this paper. However, it is essential to invest in solving the missing entities for optimal performance. For ComplEx and RDF2Vec, including a sufficiently large number of entities in the graph is considerably important.

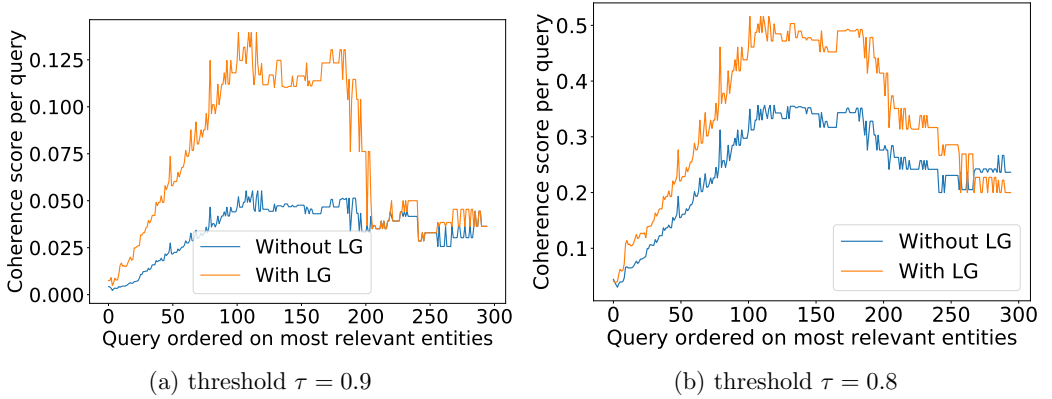


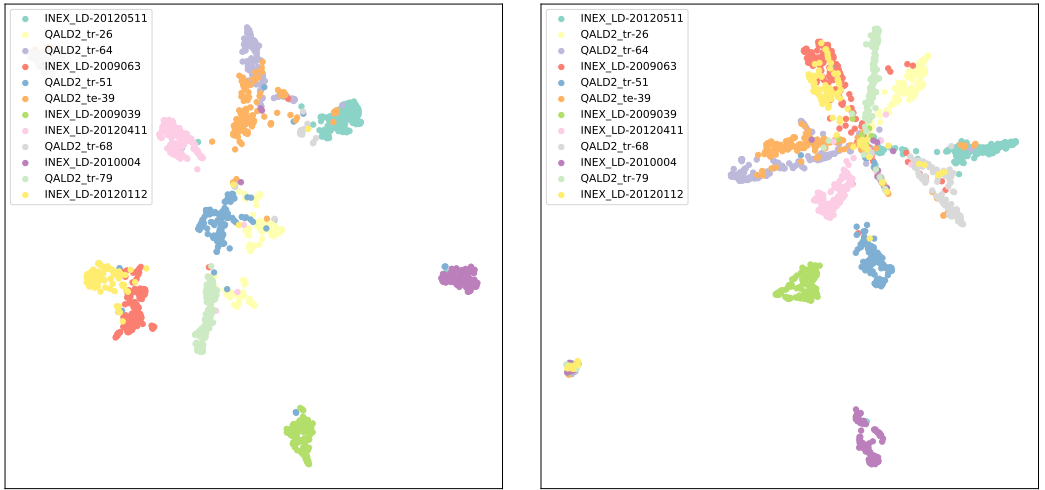
Figure 3: Coherence score of all relevant entities per query, computed for the Wikipedia2Vec embeddings without and with link graph. The queries are ordered by the number of their relevant entities in the x-axis.

5.4 Wikipedia2Vec Embedding Analysis

We empirically showed that Wikipedia2Vec graph embeddings yield better performance compared to other embeddings. To analyze why Wikipedia2Vec graph embeddings are beneficial for entity retrieval models, we conduct a set of experiments and investigate how the graph structure captured by Wikipedia2Vec embeddings improves effectiveness. Specifically, we trained two versions of Wikipedia2Vec embeddings: with and without link graph; i.e., using Eq. (4) with and without the \mathcal{L}_e component.

Figure 3 shows the coherence scores for all queries in our collection. Each point represents the coherence score of all relevant entities (according to the qrels) for a query. We considered only queries with more than 10 relevant entities, ensuring the clusters were sufficiently large to yield meaningful scores. Queries are sorted on the x-axis by the number of relevant entities. The plots clearly show that the coherence score for graph-based entity embeddings is higher than for context-only ones. Based on these performance improvements, we conclude that adding the graph structure results in embeddings that are more suitable for entity-oriented tasks.

Figure 4 helps to visually understand how clusters of entities differ for the two embedding variations. The data points correspond to the entities with a relevance grade higher than 0, for 12 queries with 100–200 relevant entities in the ground truth data. We use Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) to reduce the dimensions of the embeddings from 100 to two and plot the projected entities for each query. In Figure 4b, most of the clusters are overlapping in a star-like shape, while in Figure 4a, the clusters are more separated, and the ones with similar search intents are close to each other; e.g., queries QALD2_te-39 and QALD2_tr-64 (which are both about companies), or INEX_LD-20120112 and INEX_LD-2009063 (which are both about war) are situated next to each other. These analyses support the answer that we formulate for our third research question **RQ 3**: The graph structure, combined with the textual representation of entities incorporated in Wikipedia2Vec graph embeddings, plays an important role in improving the



(a) Embeddings with link graph

(b) Embeddings without link graph

Figure 4: UMAP visualization of entity embeddings for a subset of queries. Color codes correspond to the relevant entities per query. Queries per code are listed in Table 10 of the Appendix. Default settings of UMAP in python were used.

cluster quality of the representation of the entities and explains the enhanced effectiveness of retrieval results.

5.5 Query Analysis

Next, we analyze queries that are helped and hurt the most by our embedding-based method. Table 6 shows six queries that are affected the most by BM25F-CA+ Wikipedia2Vec compared to BM25F-CA, with respect to NDCG@100. Each of the three queries with the highest gains is linked to at least one relevant entity (according to the assessments). The losses can be attributed to various sources of errors. For the query “*spring shoe canada*”, the only relevant entity belongs to the 2.4% of entities that have no embedding (cf. §4.4). Query “*vietnam war movie*” is linked to entities VIETNAM WAR and WAR FILM, with confidence scores of 0.7 and 0.2, respectively. This emphasizes Vietnam war facts instead of its movies, and could be resolved by improving the accuracy of the entity linker and/or employing a re-ranking approach that is more robust to linking errors. The query “*mr rourke fantasy island*” is linked to a wrong entity due to a spelling mistake, which emphasizes the importance of the quality of the entity linker.

To further understand the difference between the two versions of the embeddings at the query-level, we selected the queries with the highest and lowest gain in NDCG@100 (i.e., comparing BM25F-CA + Wikipedia2Vec and BM25F-CA + Wikipedia2Vec (no graph)). For the query “*Which instruments did John Lennon play?*”, the two linked entities (with the highest confidence score) are JOHN LENNON and MUSICAL INSTRUMENTS. Their closest entity in graph embedding space is JOHN LENNON’S MUSICAL INSTRUMENTS, relevant to

Query	Gain in NDCG	
	@10	@100
st paul saints	0.716	0.482
continents in the world	0.319	0.362
What did Bruce Carver die from?	0.307	0.307
spring shoes canada	-0.286	-0.286
vietnam war movie	-0.470	-0.240
mr rourke fantasy island	-0.300	-0.307

Table 6: Top queries with the highest gains and losses in NDCG at cut-offs 10 and 100, BM25F-CA + Wikipedia2Vec vs. BM25F-CA.

Query	Gain in NDCG	
	@10	@100
What did Bruce Carver die from?	0.307	0.307
Which other weapons did the designer of the Uzi develop?	0.236	0.248
Which instruments did John Lennon play?	0.154	0.200
Companies that John Hennessey serves on the board of	-0.173	-0.173
Which European countries have a constitutional monarchy?	-0.101	-0.197
vietnam war movie	-0.276	-0.222

Table 7: Top queries with the highest gains and losses in NDCG at cut-offs 10 and 100, BM25F-CA + Wikipedia2Vec vs. BM25F-CA + Wikipedia2Vec (no graph).

the query. This entity, however, is not among the most similar entities when we consider the context-only case. For the other queries in Table 7, the effect is similar but less prominent than in the BM25F-CA and BM25F-CA + Wikipedia2Vec case, probably due to the lower value of λ .

6 Conclusion

In this paper, we investigated the use of different types of entity embeddings and different types of entity linkers for entity retrieval. We trained entity embeddings using Wikipedia2Vec, ComplEx, and RDF2Vec, combined these with state-of-the-art entity ranking models, and found empirically that using graph embeddings leads to increased effectiveness of entity retrieval.

We analyzed the effect of different entity linkers and concluded that the most suitable entity linkers are SMAPH, TagMe, and ELQ, annotating both named entities and concepts. When evaluating with the baselines of annotated entities with two different philosophies, multiple scenarios compared to named entities and concepts, we found that SMAPH, TagMe, and ELQ align the most with the annotations focused on named entities and concepts, thus confirming our conclusion that annotated concept are important for retrieval.

We then compared three classes of graph embedding methods, Wikipedia2Vec, RDF2Vec, and ComplEx, and found that first, having as many different entities in the graph embedding will give the best performance, even if they might be redundant. Second, Wikipedia2Vec performs best in all categories, provided that effort is put into solving as many entities linked to an entity without embedding as possible. Wikipedia2Vec has the highest cluster similarity score, confirming that Wikipedia2Vec is a highly suitable method for performing entity retrieval.

We conclude that enriching entity retrieval methods with entity embeddings is valuable, efficient, and effective. The choice of entity linker, graph embedding method, and effort to find missing entities are integral to the method’s performance. For future work, we would like to evaluate how these different graph embedding methods influence more modern Transformer-based entity retrieval methods, as well as how well these methods can be adapted to work on domain-specific entities or sparser knowledge graphs.

Acknowledgments and Disclosure of Funding

This research is supported by the Dutch Research Council (NWO) under project numbers NWA.1389.20.183 (LESSEN) and the EU’s Horizon Europe program under grant No. 101070014 (OpenWebSearch.EU, <https://doi.org/10.3030/101070014>). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

- Negar Arabzadeh, Amin Bigdeli, and Ebrahim Bagheri. Laque: Enabling entity search at scale. In *Advances in Information Retrieval*, pages 270–285, 2024.
- Krisztian Balog. Entity-oriented search, *Springer*, 2018.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 2787–2795. ACM, 2013.
- Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, 2020.
- Shubham Chatterjee and Laura Dietz. BERT-ER: Query-specific BERT entity representations for entity ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’22, page 1466–1477. Association for Computing Machinery, 2022.
- Viktoriia Chekalina, Anton Razzhigaev, Albert Sayapin, Evgeny Frolov, and Alexander Panchenko. MEKER: Memory efficient knowledge embedding representation for link

- prediction and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 355–365. Association for Computational Linguistics, 2022.
- Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze. Smaph: A piggyback approach for entity-linking in web queries. *ACM Transactions on Information Systems*, 37(1), 2018. ISSN 1046-8188.
- Wen Cui, Leanne Rolston, Marilyn Walker, and Beth Ann Hockey. Openel: An annotated corpus for entity linking and discourse in open domain dialogue. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2245–2256, 2022.
- Daniel Daza, Michael Cochez, and Paul Groth. Inductive entity representations from text via link prediction. In *Proceedings of the Web Conference, WWW’21*, page 798–808, 2021.
- Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. Neural ranking models with weak supervision. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74, 2017.
- Louise Deleger, Qi Li, Todd Lingren, Megan Kaiser, Katalin Molnar, Laura Stoutenborough, Michal Kouril, Keith Marsolo, Imre Solti, et al. Building gold standard corpora for medical natural language processing tasks. In *AMIA Annual Symposium Proceedings*, volume 2012, page 144. American Medical Informatics Association, 2012.
- Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 367–377, 2016.
- Laura Dietz. Ent rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR’19, page 215–224, 2019.
- Laura Dietz and John Foley. Trec car y3: Complex answer retrieval overview. In *Proceedings of Text REtrieval Conference (TREC)*, 2019.
- Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of the 19th ACM international IW3C2 on Information and Knowledge Management*, pages 1625–1628. ACM, 2010.
- Darío Garigliotti, Faegheh Hasibi, and Krisztian Balog. Identifying and exploiting target entity type information for ad hoc entity retrieval. *Information Retrieval Journal*, 22(3): 285–323, 2019.
- Emma Gerritse, Faegheh Hasibi, and Arjen De Vries. Graph-embedding empowered entity retrieval. In *Proceedings of the European Conference on Information Retrieval, ECIR ’20*, pages 97–110, 2020.

- Emma Gerritse, Faegheh Hasibi, and Arjen De Vries. Entity-aware Transformers for Entity Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, 2022.
- Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Entity linking in queries: Tasks and evaluation. In *Proceedings of ACM SIGIR International Conference on the Theory of Information Retrieval*, ICTIR '15, pages 171–180, 2015.
- Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 209–218. ACM, 2016.
- Faegheh Hasibi, Krisztian Balog, Darío Garigliotti, and Shuo Zhang. Nordlys: A toolkit for entity-oriented and semantic search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1289–1292. ACM, 2017a.
- Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. DBpedia-Entity V2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268. ACM, 2017b.
- Jiying He. *Exploring topic structure: Coherence, diversity and relatedness*. PhD thesis, University of Amsterdam, 2011.
- Parastoo Jafarzadeh, Zahra Amirmahani, and Faezeh Ensan. Learning to rank knowledge subgraph nodes for entity retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2519–2523, 2022.
- Hideaki Joko and Faegheh Hasibi. Personal entity, concept, and named entity linking in conversations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4099–4103. Association for Computing Machinery, 2022.
- Ehsan Kamaloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. Resources for brewing beer: Reproducible reference models and statistical analyses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1431–1440. Association for Computing Machinery, 2024. ISBN 9798400704314.
- Chris Kamphuis, Faegheh Hasibi, Jimmy Lin, and Arjen P. de Vries. REBL: entity linking at scale (prototype). In Omar Alonso, Ricardo Baeza-Yates, Tracy Holloway King, and Gianmaria Silvello, editors, *Proceedings of the third International Conference on Design of Experimental Search & Information REtrieval Systems (DESIREs)*, volume 3480, pages 68–75, 2022.
- Vaibhav Kasturia, Marcel Gohsen, and Matthias Hagen. Query Interpretations from Entity-Linked Segmentations. In *15th ACM International Conference on Web Search and Data Mining (WSDM 2022)*, pages 449–457, 2022.

- Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 4289–4300, 2018.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, 2018.
- Adrian Kochsiek, Apoorv Saxena, Inderjeet Nair, and Rainer Gemulla. Friendly neighbors: Contextualized sequence-to-sequence link prediction. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 131–138. Association for Computational Linguistics, 2023.
- Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6433–6441. Association for Computational Linguistics, 2020.
- Qingyang Li, Yanru Zhong, and Yuchu Qin. MoCoKGC: Momentum contrast entity encoding for knowledge graph completion. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14940–14952. Association for Computational Linguistics, 2024.
- Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Explore entity embedding effectiveness in entity retrieval. *arXiv preprint arXiv:1908.10554*, 2019.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 9802–9822. Association for Computational Linguistics, 2023.
- Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29): 861, 2018.
- Edgar Meij, Krisztian Balog, and Daan Odijk. Entity linking and retrieval for semantic search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM ’14, page 683–684. Association for Computing Machinery, 2014.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR*, pages 1–12, 2013a.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013b.

- Fedor Nikolaev and Alexander Kotov. Joint word and entity embeddings for entity retrieval from a knowledge graph. In *Proceedings of the European Conference on Information Retrieval*, ECIR '20, pages 141–155, 2020.
- Pooja Oza and Laura Dietz. Entity embeddings for entity ranking: A replicability study. In *Proceedings of the 45th European Conference on Information Retrieval*, ECIR '23, pages 117–131, 2023.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543. ACL, 2014.
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international Conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics*, ELMNLP '20, pages 803–818, 2020.
- C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Butterworths, 1979.
- Petar Ristoski and Heiko Paulheim. RDF2Vec: RDF graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.
- Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
- Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN teach an old dog new tricks! On training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020.
- Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828. Association for Computational Linguistics, 2022.
- Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2024, page 12–22. Association for Computing Machinery, 2024.
- Hai Dang Tran and Andrew Yates. Dense retrieval with entity views. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1955–1964. Association for Computing Machinery, 2022.

- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080, 2016.
- Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 2020.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.
- Ikuya Yamada and Hiroyuki Shindo. Neural attentive bag-of-entities model for text classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 563–573. ACL, 2019.
- Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *The SIGNLL Conference on Computational Natural Language Learning*, 2016.
- Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics, 2020.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
- Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–262. ACM, 2015.

Appendix A. Extra Results

	SemSearch		INEX-LD		ListSearch		QALD-2	
	@10	@100	@10	@100	@10	@100	@10	@100
TagMe	0.661 ^b	0.738 ^b	0.464 ^b	0.552 ^b	0.447 ^b	0.532 ^b	0.387 ^b	0.479 ^b
SMAPH	0.661 ^b	0.734	0.455	0.546 ^b	0.448 ^b	0.534 ^b	0.389 ^b	0.483 ^b
Nordlys	0.645	0.719 ^t	0.451 ^b	0.537 ^{bt}	0.444 ^b	0.529 ^b	0.387 ^b	0.481 ^b
REL	0.647 ^b	0.732 ^b	0.451	0.536 ^t	0.436	0.523 ^{bs}	0.376 ^{bs}	0.472 ^{bsn}
ELQ	0.645	0.729	0.474 ^{bnr}	0.554 ^{bnr}	0.455 ^b	0.537 ^b	0.402 ^{btr}	0.488 ^{br}
Combined	0.667 ^{be}	0.738 ⁿ	0.476 ^{btsnr}	0.56 ^{btsnr}	0.468 ^{btsnr}	0.549 ^{btsnr}	0.402 ^{btsnr}	0.492 ^{btsnr}
Webis	0.63 ^{tsa}	0.711 ^{tsra}	0.452 ^{ea}	0.537 ^{tea}	0.439 ^a	0.532 ^{bra}	0.396 ^{bnr}	0.488 ^{br}
Radboud	0.651	0.727	0.478 ^{btsnrw}	0.56 ^{bsnrw}	0.462 ^{bsnrw}	0.543 ^{bsnrw}	0.399 ^{br}	0.495 ^{btsnr}

Table 8: Breakdown per query type of entity retrieval results using different entity linkers and gold annotations. Wikipedia2Vec 2019 is used for re-ranking of BM25F-CA results. Superscripts denote statistically significant differences corresponding to the beginning letter of entity linkers’ names, BM25F-CA, and Webis.

	SemSearch		INEX-LD		ListSearch		QALD-2	
	@10	@100	@10	@100	@10	@100	@10	@100
<i>Base</i>								
Wikipedia2Vec 2019	0.417	0.478	0.217	0.286	0.211	0.302	0.212	0.282
Wikipedia2Vec 2015	0.249 ¹	0.345 ¹	0.152 ¹	0.24 ¹	0.153 ¹	0.26 ¹	0.181 ¹	0.265
ComplEx	0.309 ¹²	0.322 ¹	0.136 ¹	0.17 ¹²	0.139 ¹	0.178 ¹²	0.148 ¹²	0.193 ¹²
ComplEx Pagelinks	0.32 ¹²	0.371 ¹³	0.148 ¹	0.187 ¹²	0.183 ¹³	0.239 ¹³	0.168 ¹	0.224 ¹²³
RDF2Vec	0.317 ¹²	0.346 ¹	0.159 ¹	0.182 ¹²	0.128 ¹⁴	0.175 ¹²⁴	0.154 ¹	0.216 ¹²
RDF2Vec Pagelinks	0.308 ¹²	0.376 ¹³⁵	0.179 ¹³	0.239 ¹³⁴⁵	0.142 ¹⁴	0.216 ¹²³⁵	0.158 ¹	0.225 ¹²³
<i>TagMe</i>								
BM25F-CA	0.628	0.720	0.439	0.530	0.425	0.511	0.369	0.461
+ Wikipedia2Vec 2019	0.661¹	0.738¹	0.464¹	0.552¹	0.447¹	0.532¹	0.387¹	0.479¹
+ Wikipedia2Vec 2015	0.633 ²	0.724 ²	0.443 ²	0.539 ¹²	0.434 ²	0.525 ¹²	0.373 ²	0.467 ²
+ ComplEx	0.64	0.727	0.441 ²	0.53 ²	0.433 ²	0.523 ¹²	0.376 ²	0.47 ¹
+ ComplEx Pagelinks	0.653 ¹³	0.732 ¹	0.445 ²	0.537 ²	0.443 ¹	0.532 ¹⁴	0.377	0.474 ¹
+ RDF2Vec	0.633 ²⁵	0.722 ²⁵	0.429 ²³⁴⁵	0.53 ²³	0.428 ²⁵	0.519 ¹²⁵	0.374 ²	0.465 ²
+ RDF2Vec Pagelinks	0.636 ²⁵	0.727	0.436 ²	0.539 ¹²⁶	0.432 ²	0.52 ¹²⁵	0.381 ¹	0.474 ¹³⁶
<i>Concepts</i>								
BM25F-CA	0.628	0.720	0.439	0.530	0.425	0.511	0.369	0.461
+ Wikipedia2Vec 2019	0.652	0.73	0.48 ¹	0.558 ¹	0.46 ¹	0.543 ¹	0.398 ¹	0.491 ¹
+ Wikipedia2Vec 2015	0.632 ²	0.721	0.449 ²	0.541 ¹²	0.442 ¹²	0.531 ¹²	0.387 ¹²	0.479 ¹²
+ ComplEx	0.635	0.723	0.444 ²	0.535 ²	0.434 ²	0.522 ²³	0.375 ²³	0.47 ¹²³
+ ComplEx Pagelinks	0.65 ¹⁴	0.727	0.449 ²	0.539 ¹²	0.44 ²	0.53 ¹²⁴	0.384 ¹	0.477 ¹²
+ RDF2Vec	0.637 ⁵	0.723	0.429 ²³⁴⁵	0.529 ²³⁵	0.43 ²³	0.52 ²³⁵	0.373 ²³⁵	0.47 ¹²³⁵
+ RDF2Vec Pagelinks	0.637	0.733 ³⁴⁶	0.443 ²⁶	0.543 ¹²⁴⁶	0.437 ²	0.526 ¹²	0.384 ¹²⁶	0.482 ¹⁴⁶

Table 9: Breakdown per query type of entity retrieval results using different graph embeddings. Superscripts denote statistically significant differences (better or worse) corresponding to that line of the table.

Appendix B. Queries

Query ID	Query text
INEX_LD-20120511	female rock singers
QALD2_tr-26	Which bridges are of the same type as the Manhattan Bridge?
QALD2_tr-64	Which software has been developed by organizations founded in California?
INEX_LD-2009063	D-Day normandy invasion
QALD2_tr-51	Give me all school types.
QALD2_te-39	Give me all companies in Munich.
INEX_LD-2009039	roman architecture
INEX_LD-20120411	bicycle sport races
QALD2_tr-68	Which actors were born in Germany?
INEX_LD-2010004	Indian food
QALD2_tr-79	Which airports are located in California, USA?
INEX_LD-20120112	vietnam war facts

Table 10: Queries used for Figure 4.