BalancedQR: A Framework for Balanced Query Recommendation

Harshit Mishra(✉) and Sucheta Soundarajan

Syracuse University, Syracuse, NY 13244, USA
harsh.dsdh@gmail.com, susounda@syr.edu

Abstract. Online search engines are an extremely popular tool for seeking information. However, the results returned sometimes exhibit undesirable or even wrongful forms of imbalance, such as with respect to gender or race. In this paper, we consider the problem of *balanced query recommendation*, in which the goal is to suggest queries that are relevant to a user's search query but exhibit less (or opposing) bias than the original query. We present a multi-objective optimization framework that uses word embeddings to suggest alternate keywords for biased keywords present in a search query. We perform a qualitative analysis on pairs of subReddits from Reddit.com (r/Republican vs. r/democrats) as well as a quantitative analysis on data collected from Twitter. Our results demonstrate the efficacy of the proposed method and illustrate subtle linguistic differences between words used by sources with different political leanings.

Keywords: search engine · bias · recommender systems

1 Introduction

Online search engines are an extremely popular tool for individuals seeking information. However, as is well known, the results returned by search engines may over- or under-represent results in a way that exhibits undesirable or even wrongful forms of bias [26]. This occurs because search engines commonly use word embeddings to determine the relevance of a document to a search query, which can cause bias: e.g., as argued in [5], a hypothetical query for *CMU computer science phd student* may downrank results for female CS PhD students because male names are closer than female names to the search keywords in the embedding space. In addition to being ethically problematic, this phenomenon may also be unwanted by the user, who may not be aware of the latent bias embedded in their query. In the literature, this problem has been addressed in two main ways: by debiasing a word embedding [5,8] or by re-ranking search results to eliminate such bias [9,34,35].

In this paper, we consider an alternative solution, which we refer to as *balanced query recommendation*, in which an algorithm suggests less or oppositely-biased alternatives to a query. As we observe, if an individual is searching online for a particular query, nuanced, non-obvious differences in keyword choice may

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14172, pp. 420–435, 2023.
https://doi.org/10.1007/978-3-031-43421-1_25

result in major differences in the bias exhibited by the results. For example, as we will see, searching Reddit for the term 'rioting' returns results that are disproportionately from the Republican party subReddit vs. the Democratic party subReddit by a factor of 4:1. However, the term 'protests' gives results that are still highly relevant to the original query, but are much less biased.

While the existing approaches of debiasing search terms or re-ranking search results are valid approaches to the general issue of biased search results, these methods accomplish different goals than what we seek here. First, although forcing debiasing on a user may be desirable in some cases, there are other cases when it is less clearly desirable. For example, in the 'rioting' example above, it is quite possible that a user *wants* results that disproportionately represent one political party. In such cases, providing a query recommendation is a 'gentler' alternative to a behind-the-scenes debiasing, because it allows the user to decide whether she wants to see different results. Second, existing methods of debiasing terms or results do not help the document creators debias their own documents. For example, in the job recruitment application described above, it is certainly useful to the recruiter to have a less biased set of results; but it is also important that the candidates themselves know how to modify their keywords so that they are less likely to be harmed by algorithmic bias.

We present `BalancedQR`, a novel algorithm for balanced query recommendation. `BalancedQR` works in conjunction with existing search algorithms. `BalancedQR` first computes the bias of the results returned in response to a query. It then uses a word embedding to identify related terms, and then measures the bias and relevance of those keywords. Finally, it presents a Pareto front of results of varying bias and relevance. Importantly, `BalancedQR` does *not* require a debiased word embedding: one can use it with respect to any attribute (e.g., gender, race, political alignment, preferred hobby, etc.), as long as there is some way of measuring the bias of a document set with respect to that attribute.

We demonstrate use of `BalancedQR` on pairs of subReddits from reddit.com. In particular, we consider results from r/AskMen and r/AskWomen and r/Republican and r/Democrats. We perform a qualitative evaluation across several queries on these subReddits. We also perform a quantitative evaluation using popular Google Trends search queries on Twitter data.

An early proof-of-concept of `BalancedQR` was published in [24]. Here, we present the full version of `BalancedQR`, including on multi-word queries, and perform a comprehensive evaluation across numerous search algorithms, word embeddings, and datasets.

2 Related Work

To our knowledge, this is the first work to approach the problem of balanced query recommendation. However, there is a large and recent body of work that has addressed group fairness concerns in rankings, including greedy algorithms for fair ranking [23] and a framework for mitigating bias in ranking [11], a re-ranking algorithm that balances personalization with fairness [20], and

diversification-focused approaches for ranking [1]. [17] observed search bias in rankings, and proposed a framework to measure the bias of the results of a search engine. Ranking algorithms have only recently been used to increase fairness. For a set of items with given relevance scores, such algorithms generally extract partial ranking to mitigate bias [4,7,30,33,35].

In contrast to these existing works, our paper focuses on generating balanced query recommendation, as opposed to modifying or auditing the search results directly.

Also related to our work is the problem of debiasing word embeddings [5,18,27,36]. These methods rely on maximizing and minimizing certain sub-vectors of words in a word embedding. [5] examines gender bias, and propose an algorithm that achieves fairness by modifying the underlying data. [36] proposed a learning scheme, Gender-Neutral Global Vectors (GN-GloVe), for training word embedding models based on GloVe [27]. This algorithm protects attributes in certain dimensions while neutralizing other attributes during the training process, thus ensuring that gender-related information is confined to a subvector. [21] proposed a new filtering technique which uses the Z-order prefix, based on the cosine similarity measure that decreases the number of comparisons between the query set and the search set to find highly similar documents. In this paper, we use cosine similarity to find the keywords similar to our search query.

3 Problem and Framework

In this paper, we explore the *balanced query recommendation* problem, in which a user enters a query into a search engine that may return results that are biased with respect to some attribute. These attributes may be those traditionally considered 'protected', such as gender or race; or may be other attributes of interest, such as political alignment.[1] For example, as we will see, the query 'privilege' gives results that are disproportionately from a Republican-associated subreddit.

Balanced query recommendation has similar high-level goals as debiasing search rankings, including reducing 'bubbles' and echo chambers, which can create a divide between people with different views [10,15,25]. However, it provides a 'gentler' approach than directly re-ranking results, in that the user may choose whether to accept a recommended query.

More formally, the goal of the balanced query recommendation problem is to provide a set of query recommendations to the user that are *relevant* to user's original search query, and exhibit greater *diversity*. As discussed below, 'diversity' can be quantified in different ways: here, we measure it with respect to the source of a document, but the framework allows for other approaches.

`BalancedQR` is a general framework for balanced query recommendation, and can be instantiated with the user's choice of relevance and bias measures. `BalancedQR` is intended to supplement an existing search engine, and does not itself perform searches.

[1] Like all work on fairness, we acknowledge that this algorithm must be used judiciously. There exist topics for which 'balance' is not always desirable.

The output of `BalancedQR` is a set of queries that, ideally, have high relevance to the original query but are more diverse/more balanced (for example, if the original query produced results with a strong male bias, the alternatives should be less so, or should exhibit a strong female bias). `BalancedQR` uses no prior information about the dataset and therefore can be used alone or as part of a larger architecture to reduce biases present in social media.

3.1 Problem Setup

In this paper, we will refer to the user's input query as the *original query*. A query is performed on a *dataset* consisting of a set of text documents. This query is performed by an existing search engine (not provided by `BalancedQR`).

For a given search query Q, performing query Q using search engine S on database D results in a set of documents $S(Q)$ (depending on context, one might define this set as, e.g., the first page of documents shown in a browser window). Here, we assume that we are given a fixed search engine and document database, and so drop S and D from the notation when it would not lead to confusion.

There are two components to characterizing a set of documents– diversity and relevance, measured through appropriate user-provided functions g and Rel, respectively. Both of these functions are discussed further below. Ideally, the relevance Rel of $S(Q)$ would be measured by click-through rate, which is the fraction of returned documents clicked by the user. However, in practice, click-through rate is not known ahead of time, and so a different relevance function is required.

We then treat this problem as a multi-objective maximization problem over g and Rel. There are many ways in which this problem can be formulated: for example, maximize g subject to a constraint on Rel (e.g., $Rel(Q') \geq \alpha Rel(Q)$, where $\alpha \in [0, 1]$ is specified by the user); maximize Rel subject to a constraint on g (e.g., $g(Q') \geq \beta$, where β is specified by the user); and others.

The `BalancedQR` framework takes a Pareto front-based approach that returns the Pareto front of terms, as measured by the diversity function g and the relevance function Rel, which are defined as desired by the user.

4 Proposed Method

Using a word embedding, `BalancedQR` creates a list of candidate words for the original search query, and scores words in this list based on relevance and diversity to create a set of suggested words which can be used in place of a biased word to achieve a more diverse set of recommendations.

Measuring Diversity: We measure diversity in terms of *bias*. Each document returned from the search engine for a query Q has a *bias* between -1 and 1. These bias scores could be derived from, for instance, bias annotations on the sources of news articles, such as those provided by www.allsides.com. The bias for a query Q is then simply the average of the *bias* scores of the returned documents. A low *bias* means most of the documents returned were from different sets, leading to high diversity; and vice versa.

Measuring Relevance: We define the *relevance* of a candidate query Q' to an original query Q in terms of the similarity between the document sets returned for each query. There are various similarity measures that can be used such as Euclidean distance, Manhattan distance, Cosine similarity, Jaccard similarity, and others. In this work, we use Cosine similarity (bag of words representation), in which for each document in the two sets, we find the most similar document in the other set, and so define a mean Cosine similarity for each set. The overall similarity is the harmonic mean of these values (akin to F1-score).

4.1 Recommendation Framework

Denote the original query as Q and the dataset as D. As described before, `BalancedQR` works in conjunction with an existing search algorithm, which is used to perform the keyword searches. As before, denote this search algorithm as S. Without loss of generality, we assume that the search algorithm returns the top-n results for some fixed n. In the following discussion, we assume that the dataset and search algorithm are fixed. As before, let $g(d)$ be the bias of a particular document d, and $Rel(d)$ be the relevance of document d to keyword Q. Denote the word embedding used by `BalancedQR` as W.

At a high level, `BalancedQR` performs the following steps:

(1) Given query Q, `BalancedQR` applies the search algorithm S to document set D and fetches $S(Q)$, the top-n most relevant documents to Q from D.

(2) `BalancedQR` then performs an iterative process in which it identifies the k alternative keywords $Q_1, ..., Q_k$ nearest to Q in the embedding space defined by W (the choice of k depends on the termination criteria, see below). For a multi-word query, it uses a large language model to fetch alternative multi-word queries based on keywords fetched from W.[2] It then uses search engine S to perform a search of each Q_i on dataset D to obtain set $S(Q_i)$. For each of these i sets, `BalancedQR` computes the bias and relevance of those sets, where relevance is measured with respect to the *original* query Q. Using these values, `BalancedQR` produces a Pareto front along the bias-relevance axes. This Pareto front contains an alternative query Q_i if Q_i is not dominated by any of the other alternatives or by Q itself. A query is non-dominated if there is no other query whose search results have both a lower bias and higher relevance score. (Note that it may sometimes be more appropriate to use a 'pseudo'-Pareto front that allows for queries that are highly biased, but in the opposite direction.)

(3) `BalancedQR` repeats the above step until a satisfactory Pareto front has been defined, and outputs the Pareto front (or a desired subset) to the user. In our experiments we continue until 10 recommended keywords are found, or no more are available. In our analysis, we highlight both the Pareto front as well as high-relevance words with opposing bias.

Through this process, the end user is made aware that by using an alternate query she can still get relevant results, but from a different point of view (Table 1).

[2] LLMs are known to exhibit their own bias, and, if desired, debiasing may be applied at that stage [14,19]. The bias of LLMs is outside the scope of this paper.

Table 1. Collective Inputs and Outputs of Algorithm

Inputs	Q: Input Query
	Q_i: Alternative Query
	d, D: Document, set of documents
	S: Search algorithm
	$S(Q)$: Top-n most relevant documents to query Q from document set D, as found by algorithm S
	$g(d), g(D)$: Diversity of a document d or document set D
	$Rel(d), Rel(D)$: Relevance of a document d or document set D to query Q
	W: Word embedding

4.2 Our Implementation

Measuring Relevance. We compute relevance using a cosine similarity-based approach that compares the documents returned for Q_i to those returned for Q. In this approach, we compute a variant of F1 by measuring the precision and recall as follows: First, for each document $d' \in S(Q_i)$ (the top-n documents returned in response to Q_i), we compute the greatest similarity between d' and a document $d \in S(Q)$ (the top-n documents returned in response to Q). This similarity is measured using cosine similarity between the bag-of-words corresponding to the documents. The *precision* is then the average of these maximum similarities. *Recall* is computed similarly, but in the other direction (i.e., finding the closest document from $S(Q_i)$ to each document in $S(Q)$). Then the F1-score, or relevance, is the harmonic mean of precision and recall.

Measuring Bias. In the bulk of our analysis (described in Sect. 5), we use a dataset scraped from Reddit.com. We consider posts (documents) from pairs of subReddits in which each subReddit corresponds to a particular group (e.g., Republican vs. Democrats). In this case, the bias function follows directly from the dataset. For a given document/post, that document has bias of either $+1$ (indicating that it was posted in one subReddit) or -1 (indicating that it was posted in the other). We also perform an analysis on Twitter data. Here, we use the AllSides media bias annotations [2] to label the bias of sources (the bias calculation for Twitter is described in Sect. 6.2).

The bias of a set of documents D is simply the average of the biases of the individual documents.

Termination. We find the top-k closest keywords based on the word embedding. In our experiments, we set $k = 10$: this appeared empirically to be sufficient to identify alternative queries. In our analysis, we highlight both the Pareto front (computed using the scalar version of bias), as well as high-relevance words with opposing bias. Also, It is possible that in certain situations no alternative queries are found and in those cases, no alternative queries are returned.

Algorithm 1 . Balanced query recommendation

Q = original query
k = number of desired queries, n = number of returned documents
max_iter = maximum number of iterations, $num\_iters = 0$
$Bias_Q = g_Q(D)$
sim = list of k most similar words from word embedding
if Q is multi-word query **then**
    sim = list of LLM(w') for each w' in sim
end if
$recs = \{Q\}$
while $|num\_iters| < max\_iter$ and $|recs| < k$ **do**
    for each query w' in sim **do**
        $S_{w'}(D, n)$ = top-n relevant documents from D for w'
        $Bias_{w'} = g_{w'}(D)$
        $Rel(w')$ = F1-score between $S_{w'}(D, n)$ and $S_Q(D, n)$
        if w' is not dominated by any query in $recs$ **then**
            Add w' to $recs$
            Remove queries from $recs$ that are dominated by w'
        end if
    end for
    sim = {next most similar word from word embedding}
    $num\_iters++$
end while
Return $recs$

4.3 Limitations

The `BalancedQR` framework has a few important limitations. As explored in other works, word embeddings learned from collections of data often demonstrate a significant level of biases [12]. When these embeddings are used in low-level NLP tasks, it often amplifies the bias. Thus, it is quite likely that the GloVE embedding that we use is itself biased, reducing the efficacy of `BalancedQR`. Similarly, large language models may exhibit (sometimes substantial) bias as cited by Bender et al. in Sect. 4.3 of [3], which may also counter the efforts of `BalancedQR`. However, debiasing word embeddings and LLMs is a challenging problem that is the subject of much active research, and is outside the scope of this paper. `BalancedQR` is not inherently tied to any particular word embedding or LLM, and if less biased or unbiased word embeddings/LLMs are created, they can easily be used.

Second, `BalancedQR` only supports bias computations along one axis. In many cases, a query is biased along multiple dimensions. Dealing with this is challenging, but one solution is to define bias in a multidimensional space. For each candidate query, we can then calculate the final bias by finding the L2 norm of bias in this multidimensional space with respect to the original bias distribution of the dataset.

5 Experimental Setup

We conduct experiments on data from two sources– Reddit.com and Twitter.com, using multiple word embeddings and search engine/document retrieval strategies. The Reddit dataset consists of posts from pairs of subReddits (each of which can be thought of as a topic-specific discussion forum). Each pair represents a particular attribute of interest. The Twitter dataset consists of tweets from various news sources. Each of these tweets is assigned a specific bias based on political leaning of the news source according to AllSides media bias chart [2]. Later, we discuss these datasets as well as the simple search engine that we implemented to demonstrate `BalancedQR`. As described in Sect. 6.2, we use multiple word embeddings, including GloVe [28], GoogleNews [13], all-mpnet-base-v2 [31], as well as a word embedding created from the dataset, we used 'gensim' [29] to create the word embedding using it's implementation of word2vec algorithms. We also use multiple search engines for documents retrieval, including TF-IDF, BM25 [22] and FAISS [16].[3]

5.1 Data

Reddit. Given that there is no ground truth for which queries 'should' be returned, we perform a qualitative analysis in which we demonstrate the use of `BalancedQR` on real data. For our analysis, we compare pairs of contrasting subReddits from Reddit.com. Using the Python PRAW package, we crawled 'top' posts from the following pairs: (r/AskMen, r/AskWomen), and (r/Republican, r/Democrats). Additional pairs were considered, with similar results, but are not included here due to space constraints.

We collected a roughly equal number of posts from each subReddit in a pair. Dataset statistics– the number of posts collected and the total number of members of each subReddit– are shown in Table 2. Most of the data was collected in October, 2020 with additional political data collected in late January, 2021.

Next, we used data from Google Trends [32] to create a list of evaluation queries based on top trending queries. Most of these queries did not appear in the dataset or did not show substantial bias. For each pair of subReddits, we identified certain queries that showed interesting differences between the two subReddits.

Table 2. Dataset properties.

subReddit	Posts Collected	Members
r/AskMen	3618	2.2M
r/AskWomen	2431	1.8M
r/democrats	2445	143K
r/Republican	2262	147K

[3] https://github.com/harshdsdh/BalancedQR.

Twitter. We also perform an analysis on Twitter data. We collect tweets from major news sources between the period of Oct 2021 to Feb 2022. We used the AllSides media bias chart [2] to label the political leaning of each news source. We focused on tweets from news sources labeled as 'leaning left', 'left', 'center', 'leaning right' and 'right' by AllSides. We collected total 39k tweets from left and leaning-left sources, 17k tweets from leaning-right and right sources and 15k tweets from center sources.

Next, we again use data from Google Trends to create a list of evaluation queries based on top news/ political queries in years 2020, 2021 and 2022. We collect 50 relevant queries for the evaluation purpose.

5.2 Search Engines

To demonstrate `BalancedQR` on the datasets, we implement search engines based on tf-idf, Faiss and BM25.

In the case of tf-idf [22], we compute the tf-idf score of each document with respect to the original query Q, and return the 20 highest scoring documents (or fewer, if fewer than 20 documents use that query).

Faiss is a vector search library that provides a way to search for similar documents based on euclidean distance [16]. For `BalancedQR`, we convert query Q and the dataset into vectors using a pre-trained sentence embedding. For this evaluation we use MPNet [31]. We then use Faiss to retrieve 20 similar documents for original query Q.

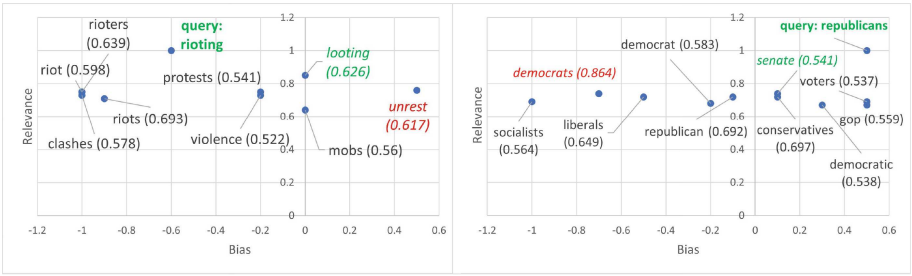
For BM25 [22], we compute the idf score of each document with respect to Q and return the top 20 highest scoring documents.

Obviously, real-world search engines are much more sophisticated than these techniques, but our goal here is to demonstrate `BalancedQR` across a variety of search algorithms.

5.3 Word Embedding

For this analysis, we use multiple word embeddings, including GloVe [28], GoogleNews [13], and our own word2vec word embedding created from the Twitter dataset, we used 'gensim' [29] to create the word embedding using it's implementation of word2vec algorithms. We also use a pretrained MPNet based sentence transformer to get a sentence-embedding for the dataset [31]. `BalancedQR` uses embeddings to calculate document similarity as well as to create a set of potential recommended queries.

We use *gpt-3.5-turbo* in our analysis to retrieve multi-word queries, using cosine similarity in word embedding we create a set of similar keywords w_i for a word w in query Q [6]. We then use *gpt-3.5-turbo* with the prompt *'I am a highly intelligent question answering bot. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with Unknown. For original query Q, frame a new query using* w_i*. Be as brief as possible'* to fetch similar related queries. GPT and other large language models are prone to misinformation, so



(a) Results for original query 'rioting'. (b) Results for original query 'republicans'.

(c) Results for original query 'loneliness'. (d) Results for original query 'sadness'.

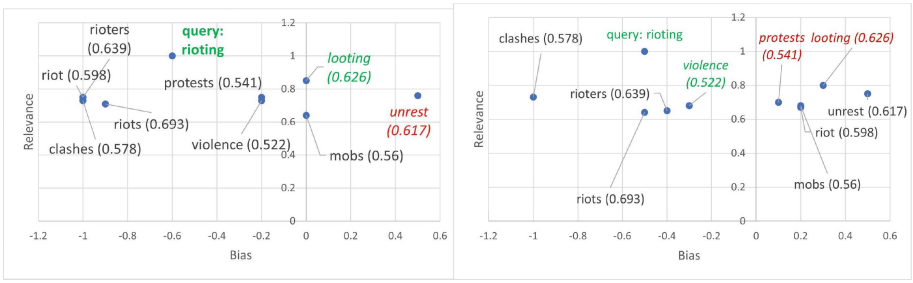
Fig. 1. Potential keyword recommendations for queries on political (top) and gender (bottom) subReddits. Along the x-axis, positive values represent bias towards the r/Democrats (r/AskMen) subReddit, while negative values represent bias towards the r/Republican (r/AskWomen) subReddit. Words on the Pareto front (where bias is a scalar) are circled in green. High-relevance words with opposite bias are indicated in red. (Color figure online)

we provide a seed word w_i for the new query, as a way to reduce wrongful or unrelated queries and to ground them in the context of the original query Q. We then use this generative query for `BalancedQR`. This is done purely for purposes of demonstrating `BalancedQR`: the user can use whatever technique is desired to find similar queries, and we make no specific recommendation on whether *gpt-3.5-turbo* should or should not be used in such a context.

6 Analysis and Discussion

6.1 Reddit

We first discuss the results of `BalancedQR` on the two pairs of subReddits described earlier. Results presented here use the document similarity-based relevance calculation, as discussed in Sect. 4.2. A bias of $\pm p$ indicates the sum of the document biases, divided by the total number of documents. A bias of 0 thus indicates that an equal number of documents from each subReddit were returned. A bias of ± 1 indicates that all results were from one subReddit. In all



(a) Results for 'rioting' based on data collected in October, 2020.

(b) Results for 'rioting' based on data collected after Jan 6th, 2021.

Fig. 2. Potential keyword recommendations for 'rioting' on political subReddits. Along the x-axis, positive values represent bias towards the r/democrats subReddit, while negative values represent bias towards the r/Republican subReddit. Words on the Pareto front (where bias is a scalar) are circled in green. High-relevance words with opposite bias are indicated in red. (Color figure online)

plots, the original query is shown in boldface. This query always has a relevance of 1 to itself.

The Pareto front, computed by treating bias as a scalar (directionless) is circled in green, and includes the original keyword. We additionally note high-relevance words that are biased in the opposite direction.

Although our implementation of `BalancedQR` considers the 10 words closest to the original keyword (using the GloVe word embedding), in some cases, some of these words occurred 0 times in the dataset and do not appear in plots. For the results mentioned in following sections, we use tf-idf as our search algorithm and the GloVe word embedding is used to calculate word and document similarity. Results were similar for other search algorithms/embeddings.

Politics. For the political subReddits (r/Democrats, r/Republican), we considered the query keywords 'rioting' and 'republicans'. On these plots, a positive bias (right side of plots) indicates a bias towards r/Democrats, and a negative bias (left side of plots) indicates a bias towards r/Republican. Results for each of these keywords are shown in Fig. 1.

Results for the query 'rioting' are shown in Fig. 1a. The keyword itself returns results disproportionately from the Republicans subReddit (by a 4:1 ratio, giving a bias of $\frac{1-4}{5} = -0.6$). When considering words that are highly relevant, we observed that 'unrest' returns results disproportionately from the Democratic subReddit (by a 3:1 ratio, for a bias of 0.5), and 'riots' returns results biased towards Republicans subReddit (with no results from the Democratic subReddit). Interestingly, almost all related keywords are either neutral or Republican-biased: the only related word with a Democratic bias is 'unrest'. The keyword returned by `BalancedQR` on the Pareto front is 'looting', as this returns documents that are evenly balanced between the subReddits. If desired, `BalancedQR`

can also return 'unrest' to provide a Democratic counterbalance to the original keyword.

For the 'rioting' keyword specifically, we were curious to see if there were any changes to the above results after the January 6th, 2021 insurrection. To answer this, we re-collected data and redid this analysis. Original results (collected in October 2020) are shown in Fig. 2a and new results (collected in early January 2021) are shown in Fig. 2a. As shown in Fig. 2b, the earlier data shows that posters in the Democratic party subReddit tended to use words such as 'unrest' instead of 'rioting', while individuals in the Republican subReddit were using words such as 'rioters' and 'riots'. Fig. 2b shows results from January 2021. People in the Democratic subReddit use words such as 'protests' and 'mobs' for this concept. Posters in the Republican subReddit still use words such as 'rioters' and 'riots', but there is a reduction in the bias of these words.

We also discuss results for the query 'republicans'. Interestingly, 'republicans' has a bias towards r/Democrats; but 'republican' has a slight bias towards r/Republican. Upon further inspection of posts, a possible explanation for this is that Democrats are more likely to discuss Republicans as a group, while individual Republicans may discuss their own identity as a (singular) Republican. In this case, `BalancedQR` suggests 'senate' on the Pareto front.

Next, we present sample results for the gender-based subReddits r/AskMen and r/AskWomen. We consider the original queries 'loneliness' and 'sadness'. Positive values of bias indicate bias towards r/AskMen, and negative values of bias indicate bias towards r/AskWomen. Results are shown in Fig. 1.

Figure 1c shows results for the query 'loneliness'. The original query returns results disproportionately from the r/AskMen (bias of 0.25). When considering words that are highly relevant, we observed that 'sadness' and 'anguish' were also biased towards r/AskMen, while 'boredom' was biased towards r/AskWomen (by a 2:1 ratio, for a bias of −0.33). The queries recommended by `BalancedQR` on the Pareto front are 'grief', which is slightly less r/AskMen-biased than 'loneliness'; and 'anxiety', which does not show bias towards either subReddit. Potential candidates with opposite bias include 'boredom' and 'longing', both of which are extremely relevant to 'loneliness'.

6.2 Twitter

Next, we perform a quantitative evaluation on Twitter data. We use a set of 50 queries, including 'Georgia Senate Race', 'Roe v Wade' and 'QAnon' applied on the dataset of 72k rows.

Figure 3 describes relevance and bias of results obtained for various implementations of `BalancedQR`. In this figure, bias under the 'Query', 'All Possible Recommend Queries', and 'BalancedQR Recommended Queries' columns refers to the average bias (cosine similarity) and relevance of the set of documents retrieved for the original query, queries produced via word embedding similarity alone (without care for bias), and by `BalancedQR`. Relevance is 1 when two queries retrieve the same documents and 0 when no documents are similar.

Search Engines	Word/ Sentence Embedding	Document similarity Embedding	Query		All Possible Recommended Queries		BalanceQR Recommended Queries	
			Bias	Relevance	Bias	Relevance	Bias	Relevance
BM25	GloVe	GloVe	0.305	1	0.340	0.898	0.146	0.929
	GoogleNews	GoogleNews	0.276	1	0.332	0.728	0.178	0.822
	Twitter_w2v	Twitter_w2v	0.284	1	0.306	0.918	**0.114**	0.936
	Twitter_w2v	all-mpnet-base-v2	0.276	1	0.306	0.465	0.117	0.604
	Twitter_w2v + gpt-3.5-turbo	all-mpnet-base-v2	0.267	1	0.395	0.652	0.211	0.785
	Twitter_w2v + gpt-3.5-turbo	Twitter_w2v	0.267	1	0.395	0.944	0.216	**0.965**
TF-IDF	GloVe	GloVe	0.398	1	0.422	0.855	**0.156**	0.893
	GoogleNews	GoogleNews	0.398	1	0.500	0.575	0.182	0.724
	Twitter_w2v	Twitter_w2v	0.398	1	0.498	0.897	0.203	0.933
	Twitter_w2v	all-mpnet-base-v2	0.398	1	0.498	0.360	0.190	0.547
	Twitter_w2v + gpt-3.5-turbo	all-mpnet-base-v2	0.408	1	0.383	0.628	0.276	0.734
	Twitter_w2v + gpt-3.5-turbo	Twitter_w2v	0.408	1	0.395	0.937	0.270	**0.953**
FAISS	GloVe	GloVe	0.421	1	0.308	0.845	0.180	0.883
	GoogleNews	GoogleNews	0.421	1	0.27	0.577	0.215	0.668
	Twitter_w2v	Twitter_w2v	0.421	1	0.295	0.899	0.141	0.916
	Twitter_w2v	all-mpnet-base-v2	0.421	1	0.295	0.342	**0.133**	0.477
	Twitter_w2v + gpt-3.5-turbo	all-mpnet-base-v2	0.418	1	0.592	0.709	0.395	0.803
	Twitter_w2v + gpt-3.5-turbo	Twitter_w2v	0.421	1	0.295	0.889	0.391	**0.963**

Fig. 3. Analysis for bias and relevance for queries on Twitter data. `BalancedQR` recommends relevant queries with less bias compared to other queries.

As we compare values across the three query columns, we observe that `BalancedQR` produces less biased and highly relevant results. As we see, `BalancedQR` achieves highest relevance when used with a combination of word embedding created from the dataset and *gpt-3.5-turbo*. In this case, the average bias of the set of recommended queries is less than the bias of set of original queries as well as they are still relevant. We also observe that a higher relevance when we combine a word embedding with *gpt-3.5-turbo* for potential recommended queries.

Table 3 shows examples of several queries and recommended queries. Consider the query 'Stimulus Check'. During the COVID pandemic, stimulus checks were provided by the US government to its citizens. The list of related terms includes 'subscribe', 'list' and 'debunk'. The original query returns result that are disproportionately from left-leaning news sources (bias of 0.62). After observing highly relevant but less imbalanced queries, `BalancedQR` (using *gpt-3.5-turbo*) returns queries such as 'can I subscribe for updates on stimulus checks' (bias of

0.3, relevance of 0.91). These are still skewed towards left-leaning sources, but less so, and are still highly relevant.

Table 3. Sample results from LLM

Original query	Suggested keyword from word embedding	Suggested query from LLM	Sample queries from `BalancedQR`
Roe v Wade	dobbs	What is the impact of dobbs v jackson womens health organization on roe v wade	which justices were on the supreme court for roe v wade
	alito	What is alitos stance on roe v wade	what would happen if roe v wade was overturned
QAnon	convincing	What is the convincing evidence for the claims made by qanon	What is the convincing evidence for the claims made by qanon
	conspiracy	What distinguishes qanon from other conspiracy theories	what are the chilling effects of qanon
Stimulus Check	drugmaker	What drugmakers have received stimulus funds	can i subscribe for updates on stimulus checks
	value	what is the value of the latest stimulus check	what is the value of the latest stimulus check

7 Conclusion and Future Work

In this paper, we considered the problem of *balanced query recommendation*, and proposed `BalancedQR`, an algorithmic framework for identifying highly-relevant but less-biased query alternatives. A major application of this problem is on web search, where balanced query recommendation can be a step in addressing problems caused by echo chambers or filter bubbles. Search engines can leverage `BalancedQR` as a post-processing method using it to recommend less biased query alternatives to the end user. It can also be used as a plugin to any web browser. In future work, we will explore dealing with multiple dimensions of bias.

Acknowledgements. Soundarajan was supported in part by NSF #2047224.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Managing popularity bias in recommender systems with personalized re-ranking. arXiv preprint arXiv:1901.07555 (2019)
2. allsides (2021)

3. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: can language models be too big. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2021, pp. 610–623. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3442188.3445922
4. Beutel, A., et al.: Fairness in recommendation ranking through pairwise comparisons. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2212–2220 (2019)
5. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In: Advances in Neural Information Processing Systems, pp. 4349–4357 (2016)
6. Brown, T.B., et al.: Language models are few-shot learners (2020)
7. Celis, L.E., Straszak, D., Vishnoi, N.K.: Ranking with fairness constraints. arXiv preprint arXiv:1704.06840 (2017)
8. Dev, S., Li, T., Phillips, J.M., Srikumar, V.: On measuring and mitigating biased inferences of word embeddings. In: AAAI, pp. 7659–7666 (2020)
9. Dutta, R.: System, method, and program for ranking search results using user category weighting (2002). US Patent App. 09/737,995
10. Flaxman, S., Goel, S., Rao, J.M.: Filter bubbles, echo chambers, and online news consumption. Public Opin. Q. **80**(S1), 298–320 (2016)
11. Geyik, S.C., Ambler, S., Kenthapadi, K.: Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2221–2231 (2019)
12. Gonen, H., Goldberg, Y.: Lipstick on a pig: debiasing methods cover up systematic gender biases in word embeddings but do not remove them. arXiv preprint arXiv:1903.03862 (2019)
13. Google
14. Guo, Y., Yang, Y., Abbasi, A.: Auto-debias: debiasing masked language models with automated biased prompts. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, pp. 1012–1023. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-long.72. https://aclanthology.org/2022.acl-long.72/
15. Himelboim, I., McCreery, S., Smith, M.: Birds of a feather tweet together: integrating network and content analyses to examine cross-ideology exposure on twitter. J. Comput.-Mediat. Commun. **18**(2), 154–174 (2013)
16. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2019)
17. Kulshrestha, J., Eslami, M., Messias, J., Zafar, M.B., Ghosh, S., Gummadi, K.P., Karahalios, K.: Quantifying search bias: investigating sources of bias for political searches in social media (2017). https://arxiv.org/pdf/1704.01347.pdf
18. Kaneko, M., Bollegala, D.: Gender-preserving debiasing for pre-trained word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL) (2019)
19. Khalifa, M., Elsahar, H., Dymetman, M.: A distributional approach to controlled text generation. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=jWkw45-9AbL
20. Liu, W., Burke, R.: Personalizing fairness-aware re-ranking. arXiv preprint arXiv:1809.02921 (2018)

21. Alewiwi, M., Orencik, C., Savas, E.: Efficient top-k similarity document search utilizing distributed file systems and cosine similarity (2017). https://arxiv.org/pdf/1704.01347.pdf
22. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008). https://doi.org/10.1017/CBO9780511809071
23. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fa*ir: a fair top-k ranking algorithm (2018). https://arxiv.org/pdf/1706.06368.pdf
24. Mishra, H., Soundarajan, S.: Keyword recommendation for fair search. In: Boratto, L., Faralli, S., Marras, M., Stilo, G. (eds.) BIAS 2022, pp. 130–142. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-09316-6_12
25. Nguyen, C.T.: Echo chambers and epistemic bubbles. Episteme **17**(2), 141–161 (2020)
26. Noble, S.U.: Algorithms of Oppression: How Search Engines Reinforce Racism. NYU Press, New York (2018)
27. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics (2014). https://doi.org/10.3115/v1/D14-1162. https://www.aclweb.org/anthology/D14-1162
28. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
29. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, pp. 45–50. ELRA (2010)
30. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2219–2228 (2018)
31. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: MPNet: masked and permuted pre-training for language understanding. In: NeurIPS 2020. ACM (2020). https://www.microsoft.com/en-us/research/publication/mpnet-masked-and-permuted-pre-training-for-language-understanding/
32. Google Trends (2021)
33. Vogel, R., Bellet, A., Clémençon, S.: Learning fair scoring functions: fairness definitions, algorithms and generalization bounds for bipartite ranking. arXiv preprint arXiv:2002.08159 (2020)
34. Zehlike, M., Castillo, C.: Reducing disparate exposure in ranking: a learning to rank approach. In: Proceedings of the Web Conference 2020, pp. 2849–2855 (2020)
35. Zehlike, M., Sühr, T., Castillo, C., Kitanovski, I.: Fairsearch: a tool for fairness in ranked search results. In: Companion Proceedings of the Web Conference 2020, pp. 172–175 (2020)
36. Zhao, J., Zhou, Y., Li, Z., Wang, W., Chang, K.: Learning gender-neutral word embeddings. CoRR abs/1809.01496 (2018). https://arxiv.org/abs/1809.01496