Keyword Recommendation for Fair Search

Harshit Mishra[1] , **Namrata Madan Nerli**[1] and **Sucheta Soundarajan**[1]
[1]Syracuse Univesity
{hamishra, nnerli, 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 bias, such as with respect to gender or race. In this paper, we consider the problem of *fair keyword recommendation*, in which the goal is to suggest keywords that are relevant to a user's search query, but exhibit less (or opposite) bias. We present a multi-objective method using word embedding to suggest alternate keywords for biased keywords present in a search query. We perform a qualitative analysis on pairs of subReddits from Reddit.com (e.g., r/AskMen vs. r/AskWomen, r/Republican vs. r/democrats). Our results demonstrate the efficacy of the proposed method and illustrate subtle linguistic differences between subReddits.

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 underrepresent results in a way that exhibits undesirable or even wrongful forms of bias. This occurs because search engines commonly use word embeddings to determine the relevance of a document to a search query: e.g., as shown in [Bolukbasi *et al.*, 2016], a query for *computer science phd student* may downrank results for female CS PhD students, because male names are closer to the search keywords than are female names 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. First, one can attempt to debias a word embedding [Bolukbasi *et al.*, 2016], [Dev *et al.*, 2020]. Second, one can re-rank search results to eliminate such bias [Dutta, 2002], [Zehlike and Castillo, 2020], [Zehlike *et al.*, 2020].

In this paper, we consider an alternative solution, which we refer to as *fair keyword recommendation*, in which an algorithm suggests less or oppositely-biased alternatives to a keyword. Our problem is motivated by conversations with an academic administration recruiter, who recounted her experiences with searching for job candidates on LinkedIn: when searching for individuals with a particular qualification, she noticed that her search results were primarily white men. However, because she was active in her field, she knew that there were many good female and minority candidates. When she looked up these female and minority candidates, she noticed that they tended to use different keywords to reflect the same type of qualification. For example, the terms 'secretary' and 'administrative assistant' are often used interchangeably. However, because of sexist connotations [Blaubergs, 1978], men may be unlikely to use the term 'secretary' to refer to themselves; in contrast, the term 'administrative assistant' may be more likely to return less gender-biased results. In such cases, a recruiter searching for 'secretary' may wish to know that 'administrative assistant' is a similar but less biased keyword. Additionally, job candidates selecting keywords for their resumes may wish to know whether their choice of keyword is encoding some sort of bias.

We present `FairKR`, a novel algorithm for fair keyword recommendation. `FairKR` works in conjunction with existing search algorithms. `FairKR` first computes the bias of the results returned in response to a keyword. 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, `FairKR` 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 `FairKR` on pairs of subReddits from reddit.com (r/AskMen and r/AskWomen, r/Republican and r/Democrats, and r/AskABrit and r/AskAnAmerican) Although the nature of our problem makes it inherently difficult to evaluate results in a quantitative way, we perform a qualitative evaluation across several queries on these subReddits. The results demonstrate the efficacy of `FairKR` and give interesting insight into subtle differences in language choice between different groups of people.

2 Related Work

To our knowledge, ours is the first work to examine the problem of fair keyword recommendation. However, there is a recent body of work that has addressed group fairness concerns in rankings. Much of the existing work uses the statistical parity criterion to detect unfairness in the top-k items of a ranking. [Meike Zehlike and Baeza-Yates., 2018] extends such statistical parity approaches, and introduces a greedy algorithm that attempts to identify a ranking that is fair but optimized for utility. [Geyik *et al.*, 2019] presented a framework for mitigating algorithmic bias for ranking individuals. This work introduced three variations of a greedy algorithm, as well as a feasible algorithm for fairness-aware ranking. [Juhi Kulshrestha and Gummadi, 2017] observed search bias in rankings, and proposed a framework to measure the bias of the results of a search engine. This framework identifies the extent to which bias in the output is due to the input, and how much is due to the bias in the system itself.

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

3 Proposed Method

In this paper, we explore the *fair keyword recommendation* problem. In this problem, a user enters a keyword into a search engine, which 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. We propose FairKR, a multi-objective optimization algorithm framework that uses a word embedding to suggest alternate keywords for a user’s search query. We use the GloVe embedding, but any embedding will do. Next, FairKR creates a list of candidate words for the original search query, and scores words in this list based on relevance and bias 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.

3.1 Problem Setup

FairKR is a general framework for fair keyword recommendation, and can be instantiated with the user’s choice of relevance and bias measures. FairKR is intended to supplement an existing search engine, and does not itself perform searches.

In this paper, we will refer to the user’s input query as the *query keyword*. A query is performed on a *dataset* consisting of a set of text documents. As discussed in Section 3.2, we assume that the algorithm has some way to measure the relevance of a particular document with respect to the query. As discussed in Section 3.2, we also assume that the algorithm is provided with a function to score the ‘bias’ of a particular document.

The output of FairKR is a set of keywords that, ideally, have high relevance to the query keyword but are either less or oppositely biased (for example, if the query keyword produced results with a strong male bias, the alternatives may be either less biased or female-biased). FairKR 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.

3.2 The FairKR Framework and Implementation

Denote the query keyword as Q and the dataset as D. As described before, FairKR works in conjunction with an existing search engine/algorithm, which is used to perform the keyword searches. Denote this search algorithm as S. $S_Q(D, n)$ denotes the top-n results returned by S applied to D. Let $B(d)$ be the bias of document d with respect to an attribute of interest, and (overloading the notation), let $B(D)$ represent the total bias of document set D. We assume $B(d)$ (the bias of an individual document) is binary (as is appropriate for our dataset), but this can easily be changed if desired. Similarly, let $r_Q(d)$ be the relevance of document d to keyword Q, and $r_Q(D)$ be the relevance of a document set D to Q. Denote the word embedding used by FairKR as W (W need not be debiased in any way). The definition for relevance and bias functions used in our implementation are discussed in Sections 3.2 and 3.2.

Let n denote the desired number of returned documents.

At a high level, FairKR performs the following steps.

As a pre-processing step, FairKR removes stop words and tokenizes each word in each document $d \in D$.

First, for a keyword Q, FairKR applies the search algorithm S to D and fetches $S_Q(D, n)$, the top-n most relevant documents from D.

Then, FairKR performs an iterative process in which it identifies the k alternative keywords $A_1, ..., A_k$ nearest to Q in the embedding space defined by W (the choice of k depends on the termination criteria). It then uses S to perform a search of each A_i using search algorithm S applied to D to obtain set $S_{A_i}(D, n)$. For each of these i sets, FairKR computes the bias and relevance of those sets with respect to the *original* keyword Q. Using these values, FairKR produces a Pareto front along the bias-relevance axes, where an alternative keyword A_i is retained if it is not dominated by any of the other alternatives or by Q itself. A keyword is non-dominated if there is no other keyword whose search results have both a lower bias and higher relevance score. (As discussed in Section 3.2, it may sometimes be more appropriate to use a ‘pseudo’-Pareto front that allows for keywords that are highly biased, but in the opposite direction.)

FairKR repeats the above step until a satisfactory Pareto front has been defined. The termination criteria are application-specific, and are discussed in Section 3.2.

Measuring Relevance

There are a number of ways to measure the relevance of document d to a keyword Q. We compute relevance using a cosine similarity-based approach that compares the documents returned for A_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 $d' \in S_{A_i}(D, n)$ (the top-n documents returned in response to A_i), we compute the greatest similarity between d' and a document $d \in S_Q(D, n)$ (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

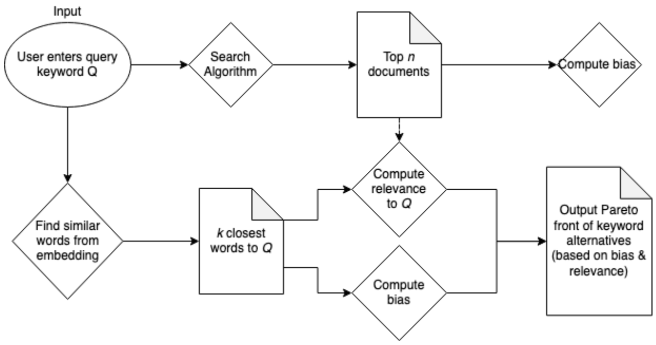
Input
User enters query keyword Q
Search Algorithm
Top *n* documents
Compute bias
Find similar words from embedding
k closest words to *Q*
Compute relevance to *Q*
Compute bias
Output Pareto front of keyword alternatives (based on bias & relevance)

Figure 1: Implementation of FairKR algorithm

document from $S_{A_i}(D, n)$ to each document in $S_Q(D, n)$). Then the F1-score, or relevance, is the harmonic mean of precision and recall.

Measuring Bias

In our dataset (described in Section 4), we obtain posts from Reddit.com. We consider posts (documents) from pairs of subReddits in which each subReddit corresponds to a particular group (e.g., AskMen vs. AskWomen). 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). The bias of a set of documents D is simply the sum of the biases of the individual documents, divided by the total number of documents.

Termination and Results

We find the top-k closest keywords based on the word embedding.[1] In our experiments, $k = 10$: this appeared empirically to be sufficient to identify several alternative keywords. 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.

4 Experimental Setup

Here, we discuss our datasets as well as the simple search engine that we implemented for purposes of demonstrating `FairKR`.

4.1 Data

For our analysis, we compare pairs of contrasting subReddits from Reddit.com. Using the Python PRAW package, we crawled 'top' posts from the subreddit pairs. Dataset statistics- the number of posts collected and the total number of members of each subReddit- are shown in Table 1. We have roughly the same number of posts from each subReddit in a pair, `FairKR` does not need to account for size differences in its bias computation.

Table 1: Dataset properties

subreddit	Posts Collected	Members
r/AskMen	3618	2.2M
r/AskWomen	2431	1.8M
r/democrats	2445	143K
r/Republican	2262	147K
r/AskAnAmerican	998	181K
r/AskABrit	986	2.4K

4.2 Search Engine

To demonstrate `FairKR`, we implement a simple search engine algorithm. For a given query word Q, we compute the tf-idf score of each document with respect to Q, and return the 20 highest scoring documents (or fewer, if fewer than 20 documents use that word). Obviously, real-world search engines are much more sophisticated than this, and `FairKR` can work with any existing search algorithm.

5 Experimental Analysis

Here, we discuss the results of `FairKR` on the three pairs of subReddits described earlier. Results presented here use the document similarity-based relevance calculation, as discussed in Section 3.2. 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 plots, the original query is shown in boldface. This query, naturally, 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 circle high-relevance words that are biased in the opposite direction. Depending on the application, `FairKR` may be implemented to return just the Pareto front, or the Pareto front plus the high-relevance, opposite-biased words.

5.1 Politics

For the political subReddits (r/democrats, r/Republican), we considered the keyword 'rioting'. On the plot, 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 the query 'rioting' are shown in Figure 2. 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'. This could indicate the extent to which riots have been discussed by Redditors from each political party (for instance, Democrats may instead choose to discuss factors leading to riots or protests, rather than the riots or protests

[1] In our current work, we are exploring ways to use the Pareto front of results to identify the wordset to be returned.

themselves). The keyword returned by FairKR on the Pareto front is ‘looting’, as this returns documents that are evenly balanced between the subReddits. If desired, FairKR can also return ‘unrest’ to provide a Democratic counterbalance to the original keyword.

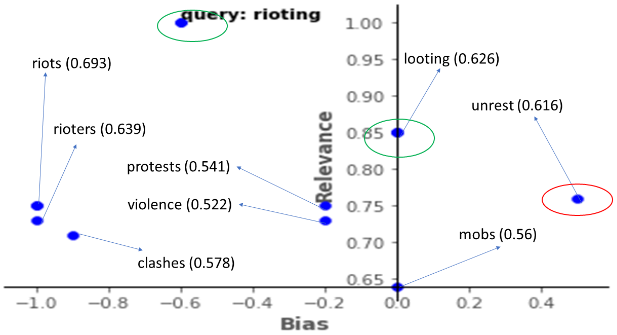

Figure 2: pareto front for word ‘rioting’

5.2 Gender

Next, we present sample result for the gender-based subReddits r/AskMen and r/AskWomen. We consider the keyword ‘loneliness’. Positive bias indicates bias towards r/AskMen, and negative bias indicates bias towards r/AskWomen.

Figure 3 shows results for the keyword ‘loneliness’. The original keyword 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 keywords returned by FairKR 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’.

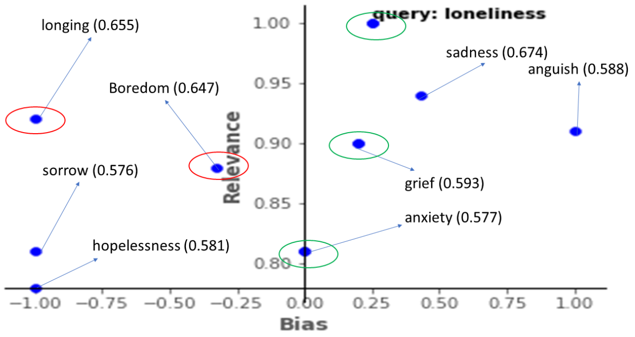

Figure 3: pareto front for word ’loneliness’

5.3 Nationality

Finally, we show results on the nationality-based subReddits r/AskABrit and r/AskAnAmerican. Positive bias values indicate more results from r/AskAnAmerican, and negative values indicate more results from r/AskABrit. We used the keyword ‘government’.

Results for the query ‘government‘ are shown in Figure 4 . The keyword itself returns results disproportionately from the British subReddit (by a 3:2 ratio). In addition to the original keyword, the Pareto front contains the word ‘authorities’, as this returns documents that are evenly balanced between the subReddits. On the opposite side, the word ‘governments’ gives results exclusively from r/AskAnAmerican. The word ‘administration’ shows strong bias towards r/AskAnAmerican, and is also highly relevant to ‘government’.

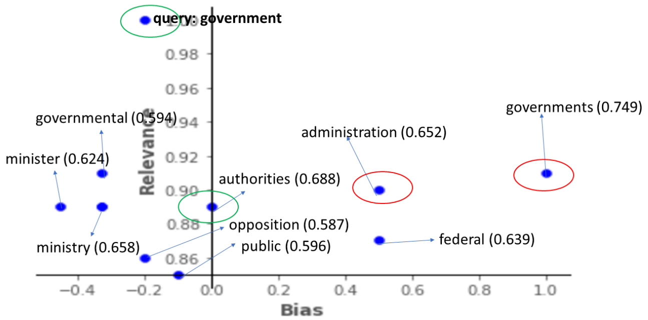

Figure 4: pareto front for word ’government’

6 Conclusion and Future Work

In this paper, we considered the problem of *fair keyword recommendation*, in which the goal is to suggest less-biased keyword alternatives to a query keyword entered by a user. To address this problem, we proposed FairKR, an algorithmic framework for identifying highly-relevant but less-biased keyword alternatives. A major application of this problem is on web search, where fair keyword recommendation can be a step in addressing problems caused by echo chambers or filter bubbles. We also hope that it will be useful on career-related social media sites (like LinkedIn), where recruiters may struggle to find a diverse applicant pool simply because of the keywords that they are using, and applicants may struggle to be found because of how they have written their profiles. Through experiments on posts from Reddit.com, we demonstrated the use of FairKR. We qualitatively analyzed posts on three keywords across three pairs of subReddits based on political alignment, gender, and nationality.

In our future work, we plan to extend this algorithm to propose alternatives to a complete search query, not just an individual keyword. Suggesting similar queries which will give relevant and less biased results will be of great help in getting a complete context when researching a topic and it can also be helpful in news-related searches. We are also considering the problem of dealing with bias along multiple axes.

References

[Blaubergs, 1978] Maija S. Blaubergs. Changing the sexist language: The theory behind the practice. *Psychology of Women Quarterly*, 2(3), 1978.

[Bolukbasi *et al.*, 2016] 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. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[Dev *et al.*, 2020] Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings. In *AAAI*, pages 7659–7666, 2020.

[Dutta, 2002] Rabindranath Dutta. System, method, and program for ranking search results using user category weighting, June 20 2002. US Patent App. 09/737,995.

[Geyik *et al.*, 2019] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 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*, pages 2221–2231, 2019.

[Juhi Kulshrestha and Gummadi, 2017] Motahhare Eslami Saptarshi Ghosh Johnnatan Messias Juhi Kulshrestha, Muhammad B. Zafar and Krishna P. Gummadi. Quantifying search bias: Investigating sources of bias for political searches in social media. 2017. https://arxiv.org/pdf/1704.01347.pdf.

[Meike Zehlike and Baeza-Yates., 2018] Carlos Castillo Sara Hajian Mohamed Megahed Meike Zehlike, Francesco Bonchi and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. 2018. https://arxiv.org/pdf/1706.06368.pdf.

[Zehlike and Castillo, 2020] 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.

[Zehlike *et al.*, 2020] Meike Zehlike, Tom Sühr, Carlos Castillo, and Ivan Kitanovski. Fairsearch: A tool for fairness in ranked search results. In *Companion Proceedings of the Web Conference 2020*, pages 172–175, 2020.