UM ESTUDO COMPARATIVO DE TÉCNICAS DE JUSTIÇA EM ALGORITMOS DE RANKING

BRENO DE SOUSA MATOS

UM ESTUDO COMPARATIVO DE TÉCNICAS DE JUSTIÇA EM ALGORITMOS DE RANKING

Monografia científica apresentada como requisito para a conclusão da Graduação em Ciência da Computação da Universidade Federal de Minas Gerais.

Orientador: Rodrygo Luis Teodoro Santos

Belo Horizonte

Outubro de 2020

de Sousa Matos, Breno

D1234p Um estudo comparativo de técnicas de justiça em algoritmos de ranking / Breno de Sousa Matos. — Belo Horizonte, 2020
xii, 20 f. : il. ; 29cm

Monografia — Universidade Federeal de Minas Gerais
Orientador: Rodrygo Luis Teodoro Santos

1. **Introdução** 2. **Abordagens Modernas** 3. **Técnicas de Justiça para Algoritmos de Ranking** 4. **Conclusão**

CDU 519.6*82.10

Para Marcos e Rosângela.

Agradecimentos

A minha família pelo apoio incondicional. Aos amigos que melhoram até os piores dias.

"If I have seen further it is by standing on the shoulders of Giants."
(Isaac Newton)

Resumo

Palavras-chave: Fairness, Information Retrieval, Learning to Rank.

Algoritmos de *ranking* são amplamente utilizados no mundo moderno: para recomendar produtos em lojas *on-line*, apresentar páginas da *web* em buscadores, ou até sugerir perfis de trabalho para se conectar em redes sociais profissionais. Entretanto, se não acompanhados de forma cautelosa, os resultados dos *rankings* gerados podem conter vieses: como de etnia ou orientação sexual, por exemplo. Neste trabalho, é apresentado um estudo da literatura que aborda técnicas de justiça em algoritmos de *ranking*.

Abstract

Palavras-chave: Fairness, Information Retrieval, Learning to Rank.

Ranking algorithms are widely used nowadays: to display web pages as search engines' results, recommend products on online stores, or even to suggest which profiles to connect to on professional networks. However, if not closely monitored, these algorithms may produce biased rankings, based on features such as ethnicity or sexual orientation. In this work, we propose a study of fair ranking concepts and algorithms.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Algoritmos de ranking possuem grande importância no mundo moderno: estão presentes nos mais variados tipos de *website*, como serviços de *streaming*, sites de compras online, redes sociais, entre outros. Além disso, estão presentes nos buscadores da Internet, como Google[1] e Bing,[2] que facilitam a navegação online, trazendo à tona os resultados mais relevantes às buscas dos usuários.

Sendo parte integrante e ubíqua do mundo digital moderno, estes algoritmos não estão isentos de falhas e erros: um problema existente é a a presença de viés nos *rankings* gerados, que pode variar desde penalizar notícias de veículos pequenos, no caso de *ranking* de notícias, até discriminar candidatos em redes sociais de trabalho, como Linkedin,[3] baseado em etnia e gênero. Para tentar solucionar estes problemas, existem abordagens propostas na literatura [7] que tentam promover justiça em *rankings*.

O estudo de algoritmos de *ranking* existe há décadas, entretanto, os avanços no desenvolvimento de algoritmos focados em justiça ainda são recentes, mesmo este sendo um problema pertinente e bastante atual.

Justiça, no contexto de *rankings*, pode ser definida como gerar listas de objetos de interesse do usuário (documentos, imagens ou perfis de trabalho, por exemplo) de forma indiferente a variáveis sensíveis presentes nos dados (como informações sobre gênero, etnia e idade).

Neste trabalho, iremos apresentar um estudo comparativo de técnicas de justiça em algoritmos de *ranking*. Primeiramente, iremos realizar uma revisão bibliográfica de diversos artigos da área. Serão definidos conceitos fundamentais que servirão de base para, em seguida, apresentar algumas das abordagens e algoritmos mais relevantes para

[1] https://google.com
[2] https://www.bing.com/
[3] https://about.linkedin.com/

geração de *rankings* justos.

Este trabalho está dividido em dois capítulos principais. O primeiro apresenta conceitos fundamentais de recuperação de informação, realizando uma breve introdução a métricas de avaliação e tipos de algoritmos de *ranking*, desde modelos clássicos até abordagens modernas de *learning to rank*.

O segundo capítulo define conceitos elementares para justiça em *ranking*, apresenta métricas de avaliação focadas em justiça, assim como alguns dos algoritmos mais relevantes da área. Por fim, apresenta os resultados da competição TREC 2019 Fair Ranking Track,[4] que é focada em busca acadêmica.

[4]https://fair-trec.github.io/2019/index.html

Capítulo 2

Abordagens Modernas

2.1 Motivação

Este trabalho busca realizar um estudo bibliográfico de técnicas de justiça em algoritmos de *ranking*. Entretanto, antes de detalhar as técnicas e abordagens disponíveis para esse fim, é necessário compreender conceitos fundamentais de recuperação de informação.

2.2 Modelos de Ranking Convencionais

Na literatura disponível, podemos distinguir os modelos de ranking em dois grandes grupos [1]: *query-dependent models* e *query-independent models*, abordados nas subseções a seguir.

2.2.1 Query-dependent Models

Esta categoria de modelos utiliza as informações fornecidas nas *queries* para gerar rankings. A exemplo, o modelo Booleano (tradução livre para *Boolean Model*) [1], que busca predizer se um documento é relevante a uma *query*, mas não é robusto o suficiente para atribuir ao documento um grau de relevância.

Exitem, porém, outros modelos, como o Vector Space Model (VSM) [19], buscam atribuir relevância aos documentos retornados pelas *queries*. Basicamente, tanto *queries* como documentos são representados como vetores em um espaço Euclidiano, utilizando o produto interno entre dois vetores para medir sua similaridade. Para obter boas representações para *queries* e *corpus* de documentos, é possível utilizar técnicas

como TF-IDF [1], que atribui, para cada palavra do *corpus*, um valor numérico que representa quão importante esta é dentro de um documento.

Nesse mesmo grupo, também é possível citar a família de modelos BM25 [1]. A ideia central deste grupo é criar *rankings* de documentos calculando o logit [9] de sua relevância.

2.2.2 Query-independent Models

Esta categoria compreende modelos de que geram *rankings* baseando-se apenas na importância intrínseca dos documentos em questão. Ou seja, não dependem de informações e características presentes nas *queries* realizadas para gerar *rankings*. Um exemplo de modelo desse grupo é o algoritmo PageRank [20], que avalia a relevância de páginas da web, medindo o interesse e atenção de usuários da internet sobre estas páginas. Este algoritmo foi o primeiro utilizado pelo buscador Google[1] para ordenação de resultados de pesquisas.

2.3 Métricas de Avaliação

Um parte crucial da geração de rankings é a definição de métricas e mecanismos de avaliação. Um procedimento padrão pode ser visto na Figura 2.1:

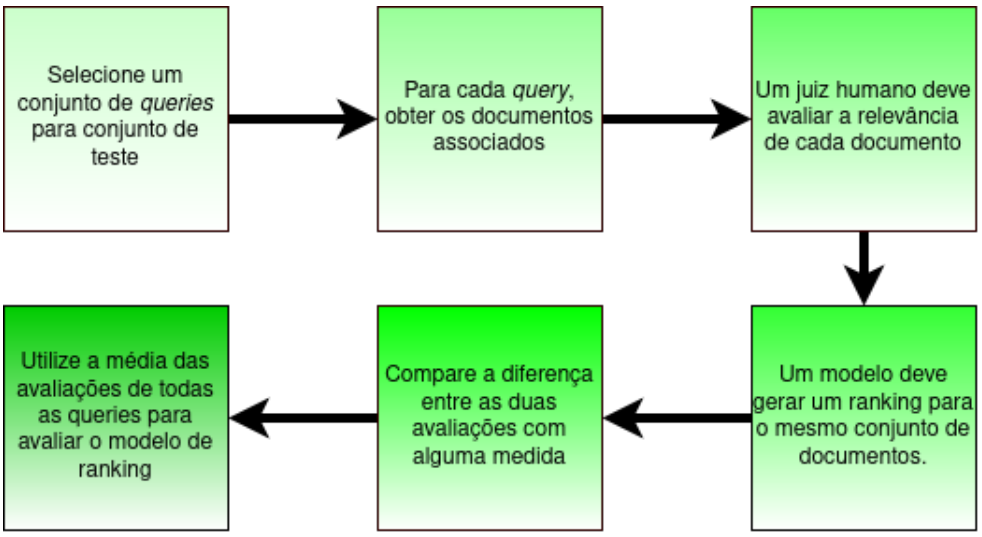

Figura 2.1. Metodologia de avaliação

Para que um juiz humano possa avaliar a relevância de cada documento associado a uma *query*, podem ser utilizadas as seguintes técnicas [19]:

[1] https://about.google/

- Especificar se um documento é relevante de forma binária (0 ou 1) ou definindo graus de relevância (com classes predefinidas).
- Comparar, de forma relativa, a relevância dos documentos. Esta técnica captura preferências relativas.
- Gerar ordem total ou parcial do conjunto de documentos.

Para avaliar a qualidade dos rankings gerados por um modelo de ranking, existem algumas métricas avaliativas disponíveis, listadas a seguir.

2.3.1 Rank Correlation (RC)

Calcula a correlação entre o ranking gerado pelo modelo e o julgamento de relevância dos documentos feito por juízes humanos.

2.3.2 Mean Reciprocal Rank (MRR)

Para um conjunto de documentos associados a uma *query* q, seu *reciprocal rank* é o inverso multiplicativo do índice do primeiro documento relevante. Ou seja, sendo Q a quantidade total de *queries*, dada uma *query* q, o primeiro elemento relevante *reciprocal ranking* será $rank_i$. Logo, a fórmula para o MRR é:

$$MRR = \frac{1}{Q} \times \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{2.1}$$

2.3.3 Mean Average Precision (MAP)

Calcula a média do valor de *precision* [1] para todos os resultados de *queries*. Dessa forma, podemos definir a MAP de um conjunto de queries Q como:

$$MAP = \frac{1}{Q} \times \sum_{q=1}^{Q} AverageP(q) \tag{2.2}$$

2.3.4 Discounted Cumulative Gain (DCG)

O *cumulative gain* é calculando somando as relevâncias dos k documentos presentes no ranking. Por sua vez, o DCG é calculado penalizando o ganho da relevância dos documentos de acordo com sua posição no *ranking*, partindo da premissa de que documentos

relevantes que apareçam em posições mais baixas no *ranking* devem ser penalizados. A Equação 2.3 apresenta a fórmula para o DCG acumulado em uma posição k do *ranking*:

$$DCG_k = \sum_{i=1}^{k} \frac{r_i}{\log_2(i+1)} \tag{2.3}$$

Também é possível utilizar a versão normalizada do DCG: nDCG, ou *Normalized discounted cumulative gain*, obtida ao dividir os valores obtidos pelo maior de todos. Sendo $IDCG_k$ o valor ideal de DCG na posição k, o nDCG acumulado em uma posição k pode ser calculado conforme a Equação 2.4:

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{2.4}$$

2.4 Learning to Rank

Os modelos apresentados anteriormente requerem ajustes de parâmetros, o que não é uma tarefa trivial [19]. Até mesmo um modelo com parâmetros corretamente ajustados pode apresentar performances ruins quando apresentado a dados novos, apresentando *overfit* [19].

Devido à dificuldade desta tarefa, técnicas de aprendizado de máquina se tornam bastante úteis no ajuste dos parâmetros dos modelos, pois possuem grande capacidade de generalização.

Dessa forma, abordagens que utilizam bases de aprendizado de máquina se tornaram cada vez mais utilizados e apresentados na literatura. Do uso crescente dessas abordagens para treinar modelos de *ranking*, surge a área de *learning to rank*.

Learning to rank se refere à utilização de técnicas de *machine learning* para treinar modelos em uma tarefa de *ranking*, sendo muito útil para diversas aplicações de recuperação de informação, processamento de linguagem natural e mineração de dados [17].

Em [19], abordagens de *Learning to Rank* são divididas em três categorias: *pointwise*, *pairwise* e *listwise*, que serão descritas em subseções seguintes.

2.4.1 Abordagem Pointwise

Para esta abordagem, as entradas são compostas por representações vetoriais dos documentos da coleção, capturando as *features* de cada um. As saídas são graus de

relevância para cada documento.

As hipóteses desta abordagem são funções que recebem como entrada os vetores que representam documentos e predizem o grau de relevância de cada um. A *loss function* para abordagens *pointwise* avalia a acurácia da predição feita pela hipótese. Modelos do tipo *pointwise* podem ser de regressão ou classificação, por exemplo. Portanto, a *loss function* deve estar de acordo com a modelagem.

Como exemplo de algoritmos, é possível citar Pranking [10], SLR [8], McRank [18] e CRR [22].

2.4.2 Abordagem Pairwise

Nesta abordagem, as entradas consistem de pares de documentos representados por vetores de *features*. As saídas são pares de valores entre -1 e 1 que denotam preferência mínima e máxima sobre as entradas.

O conjunto de hipóteses desta abordagem é composto de funções que avaliam os pares de documentos e predizem a ordem relativa entre eles. Por fim, a *loss function* avalia a diferença entre o par predito e a preferência real.

Como exemplo de algoritmos, é possível citar MART [14], RankNet [4] e RankBoost [13].

2.4.3 Abordagem Listwise

As entradas dos modelos deste tipo de abordagem são conjuntos de documentos associados a uma *query* q.

Como saída, existem dois tipos possíveis:

- Grau de relevância de todos os documentos associados a q
- Lista ordenada, por relevância, dos documentos associados a q

Vale notar que são apenas formas diferentes de representação, mas que são interpretadas de forma similar.

Como existem dois tipos de saídas possíveis, existem também dois tipos de hipóteses: as que recebem como entrada uma lista de documentos e predizem a relevância de cada um, ou as que predizem a ordem dos documentos.

Como *loss function*, podem existir funções que avaliam (utilizados métricas como as descritas em subseções anteriores) a relevância de documentos, ou que avaliam a diferença entre a ordem predita para os documentos e a ordem real.

Como exemplo de algoritmos, existem ListNet [5], RankGP [26] e BoltzRank [24].

Capítulo 3

Técnicas de Justiça para Algoritmos de Ranking

Neste capítulo, serão abordados alguns do principais avanços na área de justiça em algoritmos de *ranking*.

3.1 Motivação

Nos últimos anos, a perspectiva de uso de algoritmos de *ranking* tem passado por mudanças. Tradicionalmente, a preocupação desta área é de entregar aos usuários as informações mais relevantes, de forma mais rápida, a qualquer momento. Entretanto, além de encontrar itens de utilidade máxima, máquinas de busca modernas também são utilizadas para encontrar pessoas, como em contextos profissionais [15][16], trazendo à tona novas discussões sobre justiça e transparência na área [6].

Apesar de ser um problema atual, considerado de grande importância para o avanço da área de recuperação de informação [11], os esforços para alcançar *rankings* mais justos ainda são menos desenvolvidos que, por exemplo, os de áreas relacionadas, como aprendizado de máquina.

3.2 Conceitos Básicos

É necessário definir alguns conceitos fundamentais acerca de justiça em *rankings*, apresentados a seguir.

3.2.1 Rankings Justos

Em [6], é proposto que um *ranking* justo deve ter, no mínimo, as seguintes características:

- Presença suficiente de itens de grupos diferentes, em especial grupos considerados protegidos (socialmente desfavorecidos). A ideia por trás é evitar discriminação estatística de grupos específicos, garantindo que todos estejam presentes em quantidade suficientemente expressiva.
- Tratamento consistente para itens similares. A ideia por trás é que não haja discriminação individualizada.
- Representação adequada de itens, em especial, itens de grupos protegidos. O objetivo é mitigar danos representacionais, como presença de estereótipos em sugestões de *queries* em máquinas de busca online.

3.2.2 Diversidade e Justiça

Apesar de ser um conceito bastante utilizado em recuperação de informação, diversidade não significa o mesmo que justiça, sendo necessário fazer a diferenciação dos termos.

Ao elaborar um modelo de *ranking*, diversidade de resultados é muito importante, pois torna o modelo mais robusto e útil. Entretanto, diversidade se preocupa com a utilidade do item para o usuário, enquanto justiça foca na perspectiva do item retornado pelo modelo de *ranking*. Por exemplo, no contexto de busca de perfis em redes sociais profissionais, diversidade se preocupa em garantir que haja múltiplos perfis úteis ao usuário que realiza a *query*, enquanto justiça busca garantir que haja, na lista de perfis retornada, diversidade étnica, de gênero ou idade, garantindo que membros de classes tradicionalmente sub-representadas estejam presentes nos *rankings*.

Justiça pode ser vista como um conceito assimétrico no contexto de recuperação de informação, pois parte do pressuposto de que existem diferentes grupos, com diferentes graus de representação, mas que devem ser incluídos nos *rankings* sem discriminação.

3.3 Métricas de Avaliação

Além das métricas apresentadas na Seção 2.3, existem métodos específicos para avaliar modelos de *ranking* que buscam garantir, além de utilidade dos itens retornados, justiça

para os diferentes grupos presentes nos dados.

Existe dois grupos principais de métodos para avaliar justiça em *rankings*: baseados em exposição e baseados em probabilidade.

3.3.1 Métodos Baseados em Exposição

Métodos deste grupo tentam mensurar a atenção que o usuário dá a diferentes itens, seja estimando uma distribuição de probabilidade ou utilizando algum tipo de *feedback*, como quantidade de cliques ou mapa de calor da exibição do *ranking*.

Justiça de exposição pode ser definida da seguinte forma: seja uma matriz $R_{m \times n}$ a representação das probabilidades de m itens estarem em n posições de um *ranking*. Seja um item d_i, com $i \in [m]$ e uma posição $j \in [n]$. A exposição de um item qualquer na posição j do *ranking* é representada por e_i e pode ser obtida de formas empíricas.

Seja u_i ($i \in [m]$) a representação da utilidade do item d_i. Sejam dois grupos de itens: G_0 representando o grupo em vantagem e G_k representando o grupo protegido (em desvantagem), com $k \geq 1$. Com esta definição de grupos, é possível levar em consideração diversos grupos protegidos distintos na avaliação da exposição.

É possível definir a exposição E do grupo protegido G_k em um ranking R como a média da exposição de seus itens membros:

$$E(G_k|R) = \frac{1}{|G_k|} \cdot \sum_{i:d_i \in G_k} \sum_{j=1}^{n} R_{i,j} e_j \tag{3.1}$$

Por sua vez, a utilidade do grupo G_k pode ser definida como a utilidade média de seus itens membros:

$$U(G_k) = \frac{1}{|G_k|} \cdot \sum_{i:d_i \in G_k} u_i \tag{3.2}$$

Com estas definições, é possível então definir justiça de exposição entre dois grupos. Esta é alcançada quando a razão entre a exposição e a utilidade dos dois grupos é a mesma. Por exemplo, para os grupos G_0 (em vantagem) e G_k (algum grupo protegido), a justiça de exposição é alcançada quando:

$$\frac{E(G_0|R)}{U(G_0)} = \frac{E(G_k|R)}{U(G_k)} \tag{3.3}$$

Além da métrica proposta na Equação 3.3, também existem outras formas de quantificar a exposição de grupos distintos, como rND (*Normalized discounted difference*), rRD (*Normalized discounted ration*) e rKL (*Normalized discounted KL-*

divergence), propostos em [25] e com código diponível publicamente.[1] Breves descrições dos métodos podem ser vistas a seguir:

- **rND** : Calcula a diferença entre a proporção de itens de um grupo protegido nos top-i itens e no coleção completa. Os valores calculados são somados em posições discretas do ranking e é aplicado um desconto logarítmico nos valores, que por fim são normalizados em função do maior rND obtido.

- **rKL** : Utiliza a divergência de Kullback-Leibler para computar o valor esperado da diferença entre a presença de itens de grupos protegidos nos top-i itens e na coleção completa. Uma vantagem deste método é que pode ser utilizado com mais de um grupo protegido, expandindo além de classificações binárias propostas no rND.

- **rRD** : Similar ao rND, mas com formulação matemática ligeiramente diferente, levando em consideração a razão entre o tamanho do grupo protegido e o grupo em vantagem.

3.3.2 Métodos Baseados em Probabilidade

Este grupo de métodos pressupõe um *ranking* gerado por um processo aleatório, para, em seguida, medir a diferença entre os atributos esperados para este *ranking* e os observados como saída do modelo que está sendo avaliado.

Em [25], os autores propõem a seguinte metodologia:

1. Itens de cada grupo protegido G_k são divididos em subconjuntos ordenados por utilidade, de forma decrescente. Dessa forma, são gerados rankings.

2. Caso exista apenas um grupo protegido (G_1), é realizado um experimento de Bernoulli [21] para cada posição $j \in [n]$ do ranking. Se o ensaio retorna verdadeiro, o primeiro elemento de G_1 é selecionado. Caso contrário, o primeiro elemento de G_0 é selecionado. Este processo encerra quando n elementos são selecionados.

3. Para os dois casos anteriores (1 ou k grupos protegidos), é pressuposto que o processo descrito gera um *ranking* justo, com representatividade equivalente para todos os grupos. Por fim, para um *ranking* gerado pelo modelo que está sendo avaliado, é realizado um teste estatístico para aferir se existem grupos sub-representados ou não, utilizando o índice j como referência.

[1] https://github.com/DataResponsibly/FairRank

3.4 Abordagens Modernas

É possível dividir as abordagens de justiça em *ranking* em três categorias: pré-processamento, pós-processamento e em processamento.

3.4.1 Pré-processamento

Por se tratarem de abordagens pré-processamento, buscam reduzir o viés dos dados que serão utilizado para treino de algum modelo de *ranking*. Por exemplo, buscando garantir que grupos protegidos não são sub-representados no conjunto de treino.

3.4.2 Em processamento

Métodos deste conjunto geralmente realizam a otimização de duas funções de perda.

No algoritmo de *ranking list-wise* DELTR [28], a função de perda L captura o erro de treino, comparando, para uma query q, os *rankings* gerados pelo modelo ($\hat{y}^{(q)}$) e as saídas reais ($y^{(q)}$). A função U captura o erro relacionado à exposição dos grupos protegidos nos *rankings* produzidos. Otimizando as duas funções, é possível obter *rankings* justos e com boa utilidade.

Na Equação 3.4, é possível observar a formulação matemática para função de perda do algoritmo DELTR.

$$L_{DELTR}\left(y^{(q)}, \hat{y}^{(q)}\right) = L\left(y^{(q)}, \hat{y}^{(q)}\right) + \gamma U\left(\hat{y}^{(q)}\right) \tag{3.4}$$

O parâmetro γ é um fator de desconto que controla a troca entre acurácia (em relação ao conjunto de treino) e exposição.

A função U pode ser observada na Equação 3.5, em que G_1 representa o grupo protegido e G_0 o grupo em vantagem. Para esta equação, a formulação de exposição é a mesma apresentada na Equação 3.3.

$$U\left(\hat{y}^{(q)}\right) = max\left(0, E(G_1|P_{\hat{y}^{(q)}}) - E(G_0|P_{\hat{y}^{(q)}})\right)^2 \tag{3.5}$$

Para os cenários de teste propostos pelos autores,[2] DELTR obteve resultados superiores, em termos de exposição e relevância, que abordagens que utilizam o algoritmo FA*IR [27].

[2]https://github.com/MilkaLichtblau/DELTR-Experiments

3.4.3 Pós-processamento

Por fim, existe o grupo de métodos pós-processamento, que buscam reordenar os itens de um *ranking* de acordo com restrições pré-definidas. Este grupo de técnicas compreende a maioria das abordagens propostas na literatura.

Um dos principais conceitos deste grupo é de *amortized fairness* [3], em que a exposição de um item é distribuída de forma justa por diversos *rankings*. O objetivo é equilibrar a exposição e a relevância de um item, de modo que a qualidade dos *rankings* obtidos seja preservada. Na Figura 3.1, é possível observar um exemplo do conceito: os *scores* de relevância dos itens em cada *ranking* diferente têm valores muito próximos. Entretanto, em um cenário realista, os itens no topo do *ranking* serão priorizados pelos usuários, mesmo que todos possuam relevância similar. A ideia de *amortized fairness* é que todos os itens relevantes sejam apresentados no topo do *ranking* de tempos em tempos.

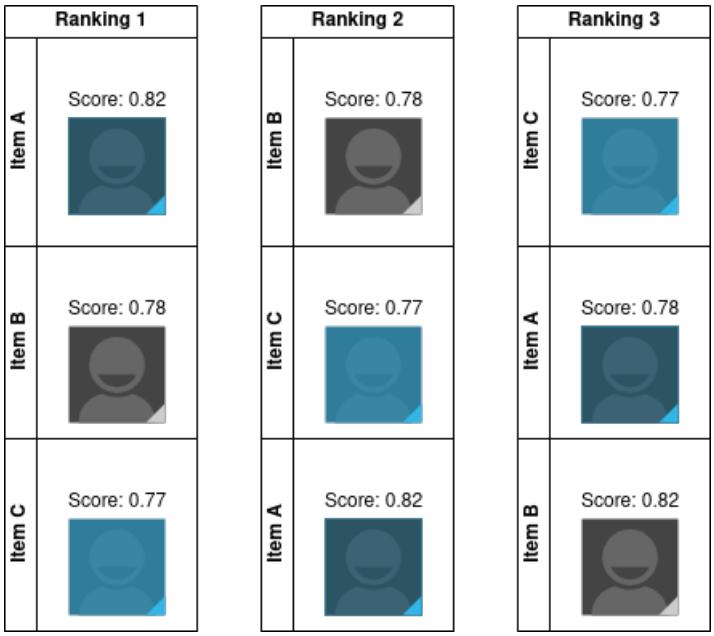

Figura 3.1. Exemplo de rankings para *amortized fairness*

Em [23], os autores propõem que a exposição de um item deve ser proporcional à sua utilidade/relevância, realizando ajustes da quantidade de exposição (seja esta em excesso ou escassez) de um item de acordo com a frequência que aparece nos *rankings*. A ideia é reordenar os itens conforme o sistema aprende quais estão sendo expostos de forma excessiva ou escassa, respeitando a restrição de que exista um limite mínimo de utilidade dos itens apresentados.

3.4.3.1 FA*IR

Uma das técnicas pós-processamento mais relevantes é o algoritmo FA*IR [27], que gera *rankings top-k* justos.

Sejam dois grupos de itens P (itens com características protegidas) e V (itens em vantagem), o algoritmo pode ser sumarizado da seguinte forma:

1. Gerar *rankings* separados para P e V.
2. Definir a quantidade mínima de itens de P para cada posição do *ranking* de acordo com testes estatísticos.
3. Para cada posição, se já existe a quantidade mínima de itens de P, escolher o mais relevante entre melhores de P e V restantes. Caso contrário, escolher o próximo item de P.

3.4.3.2 Ranking With Fairness Constraints

Em [7], o problema de obter rankings justos para múltiplos grupos protegidos é modelado como uma programação inteira, buscando maximizar a expressão:

$$\sum_{i \in [m], j \in [n]} P_{i,j} U_{i,j} \tag{3.6}$$

Seja uma matriz $P_{m \times n}$ que representa, para m itens e n posições possíveis, as permutações dos itens nas posições do *ranking*. Se um item d_i está na posição j, temos $P_{i,j} = 1$. Para cada posição do *ranking* e cada grupo $k \geq 0$, existem limites máximos ($UB^{max}_{k,l}$) e mínimos ($LB^{min}_{k,l}$) da quantidade de itens do grupo G_k que devem, necessariamente, estar presentes entre os primeiros l elementos do *ranking*.

Por sua vez, a matriz $U_{i,j}$ indica a utilidade de posicionar o item d_i na posição j do *ranking*. Sendo assim, considerando os limites mínimos e máximos já definidos, é possível utilizar a formulação da Expressão 3.6 para encontrar o resultado ótimo.

Entretanto, é mais pertinente aplicar relaxamentos nas restrições da formulação para obter uma solução aproximada de forma eficiente, visto que a formulação original do problema é da classe NP-difícil.

3.5 TREC 2019 - Fair Ranking Track

Os esforços para tornar sistemas de recuperação de informação e, especialmente, algoritmos de *ranking*, mais justos, são relativamente recentes na literatura [6]. Proporci-

onalmente, formas de avaliar esses sistemas e algoritmos, em termos de justiça, ainda são escassas.

Com isto em mente, a trilha de justiça da conferência TREC[3] propôs, à partir de 2019, uma competição de algoritmos de *ranking* voltada à justiça. Esta seção faz um breve resumo da competição [2].

3.5.1 Motivação e Descrição da Tarefa

Os objetivos da competição, como descritos por seus organizadores [2], podem ser resumidos em:

- Promover o desenvolvimento de algoritmos de *ranking* justos
- Fornecer um conjunto de dados para avaliar estes algoritmos em termos de justiça
- Desenvolver métricas para exposição justa, seja para indivíduos ou grupos, em tarefas de recuperação de informação.
- Elaborar um protocolo de experimentação para justiça em algoritmos de *ranking*.

A tarefa escolhida para a competição foi de *reranking* no contexto de busca acadêmica, focando em promover exposição justa para diversos grupos de autores, utilizando como base de dados o Open Corpus do Allen Institue for Artificial Intelligence.[4]

Em resumo, os competidores receberam um conjunto de *queries* com os documentos correspondentes a serem reordenados, exemplos de definições de grupos, além do corpus supracitado.

[3] https://trec.nist.gov/
[4] api.semanticscholar.org/corpus/

3.5.2 Abordagens Submetidas

A primeira realização contou com a participação de 5 instituições. Na Tabela 3.1, um breve resumo das abordagens utilizadas por cada uma.

Instituição	Abordagem
ICTNET	Uiliza BERT [12] para gerar *embeddings*. Não usa modelagem explícita para justiça
IR-Cologne	*Learning to rank* sem modelagem explícita para justiça
MacEwan University School of Business	Gera *rankings* realizando combinação ponderada de resultados de busca utilizando diversos campos distintos.
University of Padova	Modela distribuições de justiça e injustiça em grupos
University of Glasgow	Diversas abordagens utilizando diversificação de resultados

Tabela 3.1. Resumo das abordagens utilizadas na competição

Capítulo 4

Conclusão

A crescente utilização de algoritmos de *ranking* no dia-a-dia torna necessário reavaliar o papel que estas ferramentas desempenham na sociedade. Quando seu uso tradicional é atualizado e pessoas passam a ser um dos grandes focos destes algoritmos, é necessário tomar medidas para garantir que vieses sociais não sejam replicados por estas ferramentas. Para tentar solucionar este problema, existem diversas abordagens propostas na literatura.

Neste trabalho, foram apresentados conceitos fundamentais de recuperação de informação, utilizados como base para, em seguida, apresentarmos um resumo dos principais conceitos e técnicas de justiça em algoritmos de *ranking*, realizando revisão bibliográfica de diversos artigos relevantes da área, compreendendo algumas das principais abordagens, como FA*IR. Por fim, apresentamos um resumo da primeira competição de *ranking* focada em justiça da conferência TREC.

Em trabalhos futuros, o objetivo é utilizar a base conceitual adquirida para propor um novo algoritmo focado em *rankings* justos.

Referências Bibliográficas

[1] Baeza-Yates, R.; Ribeiro-Neto, B. et al. (1999). *Modern information retrieval*, volume 463.

[2] Biega, A. J.; Diaz, F.; Ekstrand, M. D. & Kohlmeier, S. (2019). Overview of the trec 2019 fair ranking track. In *The Twenty-Eighth Text REtrieval Conference (TREC 2019) Proceedings*.

[3] Biega, A. J.; Gummadi, K. P. & Weikum, G. (2018). Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*, pp. 405--414.

[4] Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N. & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89--96.

[5] Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F. & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pp. 129--136.

[6] Castillo, C. (2019). Fairness and transparency in ranking. *SIGIR Forum*, 52(2):64–71. ISSN 0163-5840.

[7] Celis, L. E.; Straszak, D. & Vishnoi, N. K. (2017). Ranking with fairness constraints.

[8] Cooper, W. S.; Gey, F. C. & Dabney, D. P. (1992). Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 198--210.

[9] Cramer, J. S. (2003). The origins and development of the logit model. *Logit models from economics and other fields*, 2003:1--19.

[10] Crammer, K. & Singer, Y. (2002). Pranking with ranking. In *Advances in neural information processing systems*, pp. 641--647.

[11] Culpepper, J. S.; Diaz, F. & Smucker, M. D. (2018). Report from the third strategic workshop on information retrieval (swirl).(2018).

[12] Devlin, J.; Chang, M.; Lee, K. & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[13] Freund, Y.; Iyer, R.; Schapire, R. E. & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933--969.

[14] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189--1232.

[15] Geyik, S. C.; Ambler, S. & Kenthapadi, K. (2019). Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. *CoRR*, abs/1905.01989.

[16] Lahoti, P.; Gummadi, K. P. & Weikum, G. (2019). ifair: Learning individually fair data representations for algorithmic decision making. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1334--1345. IEEE.

[17] Li, H. (2011). A short introduction to learning to rank. *IEICE Trans. Inf. Syst.*, 94-D:1854–1862.

[18] Li, P.; Wu, Q. & Burges, C. J. (2008). Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pp. 897--904.

[19] Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331. ISSN 1554-0669.

[20] Page, L.; Brin, S.; Motwani, R. & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Relatório técnico, Stanford InfoLab.

[21] Papoulis, A. & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education.

[22] Sculley, D. (2010). Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 979--988.

[23] Singh, A. & Joachims, T. (2017). Equality of opportunity in rankings. In *Workshop on Prioritizing Online Content (WPOC) at NIPS*, p. 31.

[24] Volkovs, M. N. & Zemel, R. S. (2009). Boltzrank: learning to maximize expected ranking gain. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1089--1096.

[25] Yang, K. & Stoyanovich, J. (2017). Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM '17, New York, NY, USA. Association for Computing Machinery.

[26] yuan Yeh, J.; yi Lin, J.; ren Ke, H. & pang Yang, W. (2007). Learning to rank for information retrieval using genetic programming.

[27] Zehlike, M.; Bonchi, F.; Castillo, C.; Hajian, S.; Megahed, M. & Baeza-Yates, R. (2017). Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1569--1578.

[28] Zehlike, M. & Castillo, C. (2020). Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, pp. 2849--2855.