Leveraging User History with Transformers for News Clicking: The DArgk Approach

Juan Manuel Rodriguez
jmro@cs.aau.dk
Aalborg University
Aalborg, Denmark

Antonela Tommasel
antonela.tommasel@isistan.unicen.edu.ar
CONICET - UNICEN
Tandil, Buenos Aires, Argentina

ABSTRACT

This paper provides an overview of the approach we used as team DArgk[1] for the *ACM RecSys Challenge 2024*. The competition was organized by *Ekstra Bladet* and focused on addressing both technical and normative challenges in designing an effective and responsible online news recommender system. Our proposed method aims to model user preferences based on implicit behavior while considering the news agenda's dynamic influence and the news items' rapid decay. We employed deep learning models to estimate the likelihood of a user clicking on a list of articles seen during a specific timeframe. To this end, we proposed a transformer-based model capable of encoding user reading history to rank articles according to the user preferences with a focus on beyond accuracy performance for users with different preferences than the average user. Our submission achieved the 2nd rank and overall score of 0.7709 in the competition academia-track final results. We release our source code at: https://github.com/dkw-aau/RecSys2024Challenge.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender systems, news, diversity

ACM Reference Format:
Juan Manuel Rodriguez and Antonela Tommasel. 2024. Leveraging User History with Transformers for News Clicking: The DArgk Approach. In *ACM RecSys Challenge 2024 (RecSys Challenge '24), October 14–18, 2024, Bari, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3687151.3687161

1 INTRODUCTION

News consumption has shifted dramatically in recent years, with an increasing share of news being consumed online [7]. Recommender systems have become crucial as algorithmic content curators in media, helping users find relevant content and influencing what news the public does and does not see [20]. These systems play a pivotal role in directing citizens' attention to important information, thereby supporting democratic society [20]. At the same time, for news media, recommenders help them to remain relevant in the global competition for attention, increase engagement with content, inform citizens, and offer services that people are willing to pay for [20]. However, this power to shape reading agendas also introduces new risks and responsibilities.

Traditionally, the evaluation of recommender systems has focused on maximizing accuracy, evaluated through metrics measuring the accuracy/relevance of recommended items by comparing them to users' consumption/click history [6]. Consequently, recommender systems often provide users with personalized items similar to those they have previously shown interest in. While this approach may increase click rates, it does not necessarily encourage users to explore new and diverse content [4].

Diversity refers to the idea that users can interact with information from their surroundings from multiple sources exposing different viewpoints so they can make balanced and informed decisions [8]. On the contrary, users with little or no exposure to diverse (or even contradicting) views can become unintendedly trapped in isolated filter bubbles [4, 14]. These bubbles lead to self-reinforcing patterns of narrowing content exposure causing user segregation and other biases [1, 13]. Enhancing recommendation diversity and novelty is linked to increased fairness and reduced biases [17].

Fairness has become a crucial topic in recent years [3, 12]. Different fairness notions in the literature reflect normative ideas about how a recommender system should behave [3]. One common distinction is between individual and group fairness [3]. Individual fairness requires treating similar individuals similarly, though defining task-specific similarity metrics for individuals might be challenging. Group fairness, on the other hand, requires that protected groups are treated similarly. Most group fairness studies group users by sensitive features (e.g., gender, race, ethnicity, religion). However, group definition can also be based on user activity (e.g., number of interactions, cumulative ratings) [5, 12] or preferences (e.g., distribution of item consumption across categories).

In this work, we focused on fairness for users with *atypical* behavior (i.e., users whose reading profile focus on article categories that are not popular across the average user. In this regard, we evaluated the capability of our model to adapt to the user's atypicality while preserving novelty and diversity in the recommendations. Since our model is based on a transformer architecture that represents users mostly based on their reading history, we argue that the model can easily represent atypical users.

2 PROBLEM FORMULATION

The *ACM RecSys Challenge*[2] was organized by *Ekstra Bladet*, a Danish newspaper. The digital edition presents users with a set of

[1] DArgk stands for the authors' base countries: Denmark and Argentina.

[2] http://www.recsyschallenge.com/2024/

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

RecSys Challenge '24, October 14–18, 2024, Bari, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1127-5/24/10
https://doi.org/10.1145/3687151.3687161

potentially relevant articles. The goal is to predict which of these articles the user will click. The task consists of creating a model that captures users' preferences and uses them to rank the presented articles based on clicking likelihood. To this end, the model has access to a 21-day history of users' reading habits, demographic information, and session information (used device and time).

Data collection.
The dataset is provided by *Ekstra Bladet*, and it is called "*Ekstra Bladet News Recommendation Dataset (EB-NeRD)*". The dataset contains information from April 27 to June 8, 2023, and it is temporally divided into train, validation and test. Each partition contains 21 days of historical data and 7 days of behavior data (used for evaluating task performance). The dataset also includes information related to each article, such as the title, body, category, and date and time of publication.

Evaluation.
The main evaluation metric is based on ROC AUC, using the *scikit-learn* implementation[3]. However, as the reported results are a ranking and not a probability, AUC was computed using the reciprocal rank. For example, let $y_{pred} = [2, 1, 3]$ be a proposed ranking and $y_{true} = [1, 0, 0]$, this is converted into $\hat{y}_{pred} = [\frac{1}{2}, \frac{1}{1}, \frac{1}{3}] = [0.5, 1, 0.33..]$. Hence, the $auc(\hat{y}_{pred}, y_{true}) = 0.5$. To obtain the value for the whole dataset, the ROC AUC for all the predictions is averaged. In addition to the main metric, MRR, NDCG@5, and NDCG@10 are reported.

The testing dataset includes 200 000 beyond accuracy instances without ground truth for evaluating metrics [9], such as diversity, novelty, and coverage. This instance consists of different users who are presented with the same 250 articles. The challenge organizers provide novelty, diversity, coverage, and serendipity for these instance as defined in [9].

$$cb - novelty\,(R, H) = \frac{1}{|R|\,|H|} \sum_{i \in R} \sum_{j \in H} d(i, j) \tag{1}$$

$$cb - diversity\,(R) = \frac{1}{|R|\,(|R| - 1)} \sum_{i \in R} \sum_{j \in R} d(i, j) \tag{2}$$

Finally, we analyzed user fairness in content-based novelty and diversity (eq. (2)) [16] (*cb-novelty* and *cb-diversity*) in the ground truth recommendations. Where H is the user reading history, R is the set of recommended articles, and $d(i, j) = 1 - sim(i, j)$ is the distance between articles and $sim(i, j)$ is the article embedding cosine similarity[4]. We computed the atypicality [23] of a user based on the Jensen–Shannon divergence between the article category distribution preferred by the user and the average article category distribution preferred by all users. For example, if a user reads one article in $C1$ and two articles in $C2$, while the average user reads half of the articles in the $C1$ category and half from the $C2$ category; this user atypicality is $D_{js}\left(\left[\frac{1}{3}, \frac{2}{3}\right], \left[\frac{1}{2}, \frac{1}{2}\right]\right) = 0.119$. In this context, the higher the Jensen–Shannon divergence is, the further the user preferences are from the average user. We characterized recommendations based on their atypicality regarding users' historical preferences and the content-based *cb-diversity*, and *cb-novelty*.

[3]Scikit learn ROC ACU
[4]Similarity based on Sentence Transformer "paraphrase-multilingual-mpnet-base-v2" model

3 DATASET PREPROCESSING

Here, we describe the features used and their preprocessing.
Article data.
The article data presents information about 125 541 news articles. For representing articles, we used RoBERTa text embeddings [2] and an unspecified image embedding provided by EB-NeRD. The text embedding was used as provided, while the image[5] embedding was converted from $\mathbb{R}^{1024}$ to $\mathbb{R}^{128}$ by means of a simple 2-layer encoder of an autoencoder. Since not all articles have an image embedding, we used the zero vector for those cases. We also computed embeddings for the category string using the model "*paraphrase-multilingual-mpnet-base-v2*" provided by Sentence Transformer [15]. Finally, we also considered whether the article was premium (i.e., it is behind a paywall) or not, and the sentiment of the article, namely "Positive", "Neutral", or "Negative". The publication date was used indirectly to create derived behavior and history data features.

History data.
History data consists of an article list, and the date and time when the user accessed each article. We extract the day of the week and the hour of the day as categorical features with 7 and 24 possible values, respectively. Moreover, we enrich the article features by computing the delta time between the reading and publication times. We discretize these values into 100 bins, ensuring that the number of reading events in the training historical data is equally distributed in these bins. As expected, most of the 125 182 942 reading events in training occur close to the publication day, where half of these events occur within 85 minutes of the publication.

Behavior data.
The behavior data contains user data, the used device, the time of the event, and the candidate articles that are shown to the users, which we call *in-view articles*. In addition to this, training and validation datasets have articles' clicked by the user in each interaction.

Users are characterized by whether they are logged, they have a premium membership, gender (feminine, masculine), postcode (metropolitan, rural district, municipality, provincial, big city), and age group (10-year bins). Gender, postcode, and age group can be unknown, which is always the case for users who are not logged in. Around 11.8%[6] of all interactions are from logged users. Moreover, information about gender, postcode, and age group is not available in around 30.2%, 78.9%, and 70.9% of the logged users respectively.

Regarding the interaction itself, we consider the access device and the interaction date and time. The access device can be desktop, mobile, tablet, or unknown. Desktop and mobile access represent 34.0% and 62.9% of the interactions. For the interaction time, we extract the day of the week and the hour of the day. Finally, we enrich the article features by adding the discretized delta time between the interaction and publication dates. We use the same scale for discretizing the articles as defined in "History data".

4 MODEL

Figure 1 depicts our proposed model based on the transformers architecture [19]. We propose using a transformer encoder/decoder

[5]Images were not provided due to copyright limitations.
[6]All percentages correspond to the training dataset. Nonetheless, they are very similar in validation and testing.

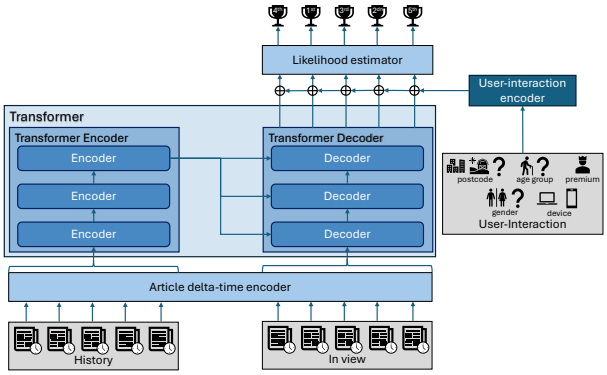

Figure 1: Model architecture.

to estimate the probability of clicking in-view articles given user history. Around the transformer, our model places the "*Article Delta-Time Encoder*" (ADTE) for capturing article information and the moment when they are either read or presented to the user, the "*User-Interaction Encoder*" (UIE) to add demographic information about the user and the type of device they are using to access the site, and the "*Likelihood Estimator*" that produces the final output of the model, i.e., which article is most likely to be clicked.

The ADTE responsibility is to encode the article considering the time delta from its publication and the moment of reading, in the case of the history, or when it is presented to the user. The ADTE is computed independently for all the articles independently from each other, but with shared weights.

$$\begin{aligned} ctx\,(a, ta) &= p_e\,(a) + s_e\,(a) + w_e\,(ta) + h_e\,(ta) \\ art\,(a) &= FF\,(t_e\,(a) + i_e\,(a) + c_e\,(a)) + ctx\,(a) \\ ADTE\,(a, ta) &= art\,(a) + d_e\,(disc\,(ta - a)) \end{aligned} \tag{3}$$

Equation (3) presents the definition of the ADTE component, where a is the article and ta is the time of access to the article, i.e., when it is read or presented in view. $p_e\,(a)$, $s_e\,(a)$, $w_e\,(ta)$, $h_e\,(ta)$ represent the embeddings for the premium, sentiment, access weekday, and access hour. These embeddings are trainable and produce a $\mathbb{R}^d$ vector, where d is the embedding dimension, which is set to 32. $d_e\,(disc\,(ta - a))$ is a trainable embedding for the discretization (see Section 3) of the delta time between access and publication. During training, a 10% dropout is applied to all embeddings. Finally, $FF\,(x)$ is a feed-forward network as defined in eq. (4), all layers in the feed-forward network output a $\mathbb{R}^d$. $t_e\,(a)$, $i_e\,(a)$, and $c_e\,(a)$ are the pre-computed (non-trainable) embedding for the article text, image, and category respectively (see Section 3); and + is a concatenation operation.

$$\begin{aligned} h_i\,(x) &= x \cdot W_i + b_i \\ FF\,(x) &= h_2\,(dropout\,(SELU\,(h_1\,(x)))) \end{aligned} \tag{4}$$

Our model central component is an encoder-decoder transformer. The rationale for using this transformer is two-fold. Firstly, transformers have been proven effective in capturing relations within the user history to make new predictions, particularly in sequence recommendation [18, 22, 24]. Secondly, the attention mechanism allows us to represent items not only based on their features but also on the features of other items. Since the goal is to estimate the likelihood of a user clicking a particular article given a set of articles, i.e., we need to characterize the articles given their context.

Table 1: Evaluation results.

Model	AUC	MRR	NDCG@5	NDCG@10
:D (1st)	0.8933	0.735	0.7923	0.8002
BlackPearl (2nd)	0.8825	0.7165	0.7762	0.7861
Tom3TK (3th)	0.8707	0.7029	0.7631	0.7751
FeatureSalad (Academic 1st)	0.8494	0.6638	0.7296	0.7451
Our model	0.7709	0.5453	0.6125	0.6479
Our model (nft)	0.7388	0.5221	0.5835	0.6234

The proposed transformer has three encode and decode layers with an internal forward representation of 128. In brief, the encoder codifies the user's history, and the decoder characterizes the in-view articles given the user's history.

The UIE is responsible for encoding all the current user and interaction information. Equation (5) presents the component, which adds learnable embeddings for whether users are logged in ($sso_e\,(u)$), their gender ($g_e\,(u)$), their postcode type ($post_e\,(u)$), their age group ($age_e\,(u)$), if they are paying the premium subscription ($sub_e\,(u)$), and the device they are using to access ($dev_e\,(i)$). These embeddings are also defined in $\mathbb{R}^d$. The UIE output is concatenated with the transformer output for each in-view article.

$$\begin{aligned} UIE\,(u, i) &= sso_e\,(u) + g_e\,(u) + post_e\,(u) + \\ & age_e\,(u) + sub_e\,(u) + dev_e\,(i) \end{aligned} \tag{5}$$

"*Likelihood Estimator*" outputs the logits for in-view articles. It consists of a two-layer feed-forward neural network as defined in eq. (4) that is applied per each in-view article. The hidden layer outputs an $\mathbb{R}^d$ vector. The output is in $\mathbb{R}$, i.e., our model outputs a single value per each in-view article. Hence, the output per each particular instance is in $\mathbb{R}^v$, where v is the number of in-view articles.

Training.

For the loss function to fit, we settle for a custom loss based on the Binary Cross Entropy depicted in eq. (6). In particular, $\hat{y}$ are the estimated logits for in-view articles, $|\hat{y}|$ is the number of in-view articles, and C is the clicked article set, which usually contains one element. Given that a median of eight articles are in view and only one is clicked, our loss function prioritizes the loss for the clicked article rather than the non-clicked. This loss function is defined for an instance; for each batch, the loss is the mean of the instance loss.

$$\mathcal{L}\,(\hat{y}, C) = \frac{-\sum_{i \in C} \log\,(\sigma\,(\hat{y}_i))}{|C|} + \frac{-\sum_{j \notin C} \log\,(1 - \sigma\,(\hat{y}_j))}{|\hat{y}| - |C|} \tag{6}$$

To optimize the weights, we use the Adam optimizer [10] with Pytorch default parameters[7] for the first epoch. In the second and fifth epochs, the learning rate is reduced to $1e-4$ and $1e-5$, respectively. The batch size was set to 64. The final model was trained for eight epochs in the training dataset and one more epoch for fine-tuning in the validation dataset.

5 RESULTS

The evaluation was performed by the challenge organizers using the CodaBench platform[8] [21]. Table 1 presents the results for our model, the best academic model, and the best general model. We

[7] Pytorch Adam
[8] CodaBench: https://www.codabench.org/

Table 2: Beyond accuracy metrics for Top-5 recommendations.

Model	Diversity	Novelty	Coverage
Top-in view	0.7905	4.6258	0.02
Popular	0.8402	3.0699	0.02
Random	0.7548	8.3617	1
:D (1^{st})	0.7697	3.7017	0.632
BlackPearl (2^{nd})	0.6877	4.9304	0.32
Tom3TK (3^{th})	0.7121	4.7701	0.612
FeatureSalad (Academic 1^{st})	0.7328	12.544	0.02
Our model	0.6862	6.4124	0.332

also present the results for our model without the fine-tuning using the provided validation data in the column "*Our model (nft)*". This result indicates that our model performance can be improved by fine-tuning with more recent data. Moreover, it shows the ability of our model to be updated with new data, resulting in a performance improvement. This is important as continuous training is a well-established practice in MLOps [11] to avoid performance degradation as user preferences vary over time.

5.1 Beyond accuracy

Table 2 presents the beyond accuracy results [9] showing that our method has lower diversity than the baselines and the winning models. However, it performs better for novelty, except for Random and the *FeatureSalad* models. However, the *FeatureSalad* model has a coverage of 0.02, which is the same as Popular and Top-in view. This might indicate little variation in the recommended articles, i.e., the model recommends the same article subset to all users. This is further evidenced when we consider that the standard deviation for these metrics for the *FeatureSalad* model is 0.0, i.e., the same standard deviation reported for Popular and Top-in view, while for our model is 0.1261 for diversity and 1.8392 for novelty.

Regarding fairness, Figure 2 presents the relation among user atypicality, recommendation atypicality, cb-diversity@3, and cb-novelty@1. The diagonal presents the density distribution of each variable, while the other plots depict the combined distribution of variables. Firstly, it can be observed that user atypicality is moderately correlated with recommendation atypicality, with a Spearman correlation of 0.51 for logged users and 0.48 otherwise. This shows that atypical users tend to receive atypical recommendations, regardless of whether they are logged in. A moderated correlation is found between *cb-novelty* and *cb-diversity* (0.44 and 0.42 for logged users and non-logged users). This implies that users with low *cb-novelty* in their recommendations also have low *cb-diversity*.

When considering atypical users, they are more likely to receive recommendations with lower *cb-novelty*, as there is a correlation of −0.23 for logged users and −0.18 otherwise. A similar effect is found for *cb-diversity*, but with lower correlations −0.15 and −0.07, with the last correlation being negligible. We repeated the experiments with "beyond accuracy" instances and got similar results with the exception of the negative correlation between user atypicality and *cb-diversity* that in this case is slightly positive but negligible, at 0.03 for logged users and 0.08 otherwise. This points out that user atypicality does not affect our model diversity; however, the

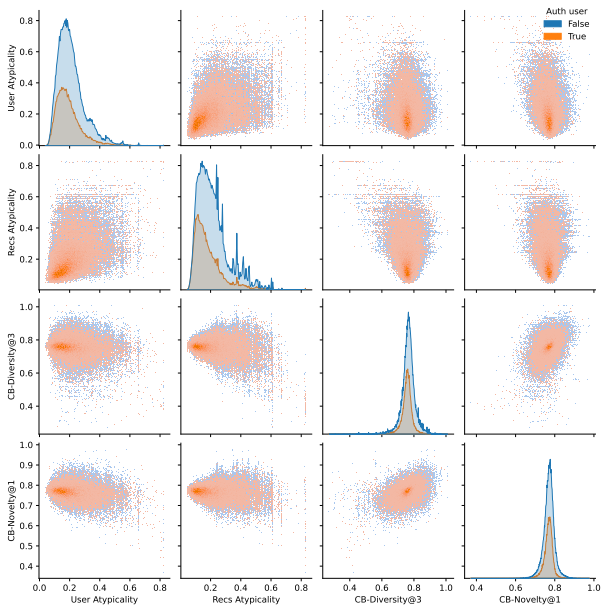

Figure 2: Atypicality, cb-diversity, and cb-novelty.

pre-filtering of the articles might by affected by user atypicality, selecting slightly less *cb-diverse* articles for such users.

Regarding user demography[9], non-logged users tended to be more atypical and receive more *cb-diverse* and *cb-novel* recommendations. Females exhibited more atypical behavior and received recommendations with lower *cb-novelty* than males. Users in metropolitan areas received lower *cb-novelty* recommendations than those in other areas. Finally, users younger than 40 tended to exhibit atypical behavior when compared with users older than 50, but the *cb-novelty* was not affected. In general, this variable's effect size was small to negligible. However, there are notable exceptions for users aged 20-29 compared to those aged 60-69 and 70-79, where the effect sizes were medium for user atypicality. All in all, our model's *cb-novelty* and *cb-diversity* were slightly affected by whether the user had an account and their gender, though the effect was small.

6 CONCLUSIONS

We presented a clicking likelihood estimation model based on the transformer architecture. The likelihood estimation is mostly based on the user's reading history but considers demographic information when available. Although it did not achieve the top AUC performance, recommendations tended to have more *novelty* than the top AUC models. Moreover, our results showed a tendency to provide personalized recommendations as atypical users receive atypical recommendations, showing a tendency to modify the recommender's behavior based on users' behavior. However, the model might yield recommendations with lower *cb-novelty* recommendations for atypical users. As a result, these users are less likely to be presented with new content to explore. Further research is needed to determine whether this is a result of atypical users consuming articles in categories with fewer articles than common categories.

[9] All effects have a p-value < 0.01

REFERENCES

[1] Guy Aridor, Duarte Goncalves, and Shan Sikdar. 2020. Deconstructing the filter bubble: User decision-making and recommender systems. In *Proceedings of the 14th ACM conference on recommender systems*. Association for Computing Machinery, New York, NY, USA, 82–91.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[3] Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. 2024. Fairness in recommender systems: research landscape and future directions. *User Modeling and User-Adapted Interaction* 34, 1 (2024), 59–108.

[4] Sarah Frost, Manu Mathew Thomas, and Angus G Forbes. 2019. Art I don't like: An anti-recommender system for visual art. In *Proceedings of Museums and the Web*.

[5] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. Association for Computing Machinery, New York, NY, USA, 69–78.

[6] Alireza Gharahighehi and Celine Vens. 2023. Diversification in session-based news recommender systems. *Personal and Ubiquitous Computing* 27, 1 (2023), 5–15.

[7] Mario Haim, Andreas Graefe, and Hans-Bernd Brosius. 2018. Burst of the filter bubble? Effects of personalization on the diversity of Google News. *Digital journalism* 6, 3 (2018), 330–343.

[8] Natali Helberger, Kari Karppinen, and Lucia D'acunto. 2018. Exposure diversity as a design principle for recommender systems. *Information, Communication & Society* 21, 2 (2018), 191–207.

[9] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1, Article 2 (dec 2016), 42 pages. https://doi.org/10.1145/2926720

[10] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] https://arxiv.org/abs/1412.6980

[11] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* 11 (2023), 31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138

[12] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*. Association for Computing Machinery, New York, NY, USA, 624–632.

[13] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) *(WWW '14)*. ACM, New York, NY, USA, 677–686. https://doi.org/10.1145/2566486.2568012

[14] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

[15] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[16] Dimitris Sacharidis. 2019. Diversity and Novelty in Social-Based Collaborative Filtering. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization* (Larnaca, Cyprus) *(UMAP '19)*. Association for Computing Machinery, New York, NY, USA, 139–143. https://doi.org/10.1145/3320435.3320479

[17] Javier Sanz-Cruzado and Pablo Castells. 2018. Enhancing structural diversity in social networks by recommending weak ties. In *Proceedings of the 12th ACM conference on recommender systems*. Association for Computing Machinery, New York, NY, USA, 233–241.

[18] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1441–1450. https://doi.org/10.1145/3357384.3357895

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[20] Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. 2021. Recommenders with a mission: assessing diversity in news recommendations. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. Association for Computing Machinery, New York, NY, USA, 173–183.

[21] Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns* 3, 7 (2022), 100543. https://doi.org/10.1016/j.patter.2022.100543

[22] Zhiyu Yao, Xinyang Chen, Sinan Wang, Qinyan Dai, Yumeng Li, Tanchao Zhu, and Mingsheng Long. 2024. Recommender Transformers with Behavior Pathways. In *Proceedings of the ACM on Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 3643–3654. https://doi.org/10.1145/3589334.3645528

[23] Mert Yuksekgonul, Linjun Zhang, James Zou, and Carlos Guestrin. 2023. Beyond Confidence: Reliable Models Should Also Quantify Atypicality. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*. https://openreview.net/forum?id=nPOKJCCvlLF

[24] Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. 2023. Attention Calibration for Transformer-based Sequential Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3595–3605. https://doi.org/10.1145/3583780.3614785