RADio-: a simplified codebase for evaluating normative diversity in recommender systems

Sanne Vrijenhoek[1,2]

[1] *AI, Media and Democracy Lab, Amsterdam, the Netherlands*

[2] *Institute for Information Law, University of Amsterdam, the Netherlands*

Abstract

Diversity is one of the core beyond-accuracy objectives considered in the development of news recommender systems. However, there is a clear gap between its technical conceptualization, typically as an intra-list distance, and a more normative interpretation, which touches upon the role the recommender system plays in society. Vrijenhoek et al. [1] proposed to instead use rank-aware divergence metrics to express normative diversity in news recommendations. This work describes a repository that allows for easy implementation of these metrics, by making the different diversity aspects and tactics configurable. It also contains an example implementation and analysis of the results. With its modular setup, the repository thus allows for conceptualizations of diversity that can be tailored to the news domain they need to be applied in.

Keywords

LaTeX class, paper template, paper formatting, CEUR-WS

1. Introduction

In its technical conceptualization, 'diversity' prevents a recommender system from recommending the same type of content over and over again [2], and is one of the primary values considered in research on news recommender systems [3]. Diversity is usually expressed as an 'intra-list distance', measuring whether the items within the recommendation are sufficiently different from each other [4]. However, this definition based on intra-list distance does not fully reflect the requirements of a *normative* interpretation of diversity, which relates to the role the recommender system plays in society [5, 6, 7].

Diversity has characteristics of an essentially contested concept [8], and its interpretations can vary greatly among different people and implementations [9]. We may consider different *aspects* when talking about diversity, such as political viewpoints [10], different ethnicities [11] or diversity of category and topic [12]. We may also talk about diversity in different *contexts*: for example, the recommendation should reflect society [13], it should counter existing biases [14], or expose the reader to new things [15].

Most, if not all, of these conceptualizations are relevant to the domain of news recommendations. News recommenders may play a gatekeeping role in the type of news that is exposed to the public, and thus need to be capable of incorporating editorial values [16, 17]. The different aspects could logically be incorporated in the intra-list distance formalization mentioned above with a sufficiently sophisticated way to calculate distance between articles. However, the different *contexts* cannot be captured by a metric that only considers the items within a recommendation. To solve this, Vrijenhoek et al. [1] proposed an alternative diversity metric, conceptualized as a rank-aware divergence metric. This was called the RADio framework, where RADio is short for **R**ank-**A**ware **D**ivergence (plus -io).

With these divergence metrics, the presence of a certain diversity aspect in the recommendation is compared to the presence of that aspect in an external context distribution. When these distributions are similar, the divergence is low; when they are very different, divergence is high. There is no clear-cut 'optimal' divergence score. In some cases one could strive for a recommendation similar to the context distribution (for example, be reflective of political voices in government), in others for a higher divergence score (for example, expose a reader to new perspectives). To show how this could work in

INRA'24: News Recommendation and Analytics Workshop, October 18, 2024, Bari, IT

s.vrijenhoek@uva.nl (S. Vrijenhoek)
0000-0002-1031-4746 (S. Vrijenhoek)

practice RADio implemented the diversity metrics (DART) outlined in Vrijenhoek et al. [18]: Calibration, Fragmentation, Activation, Representation and Alternative Voices, which are inspired by democratic theory. The metrics were prototyped with news recommenders trained on the Microsoft News Dataset (MIND) [19].

In order to do justice to the normative underpinnings of the DART metrics, the RADio metrics needed metadata that was not included in MIND. This metadata would include things like which political viewpoints are expressed in an article, is the article written in a neutral or subjective tone, or does the article mention people from a minority background. This type of information is notoriously hard to extract from just a text, and often RADio needed to rely on *proxies* that were known not to be exactly right, but were necessary to prototype how the framework could theoretically function.

Despite the fact that they were simplifications, the data preprocessing and augmentation steps to identify these proxies were already quite elaborate. For example, political opinions would be approximated by the mention of political actors in the text. These actors would be identified by 1) scraping article body; 2) running Named Entity Recognition on the fulltexts; 3) attempting to match entities of type Person to their entry on Wikidata; 4) checking whether this person was a politician, and for which party. Without a Golden Standard it was not possible to evaluate the performance of this approach, but even just looking at the procedure makes it quite clear that there are a lot of ways in which this process can fail. Alterations in the spelling of a name (Barack Obama vs. President Obama) could leave a political actor unidentified, and new elections or party compositions would render past results invalid. This approach, or even one based on regular expressions and/or string matching, would probably work well enough in a contained experiment over a limited amount of time where the relevant actors are already known, such as in Michiels et al. [20] and Einarsson et al. [13]. An implementation that monitors an algorithm in real-time would probably benefit from a more sophisticated approach to viewpoint diversity, such as in Draws et al. [21]. Lastly, RADio's implementation of the DART metrics also distracts from the findings of Vrijenhoek et al. [9], which claims that diversity can be conceptualized in many ways, depending on the domain's requirements.

This work describes a repository that allows for easy implementation of the divergence-based metrics, by making the different diversity aspects and tactics configurable. The code can be found on Github.[1] This paper works under the assumption that whoever implements the framework has a data preprocessing or annotation pipeline that contains the required metadata for the metrics to work. While it still keeps the DART metrics in the repository to give examples of metric configuration, the framework can also accommodate domains beyond news recommendation. Keep in mind that the repository does **not** provide plug-and-play metrics, and that conceptualizing diversity within a news recommender system is still very much a matter of discussion with stakeholders from outside technical teams [22, 23].

2. The repository

The repository consists of three primary components: a Jupyter notebook which showcases how metrics could be configured, a class for building the rank-aware distributions, and a class for calculating the divergence scores.

2.1. Building distributions

In this part of the framework, we aim to build the distributions for the recommendation and context respectively. In order to do this, we pass the framework the list of relevant articles (either in the recommendation or in the context), and tell it which feature to look for. When building the distribution, the framework can optionally account for the *rank* of an article in the recommendation. It will then count articles that appear higher up in the list more strongly than those that appear lower by weighing it

[1] https://github.com/svrijenhoek/radio-/

with the *harmonic number*[2] of the length of the list. Making a distribution rank-aware only makes sense when there is some sort of meaning in the ordering of the articles; for example, in a recommendation ranked by predicted relevance, or in a reading history when the most recently read articles are listed first. It does not make sense in cases where such a meaning cannot be found; for example, when considering all articles that have been published over the last few days. The framework can accommodate both categorical and numerical data. Categorical data can have both single and multiple values per article. In case of numerical data, the values will need to be discretized into bins. The number of bins to be used can be set, but defaults to 10. With this approach, we lose a lot of important information. For example, we will not know that certain bins may be closer to each other than others. Future work may look into alternative ways of calculating divergence for numerical data.

2.2. Calculating divergence

Within RADio, diversity is conceptualized as *a rank-aware divergence score between a recommendation and a context*:

$$D_f^*(P,Q) = \sum_x Q^*(x) f(\frac{P^*(x)}{Q^*(x)}) \tag{1}$$

where x refers to the relevant feature to consider; P to the recommendation, and Q to the context. As explained in the previous section, both the recommendation P and context Q can be set up to be *rank-aware*. For more details regarding the justification of setting up diversity as a rank-aware divergence score, see Vrijenhoek et al. [1]. Within this framework, we can calculate the divergence using both Kullback-Leibler (KL) divergence and the Jensen-Shannon (JS) divergence [24]. While KL Divergence is commonly known, JS Divergence has the added benefit of being 1) symmetric and 2) bound between 0 and 1, and is thus the default option within the framework.

2.3. Configuring the metrics

While the repository contains instructions for configuring all the original RADio metrics, for this paper we will discuss the configuration and output of the Calibration metric in more detail as an example. While Calibration is from a normative perspective not the most interesting metric, it relies on data that is supplied in MIND itself, and therefore does not rely on complicated data augmentation to show meaningful results.

The goal of Calibration is to measure to which extent a recommendation is tailored to a user's preferences. Thus, we want this score to show low divergence, meaning that there is actually a large overlap between the recommendation and what a user wants to see. In this setup, we approximate a users' preferences by looking at the categories of articles they have consumed in the past: their reading history. Note that this is just an example implementation, and that there are likely many better ways to express a users' interests than through categories in past reading behavior.

In summary, we configure the metric in the following way:

Table 1
Configuration of the Calibration metric

Metric component	**Configuration**
Feature (x)	article category
Context (Q)	user history
Feature type	categorical; here single but could be multi
Rank-aware	both recommendation and context (P and Q)
Desired value	low divergence

We expect all recommendations to be represented in a DataFrame, with columns for 1) the impression ID; 2) the time of the impression; 3) the ID of the user this impression was shown to; 4) the reading

[2] https://en.wikipedia.org/wiki/Harmonic_number

history of that user; and 5) one or more generated recommendations, corresponding to different recommendation algorithms. We assume that an apply-method will be called to calculate the diversity metrics for each line, and thus for each of the different algorithms. We first configure a Metric:

```
Calibration = DiversityMetric(
    feature_type='cat',
    rank_aware_recommendation=True,
    rank_aware_context=True,
    divergence='JSD',
    context = 'dynamic'
    )
```

Here, 'feature_type', 'rank_aware_recommendation', 'rank_aware _context' and 'divergence' correspond to the information summarized in Table 1. The *context* parameter is there for efficiency. If the context is *dynamic*, it will need to be calculated for every line. This is the case here, as we are looking at the users' reading history, which is of course different for every user. The context can also be *static*, or the same for all users. This is the case when for example looking at all articles published, or when considering an external distribution. Next, we write a *calculate_calibration* function to pass the right recommendation and context to the framework:

```
def calculate_calibration(recommendations, history):
    scores = []
    context_features = get_features(history, 'category')
    for recommendation in recommendations:
        recommendation_features = get_features(recommendation, 'category')
        if context_features and recommendation_features:
            calibration =  Calibration.compute(context_features,
             ↪  recommendation_features)
            scores.append(calibration)
        else:
            scores.append(None)
    return scores
```

We expect 'recommendations' to be a list where each entry in the list corresponds to a different algorithm. Each entry again consists of a list of article IDs. We also expect that these are ordered by which article is going to be recommended first according to that algorithm. Next, we tell the framework to retrieve the 'category' feature for each article in both the recommendations and the reading history. The resulting lists of features are given to the framework to, under the hood, build the corresponding distributions and calculate the divergence. The resulting 'scores' is a list of scores, each entry corresponding to one of the recommendation algorithms.

3. Output

We run the configured metric on the news articles and recommendations of the *'MINDsmall_dev'* dataset, which can be obtained from the Microsoft website[3]. We compare the recommendations generated by the LSTUR [25] and NRMS [26] algorithms, trained using the code supplied by Microsoft[4], to those from two simple baseline algorithms: a random selection, and a selection based on article popularity. For

[3] https://msnews.github.io/

[4] https://github.com/recommenders-team/recommenders/tree/main

Table 2
Statistics of the Calibration scores for each recommendation algorithm

	lstur	nrms	pop	random
mean	0.575	0.572	0.665	0.662
min	0.000	0.000	0.000	0.000
25%	0.461	0.458	0.581	0.558
50%	0.564	0.559	0.666	0.662
75%	0.681	0.677	0.752	0.768
max	0.994	0.994	0.994	0.994
std	0.159	0.160	0.132	0.154

the most popular baseline, the popularity of an item is derived from the clicks recorded in the dataset. However, there are many articles with zero recorded clicks, and in case of a tie in the number of clicks the recommender makes a random selection.

It is quite hard to pinpoint what exactly a ‘good’ divergence score would be. However, when we compare the algorithm we are interested in to a baseline algorithm, we can draw some conclusions on how that algorithm impacts the behavior of the metric. In this example, the first difference in metric outcomes can already be observed from calculating basic statistics on the outcomes, visualized in Table 2. At each point, the divergence in the neural recommenders is lower than those for the baseline recommenders. As expected, the neural recommenders are more tailored to the users’ preferences than the baselines. Note that this does not mean that generally neural recommenders are more diverse than baseline ones; it just means that in *this* conception of diversity, and in this setting, the neural recommenders show more of the desired behavior than the baseline does.

Figures 1 and 2, which can also be found in the repository, provide more details into the behavior of the metric. In Figure 1 we see that the neural recommenders show similar patterns, and that the baseline recommenders behave similarly between them too. It also shows the effect of the time of day; there may have been meaningful events that influence the type of articles a recommender system can choose from, and thus make the algorithm choose articles that diverge from the users’ personal preferences. Note that this is not necessarily bad, if the primary goal of the recommender is to inform readers about important events happening in the world. In Figure 2, the neural recommenders have distinctly lower divergence, which means that the recommendations they generate are closer to the users’ reading history. Given that there are clear differences between the baseline recommenders in this image but not in Figure 1, some meaningful consequences happen when aggregating scores per user; the most popular recommender may generate more Calibrated recommendations for some users than for others.

4. Discussion

Section 3 explains how to *technically* implement the RADio- framework to measure normative diversity in recommendations. The example metrics are tailored towards news recommendation, but the framework can be adapted to suit a wide range of applications. Yet, this does not yet answer the question of *how* one should go about conceptualizing diversity for their application. This can be exceptionally challenging for technical teams that, while they are the ones that need to implement the metric, often do not have all the domain knowledge necessary for making such decisions. As such, it is important that all relevant stakeholders are brought to the table. In the case of news recommendation, these would include editorial, but also strategic and business roles [22]. Readers themselves also bring a different perspective on what they value in their news, and why they would choose to read certain items but others not [27, 28, 29, 30]. Lastly, one should not underestimate the effects of interface design on users’ reading behavior. Even a perfectly built and diverse algorithm may not accomplish what it is intended to do due to position bias or simply differences between users [31].

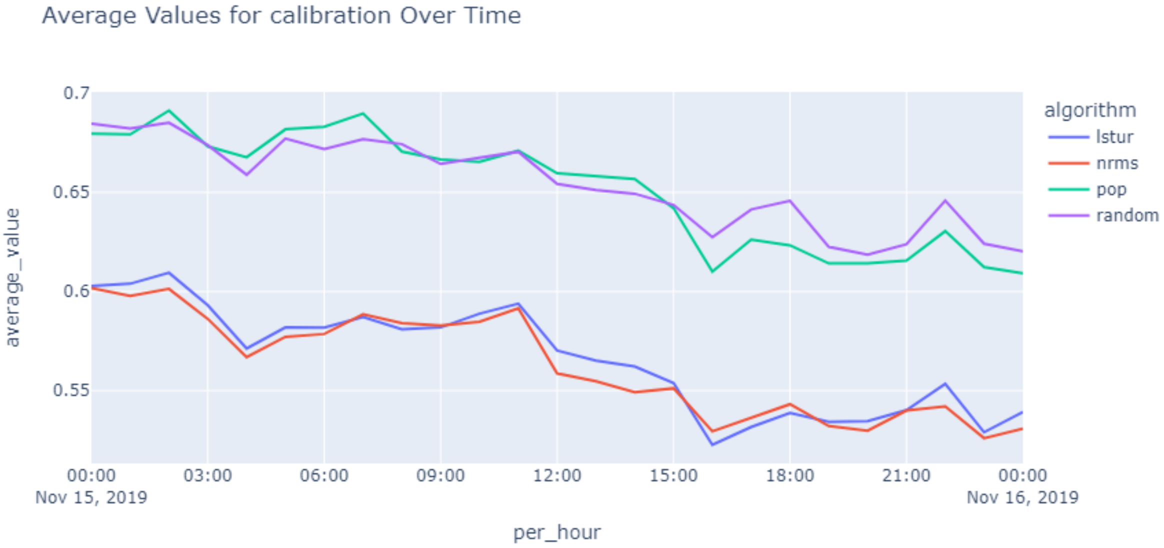

Figure 1: Lineplot of the average Calibration scores over time

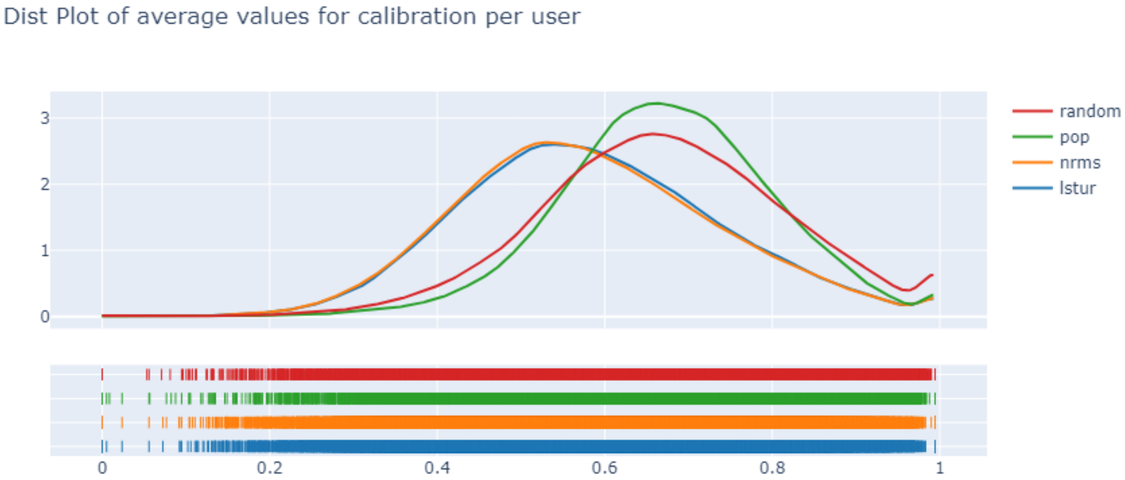

Figure 2: Distributional plot of the average Calibration score per user

Vrijenhoek et al. [9] interviewed professionals in the media sector, and noted all the different ways they spoke about diversity. The taxonomy that resulted from this, which is split into *goals*, *aspects* and *contexts* of diversity, could serve as a starting point for other implementations; at the very least, it should facilitate discussion and ease the identification of domain-specific needs and requirements. Furthermore, one could take inspiration from literature beyond the technological domain. For example, those working on news recommendation could look into how social scientists conceptualize diversity, and draw inspiration from democratic theory and the role news plays in society [6, 18]. However, while democratic theory is directly relevant to news, it should not be blindly applied to other domains. Rather, we would encourage those from other domains to invest time choosing or building their own normative framework [7].

Without a doubt, conceptualizing and implementing diversity in any kind of recommender system is a complicated process, and it is unlikely that a perfect (or even a good) solution will be attained in a single iteration. One could argue that aiming for one would only prevent any progress from happening. Rather, perhaps we should aim for *imperfect* solutions; ones that we fully understand, and where we can exactly pinpoint what the metric does and does not do. As such, we would also urge readers not to resort to opaque solutions such as off-the-shelf Large Language Models, which may be easy to implement but are not under the control and full understanding of your organization. Solutions that we know are simplified, perhaps even 'stupid', can be discussed and criticized, and thus be improved upon. It is our hope that the RADio- codebase will make at least the technical part of the process more straightforward.

Acknowledgments

This work builds upon the code of the original RADio framework, which was a collaboration between the author of this work, Gabriel Bénédict and Mateo Gutierrez Granada. I would like to thank Savvina Daniil for testing and reviewing the code, Johannes Kruse for making the repository's very first pull request, and Lucien Heitz and Alain Starke for proofreading.

References

[1] S. Vrijenhoek, G. Bénédict, M. Gutierrez Granada, D. Odijk, M. De Rijke, RADio–Rank-Aware Divergence Metrics to Measure Normative Diversity in News Recommendations, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 208–219.

[2] M. Kunaver, T. Požrl, Diversity in recommender systems–a survey, Knowledge-based systems 123 (2017) 154–162.

[3] C. Bauer, C. Bagchi, O. A. Hundogan, K. van Es, Where are the values? a systematic literature review on news recommender systems, ACM Transactions on Recommender Systems (2024).

[4] S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 109–116.

[5] N. Helberger, K. Karppinen, L. D'acunto, Exposure diversity as a design principle for recommender systems, Information, communication & society 21 (2018) 191–207.

[6] N. Helberger, On the democratic role of news recommenders, in: Algorithms, Automation, and News, Routledge, 2021, pp. 14–33.

[7] S. Vrijenhoek, L. Michiels, J. Kruse, A. Starke, N. Tintarev, J. Viader Guerrero, Normalize: The first workshop on normative design and evaluation of recommender systems, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1252–1254.

[8] W. B. Gallie, Essentially contested concepts, Aristotelian Society, 1956.

[9] S. Vrijenhoek, S. Daniil, J. Sandel, L. Hollink, Diversity of what? on the different conceptualizations of diversity in recommender systems, in: The 2024 ACM Conference on Fairness, Accountability, and Transparency, 2024, pp. 573–584.

[10] M. Haim, A. Graefe, H.-B. Brosius, Burst of the filter bubble? effects of personalization on the diversity of google news, Digital journalism 6 (2018) 330–343.

[11] M. Mitchell, D. Baker, N. Moorosi, E. Denton, B. Hutchinson, A. Hanna, T. Gebru, J. Morgenstern, Diversity and inclusion metrics in subset selection, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 117–123.

[12] J. Möller, D. Trilling, N. Helberger, B. van Es, Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity, in: Digital media, political polarization and challenges to democracy, Routledge, 2020, pp. 45–63.

[13] Á. M. Einarsson, R. Helles, S. Lomborg, Algorithmic agenda-setting: the subtle effects of news

recommender systems on political agendas in the danish 2022 general election, Information, Communication & Society (2024) 1–21.

[14] B. Huebner, T. E. Kolb, J. Neidhardt, Evaluating group fairness in news recommendations: A comparative study of algorithms and metrics, in: Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, UMAP Adjunct '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 337–346. URL: https://doi.org/10.1145/3631700.3664897. doi:10.1145/3631700.3664897.

[15] L. Heitz, J. A. Lischka, R. Abdullah, L. Laugwitz, H. Meyer, A. Bernstein, Deliberative diversity for news recommendations: Operationalization and experimental user study, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 813–819. URL: https://doi.org/10.1145/3604915.3608834. doi:10.1145/3604915.3608834.

[16] L. A. Møller, Recommended for you: how newspapers normalise algorithmic news recommendation to fit their gatekeeping role, Journalism Studies 23 (2022) 800–817.

[17] S. Blassnig, E. Strikovic, E. Mitova, A. Urman, A. Hannák, C. de Vreese, F. Esser, A balancing act: How media professionals perceive the implementation of news recommender systems, Digital Journalism (2023) 1–23.

[18] S. Vrijenhoek, M. Kaya, N. Metoui, J. Möller, D. Odijk, N. Helberger, Recommenders with a mission: assessing diversity in news recommendations, in: Proceedings of the 2021 conference on human information interaction and retrieval, 2021, pp. 173–183.

[19] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu, et al., Mind: A large-scale dataset for news recommendation, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 3597–3606.

[20] L. Michiels, J. Vannieuwenhuyze, J. Leysen, R. Verachtert, A. Smets, B. Goethals, How should we measure filter bubbles? a regression model and evidence for online news, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 640–651.

[21] T. Draws, N. Roy, O. Inel, A. Rieger, R. Hada, M. O. Yalcin, B. Timmermans, N. Tintarev, Viewpoint diversity in search results, in: European Conference on Information Retrieval, Springer, 2023, pp. 279–297.

[22] A. Smets, J. Hendrickx, P. Ballon, We're in this together: a multi-stakeholder approach for news recommenders, Digital Journalism 10 (2022) 1813–1831.

[23] N. Helberger, M. van Drunen, J. Moeller, S. Vrijenhoek, S. Eskens, Towards a normative perspective on journalistic ai: Embracing the messy reality of normative ideals, 2022.

[24] M. L. Menéndez, J. Pardo, L. Pardo, M. Pardo, The jensen-shannon divergence, Journal of the Franklin Institute 334 (1997) 307–318.

[25] M. An, F. Wu, C. Wu, K. Zhang, Z. Liu, X. Xie, Neural news recommendation with long-and short-term user representations, in: Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, pp. 336–345.

[26] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, X. Xie, Neural news recommendation with multi-head self-attention, in: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, pp. 6389–6394.

[27] A. Starke, A. S. Bremnes, E. Knudsen, D. Trilling, C. Trattner, Perception versus reality: Evaluating user awareness of political selective exposure in news recommender systems, in: Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, 2024, pp. 286–291.

[28] J. Moeller, F. Löecherbach, J. Möller, N. Helberger, Out of control?: Using interactive testing to understand user agency in news recommendation systems, in: News Quality in the Digital Age, Routledge, 2023, pp. 117–133.

[29] L. Van den Bogaert, D. Geerts, J. Harambam, Putting a human face on the algorithm: co-designing recommender personae to democratize news recommender systems, Digital Journalism (2022) 1–21.

[30] F. Loecherbach, K. Welbers, J. Moeller, D. Trilling, W. Van Atteveldt, Is this a click towards diversity? explaining when and why news users make diverse choices, in: Proceedings of the 13th ACM Web Science Conference 2021, 2021, pp. 282–290.

[31] N. Mattis, T. Groot Kormelink, P. K. Masur, J. Moeller, W. van Atteveldt, Nudging news readers: A mixed-methods approach to understanding when and how interface nudges affect news selection, Digital Journalism (2024) 1–21.