MSc Artificial Intelligence
Master Thesis

PD-Rec: Should We Personalize the Level of Diversity per User?

by
Luuk Kaandorp
11992190

July 21, 2022

48EC
November 1st 2021 - June 30th 2022

External Supervisors:
D. Odijk
M. Gutierrez Granada
Supervisor:
S. Bhargav

Second reader:
E. Kanoulas

Universiteit van Amsterdam

Contents

Abstract

With the rise of the internet and accompanying news websites, it has never been easier to be up-to-date on the latest news. Newman et al. (2021) found that a large portion of the population in developed countries indicated that their primary news source is digital. However, the constant stream of published news articles that news websites provide can lead to information overload for users.

To help users navigate the abundance of news articles, recommender systems have been introduced to online news platforms. These recommender systems suggest relevant news articles to users based on their past reading behaviour.

However, these recommenders are optimized for recommendation accuracy, which means it is optimized to correctly predict whether a user clicks on a candidate news article. Unfortunately, as Helberger (2019) point out, even though this optimization objective leads to high personalization, which is desirable in general, it can also lead to adverse effects like echo chambers (Sunstein, 2001) and filter bubbles (Pariser, 2011).

One way to prevent echo chambers and filter bubbles is to diversify the recommendations. Research, such as by Raza and Ding (2021a) and Raza (2021), uncovered that accuracy/relevancy and diversity trade between one another. Therefore, providing more diverse recommendations inherently comes at the cost of less accurate and less relevant recommendations.

Therefore, we introduce a re-ranking algorithm called *PD-Rec*, that personalizes the degree of diversity for each user individually. The main idea behind *PD-Rec* is that users with diverse preferences and news consumption should not be pushed to even more diverse consumption. In contrast, users with very uniform news consumption should be nudged to read more diverse articles. This novel approach can trade off diversity and relevancy on a per-user basis.

We also test this approach on two offline, real-world datasets. First, experiments using NRMS (Wu et al., 2019c), NAML (Wu et al., 2019a), and PLM-NR (Wu et al., 2021a) have shown that state-of-the-art news recommenders are not more diverse than older models, such as LSTUR (An et al., 2019), and are also not more diverse among each other.

Second, experiments using the same models with and without our *PD-Rec* re-ranking approach have also proven that *PD-Rec* can help increase diversity in terms of ILD and RR-ILD. However, it can also harm coverage, as we found to be the case on the MIND dataset. We have also shown that the top-10 recommendations change significantly after applying the personalized diverse re-ranking. Furthermore, we also find that the personalized diverse re-ranker works as intended and has a more significant impact on the recommendations shown to users with undiverse reading habits than on the recommendations shown to users with diverse reading habits. Thus, personalized levels of diversity can help increase diversity for users with undiverse reading habits while preserving accurate recommendations for users with diverse reading habits.

Last, experiments also show that by using personalized levels of diversity over fixed levels, we can better achieve the goal of providing diverse recommendations to users with undiverse reading habits while preserving accuracy for users who already consume diverse news.

List of Acronyms

NLP Natural Language Processing

PLM Pre-trained Large Language Model

MIND Microsoft News Dataset

BERT Bidirectional Encoder Representations from Transformers

ILS Intra-List Similarity

ILD Intra-List Distance

COV Coverage

nDCG normalized Discounted Cumulative Gain

RR-ILD Rank and Relevance sensitive Intra-List Distance

RTL-NR RTL - News Recommendation

AUC Area Under the Curve

MRR Mean Reciprocal Rank

RBO Rank-Biased Overlap

PD-Rec Personalized Diverse Recommendation

Chapter 1

Introduction

With the rise of the internet and accompanying news websites, it has never been easier to be up-to-date on the latest news. Newman et al. (2021) found that the recent COVID-19 pandemic accelerated the shift toward digital media. Furthermore, a large portion of the population in developed countries indicated that their primary news source is digital. Where traditional media such as newspapers and news programs on TV have an inherent delay due to redacting, printing, TV scheduling, and other factors, the digitalization of news pushes the boundaries of freshness and immediacy. However, the constant stream of published news articles can lead to information overload for users.

To help users navigate the abundance of news articles, recommender systems have been introduced to online news platforms. Recommender systems are algorithms or Machine Learning models aimed at suggesting relevant items (e.g. movies, products, or news articles) to users based on past user behaviour. Where recommender systems have already been employed in other applications to overcome choice overload, such as e-commerce (Linden et al., 2003), the same has been applied to online news (Wu et al., 2020b; An et al., 2019; Wu et al., 2019c; Zheng et al., 2018). However, news recommendation is fundamentally different from product recommendation.

A core difference between news and product recommendation is how items (either products or news articles) are represented. E-commerce recommender systems benefit from rich meta-data about items, such as smartphone screen size and operating system, whereas news articles lack such features. This ultimately means that news recommendation relies primarily on text features, while product recommendation can utilize far richer meta-data. Another difference is timeliness, whereas it generally does not matter when an item was added for product recommendation. For news recommendation, one inherently wants to recommend 'news'; thus, the usefulness of a particular news article decays over time (Raza and Ding, 2021b). Furthermore, user behaviour in news recommendation tends to be highly dynamic, with short-term and long-term reading preferences that can change over time (Li et al., 2014). This poses difficulties in capturing the preferences and intents of users (An et al., 2019).

Since we need to deduce the meaning of an article from the textual content of an article, text modelling from the Natural Language Processing (NLP) domain has been introduced to news recommender systems. In text modelling, NLP models are employed to capture the content of text in a numerical representation. More recently, Pre-trained Large Language Models (PLMs) have achieved great success in multiple NLP tasks due to their strength in text modelling, as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) has proven. This strength can be primarily attributed to three aspects: the large scale of the models (ranging from millions to billions of parameters), the large amounts of data on which these models are trained, and the architecture which utilizes attention mechanisms at its core. In their work, Wu et al. (2021a) utilized a PLM in the news encoder of their

PLM-NR news recommendation system. Their proposed PLM-NR model achieves new state-of-the-art performance on Microsoft News Dataset (MIND), a news recommendation dataset and benchmark. According to the MIND leaderboard, this architecture with Fastformer (Wu et al., 2021b) as text encoder performs the best on the benchmark [1]. This highlights the usefulness of PLMs in news encoders in news recommendation.

However, like many other existing models and algorithms (Wu et al., 2019c,a; de Souza Pereira Moreira, 2018; Wang et al., 2018), the PLM-NR architecture optimizes recommendation accuracy, which means it is optimized to correctly predict whether a user clicks on a candidate news article. Unfortunately, as Helberger (2019) point out, even though this optimization objective leads to high personalization, which is desirable in general, it can also lead to adverse effects like echo chambers (Sunstein, 2001) and filter bubbles (Pariser, 2011).

Echo chambers occur whenever a user only gets presented with information that echoes what he is already thinking and conforms to his existing preferences. Echo chambers can, in turn, lead to filter bubbles where like-minded newsreaders are isolated from different viewpoints and perspectives, with their existing viewpoint being reinforced by information that supports it. Due to the crucial role of media in a democratic society, one could even argue that news sites have a democratic duty to prevent echo chambers and filter bubbles. News recommender systems, therefore, can be pivotal in deciding what kind of news the public does and does not see (Vrijenhoek et al., 2021).

One way to prevent echo chambers and filter bubbles is to diversify the recommendations. There exist multiple views from different research fields on what diversity entails. As Vrijenhoek et al. (2021) note, computer scientists view diversity as the intra-list distance of the recommended items. Intra-list distance measures the average pairwise distance among the recommendations and is often defined by the cosine distance between the representations of the items (Carbonell and Goldstein, 1998). Social studies, however, view diversity as larger societal concepts concerned with democracy, freedom of expression, cultural inclusion, mutual respect, and tolerance (Helberger, 2019). Vrijenhoek et al. (2021) have bridged the gap between different research fields by providing a set of interdisciplinary metrics. However, the question remains about how models can be optimized toward both these metrics and accuracy.

Earlier research on diversity in news recommendation aimed to provide more diverse recommendations. However, research, such as by Raza and Ding (2021a) and Raza (2021), uncovered that accuracy/relevancy and diversity trade between one another. Therefore, providing more diverse recommendations inherently comes at the cost of less accurate and less relevant recommendations. Earlier works on diversity, like those by Wu et al. (2020a), Raza (2021), and Qi et al. (2021), aimed to introduce more diversity to news recommendations without significantly decreasing accuracy through architectural changes in the recommender system.

However, these gains in diversity and losses in accuracy are the same across all users. Since the works by Raza and Ding (2021a) and Raza (2021) have proven that diversity and accuracy are a trade-off in terms of news recommendation, we hypothesize that we might want to present different levels of diversity to different users. Therefore, we introduce a novel re-ranking algorithm, Personalized Diverse Recommendation (PD-Rec), that personalizes the degree of diversity for each user.

Our first contribution is the comparison of state-of-the-art news recommendation models on their diversity. To the best of our knowledge, no work exists that compares recent neural news recommenders, such as NRMS (Wu et al., 2019c), NAML (Wu et al., 2019a), or PLM-NR (Wu et al., 2021a), on their diversity. Furthermore, baselines in earlier diversity research have used models that have become outdated and are outperformed by current state-of-the-art attention models.

Our second contribution is the introduction of the novel PD-Rec re-ranking algorithm that

[1] As last checked on July 20th 2022.

personalizes the degree of diversity. This novel approach can trade off diversity and relevancy on a per-user basis. We test this approach on two offline, real-world datasets.

Our third contribution is that we open-source this research. We find that many recent papers on news recommendation and diversity in news recommendation do not make their code-base publicly available. This is generally not ideal in research as it makes the results in a paper much harder to reproduce and verify.

More specifically, this paper answers the following research questions:

RQ1 How diverse are state-of-the-art attention-based news recommenders?

RQ2 How can personalized levels of diversity contribute to better diversity in news recommendation?

RQ2.1 Do personalized levels of diversity provide benefits over fixed levels?

Experiments using NRMS (Wu et al., 2019c), NAML (Wu et al., 2019a), and PLM-NR (Wu et al., 2021a) have shown that state-of-the-art news recommenders are not more diverse than older models, such as LSTUR (An et al., 2019), and are also not more diverse among each other.

Experiments using the same models with and without our PD-Rec re-ranking approach have also proven that PD-Rec can help increase diversity in terms of Intra-List Distance (ILD) and rr-ild. However, it can also harm Coverage (COV), as we found to be the case on the MIND dataset. We have also shown that the top-10 recommendations change significantly after applying the personalized diverse re-ranking. Furthermore, we also find that the personalized diverse re-ranker works as intended and has a more significant impact on the recommendations shown to users with undiverse reading habits than on the recommendations shown to users with diverse reading habits. Thus, personalized levels of diversity can help increase diversity for users with undiverse reading habits while preserving accurate recommendations for users with diverse reading habits.

Lastly, experiments also show that by using personalized levels of diversity over fixed levels, we can better achieve the goal of providing diverse recommendations to users with undiverse reading habits while preserving accuracy for users who already consume diverse news.

To summarize, we can conclude that by using personalized levels of diversity, we are better able to provide more diverse recommendations to users with undiverse reading habits to incentivize them to consume more diverse news. Meanwhile, we still preserve accurate recommendations for users with already diverse reading habits.

Chapter 2

Related work

In the following sections, we will discuss different fields of related work. The first section discusses advancements in personalized news recommendation. The second section elicits diversity metrics that are used in both the general recommendation field and news recommendation. Lastly, the third section covers past work in diversifying news recommendations.

2.1 Personalized News Recommendation

Traditional algorithms used in recommender systems can also be applied to news recommendations. These algorithms are often classified into three categories: Collaborative Filtering (CF), Content-Based Filtering (CBF), and hybrid approaches that combine CF and CBF (Adomavicius and Tuzhilin, 2005). Whereas Content-Based Filtering algorithms provide recommendations by comparing a user profile with candidate item representations, Collaborative Filtering algorithms are content-free and exploit user behaviours and interactions on items.

Following the same trend as the general recommender domain, these algorithms were replaced by factorization models, like the Matrix Factorization (MF) algorithm that achieved popularity after the Netflix competition (Koren et al., 2009). MF works by decomposing a user-item interaction matrix into a product of matrices of lower dimensionality such that it reveals latent features within the interactions between users and items. As Raza and Ding (2021b) note, other related algorithms emerged that solved problems of the original MF algorithm. For instance, Tensor Factorization (Frolov and Oseledets, 2018) allows for the inclusion of additional information about users and items, such as time, location, and social interactions, which has proven to be beneficial in news recommendation (Wang et al., 2015). However, these methods have fallen out of favour with the emergence of deep learning.

With the rise of deep learning, the news recommender domain also changed. Since the work by Karatzoglou et al. (2016), many deep learning-based approaches have been proposed for news recommendation. After Multi-Layer Perceptrons (MLPs) (He et al., 2017; Song et al., 2016), Autoencoders (AEs) (Wu et al., 2016; Okura et al., 2017; Cao et al., 2017), Convolutional Neural Networks (CNNs) (Wang et al., 2018; Zhu et al., 2019), and Recurrent Neural Networks (RNNs) (de Souza Pereira Moreira, 2018; An et al., 2019) were introduced to news recommendation, it was only natural to introduce Neural Attention and Transformers. Their performance increased as the models got more complex (from MLP to RNN). However, compared to Transformers, the significant performance gains from Transformers prove that they are better at capturing the meaning of text (Vaswani et al., 2017).

Neural attention (Vaswani et al., 2017) is based on the idea that a model can learn what is important when processing a vast amount of information. For instance, in the context of news recommendation, it is essential to learn which articles in the user's reading history are important for predicting the user's next click and uncovering patterns in the user's behaviour.

Attention can be used on different levels in news recommendation: on a word level to extract informative words from a news article (Wang et al., 2018; An et al., 2019; Wu et al., 2019b), on a news level to extract what information (e.g. title, topic, or abstract) is informative (Wu et al., 2019c), or on a user level to extract important features for user representations (Wu et al., 2019c). Neural attention is at the core of Transformer networks, which have achieved state-of-the-art performance on numerous Natural Language Processing (NLP) tasks.

Transformers, as introduced by Vaswani et al. (2017), have changed the NLP field. Models such as Bidirectional Encoder Representations from Transformers (BERT) (Vaswani et al., 2017) achieved significantly better performance on numerous NLP tasks due to strong text modelling ability. Furthermore, Transformers handle sequential data efficiently by processing it in parallel, while RNNs handle the data sequentially by design. At the Transformer architecture's core, the self-attention mechanism decides for each step in the input sequence what other parts of the sequence are important for that particular step. In an NLP setting, one might say that self-attention assesses each word in the sequence and calculates the importance of all other words for the meaning of that word. Wu et al. (2021a) utilize BERT as a news encoder by leveraging its text modelling abilities for creating news article representations out of the text of an article. Their PLM-NR architecture achieves new state-of-the-art performance on the Microsoft News Dataset (MIND) news recommendation benchmark. Even though these models provide very accurate recommendations, their recommendations might have adverse effects. As mentioned, solely optimizing for accuracy might lead to echo chambers and filter bubbles (Helberger, 2019). Furthermore, as McNee et al. (2006) argue, accurate recommendations might not be the most valuable recommendations. For example, during the COVID-19 pandemic, many news articles on the topic were released daily. If we were to only optimize for accuracy, users who have read COVID-19-related articles are prone to receiving many articles related to the pandemic. This might not be ideal since users might also be long-term interested in topics like sports or politics. In the next section, we introduce different metrics of diversity that have been introduced in previous works.

2.2 Diversity Metrics for Recommendation

This section discusses different measures of diversity from previous works in the recommendation field. Each of the following paragraphs explains a particular metric.

Intra-List Similarity (ILS) is the most cited measure of diversity. ILS measures the average similarity between any pair of items in a list (Ziegler et al., 2005). Often the reverse of ILS, Intra-List Distance (ILD), is used to denote the dissimilarity between two lists of recommended items. Thus, whereas ILS measures similarity, ILD measures dissimilarity. Diversity is most commonly defined as the average dissimilarity (ILD) of all the pairs of items in a user's recommended list (Raza and Ding, 2021a). ILD can be measured among different axes in neural news recommendation: item (the title and/or body of the article), category, tags, or sentiment (Helberger, 2019).

Coverage (COV) measures the percentage of distinct items that a recommender can recommend. For news recommendation, coverage refers to the percentage of articles that can be recommended (De Francisci Morales et al., 2012; Maksai et al., 2015). Where ILD attempts to measure how diverse a user's recommendations are, a high COV ensures that most of the available items get recommended to at least one user.

Novelty measures how different or unknown a recommendation is from what has been previously recommended to a user (Vargas and Castells, 2011). However, introducing novelty is more challenging in news recommendation because all news is novel by definition (Raza and Ding, 2021b). Novelty in news recommendation is often defined as the inverse popularity or the ratio of unknown items in the top-N recommendations of news items (Garcin and Faltings,

2013; Gu et al., 2014; Maksai et al., 2015; Saranya and Sudha Sadasivam, 2017; Raza and Ding, 2020).

Serendipity is a measure that combines relevance (usefulness), novelty, and unexpectedness (surprise) (Kotkov et al., 2016). Serendipity and novelty are closely related but are fundamentally different. Where novelty measures if a user is unfamiliar with an item or has not consumed it in the past, serendipity measures if a user does not expect or would not have found an item but was happy that it was recommended. For example, if the user gets recommended a news story that he has never heard of, this news story is novel to him but not serendipitous if he is not interested in that topic. On the contrary, if the user finds this news story interesting enough to change his attitude on that news category or topic, this news item is serendipitous (Asikin and Wörndl, 2014). Serendipity is relatively underused in news recommendation due to the metric's compositional nature, making it challenging to use during evaluation.

Senti is a measure of sentiment diversity. Wu et al. (2020a) proposed the metric directly for news recommendation. It measures the sentiment diversity between each candidate item in the top K recommendations and the aggregated sentiment of the articles read by the user in the past. It can also be calculated over the entire set of recommendations as $Senti_{MRR}$.

Accuracy-diversity trade-off is a measure proposed by Raza (2021). As the name suggests, it measures the trade-off between accuracy and diversity. The metric is a weighted sum of the F1 score and the mean of diversity, measured as ILD, and novelty. One drawback of this metric is that the weights are hyperparameters and are open to interpretation. For example, where one might assign a higher weight to diversity, others might find accuracy more important. The metric does not provide a genuinely optimal trade-off in all situations. In later work, Raza and Ding (2021a) redefine the trade-off metric as $tradeoff = 2 * accuracy * diversity/(accuracy + diversity)$, where the accuracy is the mean normalized Discounted Cumulative Gain (nDCG) and diversity is the mean ILD. This redefined metric has no more need for hyperparameters but also means that accuracy and diversity are weighted equally, and no prioritization can be made.

New topic ratio is a metric proposed by Qi et al. (2021) and measures the topic similarity between the recommendations and the users' historical clicks. The metric counts the number of topics in the top K recommended news articles that are not included in the topics of the user's historical clicked news. The metric is then normalized by the value of K.

Dynamism is a metric first proposed by Lathia et al. (2010) and measures the inter-list diversity between two updates. Its focus is on measuring how dynamic the recommendations are over time and be viewed as diversity over time.

Rank and Relevance sensitive Intra-List Distance (RR-ILD), as proposed by Vargas and Castells (2011), is a version of ILD that also considers the rank and relevance of the items. The metric assigns more weight to recommended items labelled as clicked and thus relevant. The metric also assigns more weight to items with higher scores, which are thus ranked higher, assuming that higher-ranked items are more important due to their increased visibility.

α-nDCG is a metric first coined by Clarke et al. (2008) and improved by Vargas (2014). It assumes that the relevance of a document can not only be judged in isolation, as is the default for nDCG, but it should be considered in the context of the preceding documents. Even though α-nDCG was initially developed for search queries, Vargas (2014) show that it can also be used in recommendation for class diversification, e.g. genres or categories.

$\alpha\beta$-nDCG proposed by Parapar and Radlinski (2021) is a metric that continues on the work of α-nDCG. This metric is a unified metric for item relevance and aspect redundancy. In the context of news recommendation, an aspect can be the category or a particular topic. Parapar and Radlinski (2021) show that $\alpha\beta$-nDCG satisfies multiple important axioms for relevant and diverse recommendations, such as non-priority on saturated aspects. This means that aspects (e.g. categories) that are already well represented in the recommendations will

not get prioritized further.

2.3 Diverse News Recommendation

In recent years, research has paid more and more attention to diversity in news recommendation. Previously focusing on providing accurate recommendations, the field has recognized the importance of diversity. However, there is still no consensus on a definition of diversity, how it should be measured, and what algorithmic changes should be made to existing state-of-the-art recommender systems. This section discusses different works in the field and the approaches for diverse news recommendation.

Wu et al. (2020a) view diversity as sentiment diversity. The example they illustrate is that a user who has read many articles about deadly accidents and crime is likely to be recommended other articles with a negative sentiment. This may not benefit users as they do not receive diverse opinions and news events that convey other sentiments. The sentiment diversity is measured via three metrics: $Senti_MRR$, $Senti$@5, and $Senti$@10. In their approach, the authors utilize a sentiment-aware news encoder by training on sentiment prediction. Additionally, a sentiment diversity regularization method is used to penalize the recommender according to the overall sentiment score of browsed news, the sentiment score of candidate news, and its predicted click score. Their results show more sentiment diverse news recommendations while maintaining state-of-the-art accuracy metrics compared to non-diverse news recommenders. Even though Wu et al. (2020a) provided a good first step towards diversity, it is only limited to sentiment diversity. It does not include other facets of diversity such as topics, political orientation, or entities. This differs from our work, where we consider diversity as the ILD, which considers more aspects captured in the embeddings.

Raza (2021) propose a temporal-aware recommender model that encodes short-term and long-term preferences separately. The idea behind this approach is that some preferences can shift often and quickly in the short term, for example, during a pandemic or natural disaster. In contrast, long-term preferences remain more or less stable. Raza (2021) use the aforementioned Accuracy-diversity trade-off metric to measure diversity. The initial findings show that a temporal-aware model provides better accuracy in terms of a higher F1 score. However, the results also show a negative correlation between the recommendation results' accuracy and diversity. When optimizing for higher accuracy, the diversity drops, and vice versa. One shortcoming of this work is that it does not include the final results and findings, only preliminary findings. This work also differs from ours in that Raza (2021) aim to achieve better diversity through architectural changes to the news recommender, while our method uses re-ranking with personalized degrees of diversity.

The work by Raza and Ding (2021a) continues the work of Raza (2021) by providing thorough experiments and results. Again, short-term and long-term preferences are modelled separately using LSTMs. The diversity metrics remain the same, namely the average dissimilarity of all the pairs of items in the user's recommendations, but this time at a specific cut-off value (K). Raza and Ding (2021a) utilize the redefined Accuracy-diversity trade-off metric, where accuracy is the mean nDCG, and ILD is the mean diversity metric. Raza and Ding (2021a) find that their proposed model, D2NN, achieves the best prediction accuracy on the NYTimes dataset and close to that on the MIND dataset while having significantly better diversity than the baselines. This results in the D2NN model having the best trade-off between accuracy and diversity. However, when we compare the accuracy, diversity, and trade-off metrics on the MIND dataset with the DKN (Wang et al., 2018) baseline, we see that the DKN and D2NN models alternate in best performance across these metrics. This is probably due to the MIND dataset containing shorter sequences, where the D2NN model's usefulness diminishes. Again, since this work is based on the work by Raza (2021), it also differs from our work in the

same way. D2NN aims to achieve better diversity through architectural changes to the news recommender, while our method uses re-ranking with personalized degrees of diversity.

Qi et al. (2021) propose a framework called PP-Rec that combines personalized recommendation scores with popularity scores for more diverse recommendations and to alleviate cold start problems. Their framework includes a popularity-aware user encoder that eliminates the popularity bias in user behaviours to model more accurate user representations. Qi et al. (2021) argue that this popularity-aware user encoder is needed because news popularity can affect users' click behaviours (Zheng et al., 2010). The diversity measures used in this paper are ILD and new topic ratio. Their results on two self-made real-world datasets show that PP-Rec achieves better accuracy and diversity than state-of-the-art baselines. However, one limitation is that the self-made datasets are proprietary and unavailable. As of writing, there are no other publicly available news recommendation datasets that include popularity data. This work differs from ours because it discounts articles by popularity, but this is the same for all users. Our method introduces personalized levels of diversity instead.

The work by Lu et al. (2020) focuses on incorporating the journalistic values of news organizations in personalization algorithms. These values are: (i) the ability to surprise readers, (ii) providing timely and fresh news, (iii) yielding more diverse reading behaviour, and (iv) increasing item coverage. The work focuses on the online evaluation and employs a Gradient Boosted Decision Tree (GBDT) (He et al., 2014) which is trained nightly on user interactions from the previous 7 days and, during inference, ranks a set of candidate news articles from the previous 7 days. Lu et al. (2020) define 'usefulness' as a set of metrics, namely: (i) diversity, measured as the *intra-list distance*, (ii) dynamism, measured as the ILD between two updates, (iii) serendipity, and (iv) COV. The authors incorporate dynamism in a re-ranking step where more recent articles are ranked higher. The results show that this increases dynamism, serendipity, and diversity without hurting accuracy. A limitation of this work is that the recommendations are only made daily and do not change within a day. Thus, the dynamism is limited to providing different recommendations for two consecutive days. However, the news field is very dynamic, and articles often stay relevant for only a very short period. So naturally, we would never want to recommend the same items from the previous day. The dynamism re-ranking differs from our approach in the sense that it is concerned with providing diverse consecutive lists of recommendations. In contrast, our approach focuses on personalized levels of diversity within each list.

Gharahighehi and Vens (2021) propose two methods to make session-based recommenders diversity-aware. The first method, called diverse neighbour, assigns higher weights for the neighbour sessions, which are other users' sessions that are similar to the user's current session and have more content diversity. The second method, called diverse candidate item, assigns a higher diversity weight for a candidate item with higher average dissimilarity with items of the user's active session. Gharahighehi and Vens (2021) use three diversity metrics: (i) ILD, (ii) RR-ILD, and (iii) covered topics, which is the average number of unique topics or keywords in the recommendations. The results show that combining both diversification methods achieves the best diversity but leads to a significant drop in accuracy. The authors conclude that choosing the most effective diversification approach depends on the session-based recommender and the dataset used. However, there is a trade-off between diversity and accuracy in all cases. Where the methods proposed in this paper work for session-based recommenders, they do not work for other state-of-the-art neural recommenders that consider entire histories instead of sessions. This work is fundamentally different from ours since it uses session-based data. This allows Gharahighehi and Vens (2021) to utilize other users' sessions and candidates diverse to the current user's session to improve diversity. Since we use aggregated data that is not split into sessions, we cannot make use of a similar approach.

Vrijenhoek et al. (2021) view news recommenders from a democratic perspective due to the

media's vital role within society. They argue that the traditional diversity measures are good at generalising but lack specificity. Furthermore, the metrics are not grounded in the normative understanding of diversity media law, fundamental rights law, democratic theory and media studies/communication science literature, as is also demonstrated by Loecherbach et al. (2020). Vrijenhoek et al. (2021) propose five diversity metrics:

- Calibration: reflects to what extent the recommendations reflect the user's preferences. A score of 0 indicates perfect calibration, whereas a higher score indicates a more significant divergence from the user's preferences.
- Fragmentation: measures the amount of overlap between news story chains shown to different users. A score of 0 indicates perfect overlap, whereas a score of 1 indicates no overlap.
- Activation: expresses whether the issued recommendations are aimed at inspiring the users to take action. A score close to 1 indicates a high amount of activating content, whereas a score close to 0 indicates more neutral content.
- Representation: reflects whether the recommendations balance different opinions and perspectives, where all are more or less equally represented. A score close to 0 indicates balance, whereas a higher score indicates larger inequalities.
- Alternative Voices: measures the relative presence of people from a minority or marginalised group in the recommendations. A higher score indicates a proportionally larger presence.

Even though ways of operationalizing the metrics are provided, they have not been applied or empirically proven in the paper. This work is naturally fundamentally different from ours. The work by Vrijenhoek et al. (2021) was intended to provide theoretical metrics for measuring diversity. In contrast, our work focuses on a practical framework that provides personalized levels of diversity to users.

Heitz et al. (2022) test a diversity-aware news recommender's utility and external effects in a news app setting. Heitz et al. (2022) view diversity in light of people's political preferences, where diverse news articles differ from the user's average political preference. The news app combines real-time news from six major German-language Swiss outlets. The participants of the experiments are assigned to a narrow group, diverse group, or control group. The narrow group receives accuracy-optimized news, the diverse group receives diversity-optimized articles, and the control group receives news in chronological order. They found that the perceived utility of the app remains equal across the diverse, narrow, and control groups. Furthermore, diverse news recommendations appear to be related to a higher tolerance for opposing views, especially for politically conservative users. Lastly, most users indicated to prefer news with multiple political views over news with majority views. Where political diversity may be necessary for parties that aggregate and recommend articles from multiple sources, this may be less relevant for news sites that make their own objective news. This work differs from ours in that Heitz et al. (2022) provide either accuracy-optimized or diversity-optimized recommendations without any middle ground that tries to combine both. In comparison, our work attempts to find this middle ground for each user individually.

Chapter 3

Methodology

In this chapter, we will discuss the methodology used in this research. In the first section, we cover a general state-of-the-art news recommendation architecture, as used in works by Wu et al. (2019c, 2021a, 2019a). In the second section, we discuss how Pre-trained Large Language Models (PLMs) fit into this architecture. Finally, the third section covers the diversification approach proposed in this paper.

3.1 News Recommendation in General

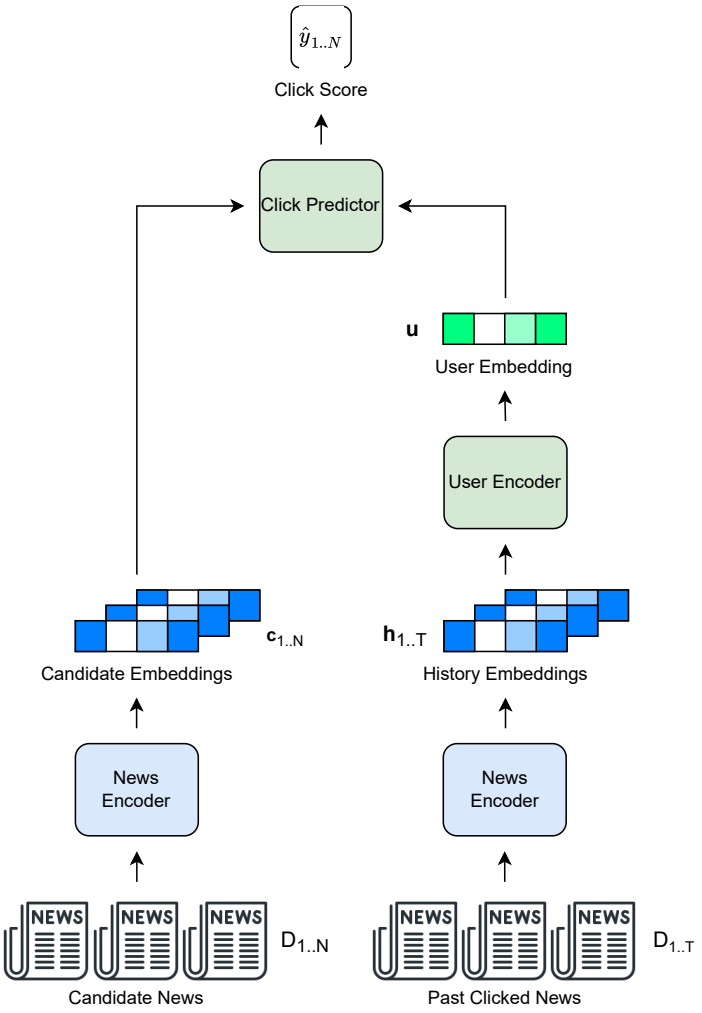

Figure 3.1: General news recommendation architecture.

A frequently used framework for state-of-the-art news recommendation is the one described in Figure 3.1. Many existing models use this framework (An et al., 2019; Okura et al., 2017; Wu et al., 2019a,c, 2021a). On the right side of the diagram, we have a *news encoder* used to learn representations/embeddings from a set of historical news articles that the user has previously clicked on, denoted as $[D_1, D_2, .., D_T]$. The resulting embeddings, $[\mathbf{h}_1, \mathbf{h}_2, .., \mathbf{h}_T]$, are dense vectors that represent the articles. These history embeddings are then passed to a *user encoder*, which aggregates these embeddings into a single user embedding of fixed dimensionality. The resulting vector, $\mathbf{u}$, represents a user's interests, as learnt from the news articles that the user has previously clicked. The same news encoder is utilised on the left side of the diagram to embed a set of candidate news articles, $[D_1, D_2, .., D_N]$, for which we want to predict whether the user will click. The resulting candidate embeddings, $[\mathbf{c}_1, \mathbf{c}_2, .., \mathbf{c}_N]$, are fed to a *click predictor* together with the user embedding, $\mathbf{u}$, to predict a click score per candidate news article. This click score ranges from 0 (very unlikely) to 1 (very likely). The news encoder often utilizes a text encoder model that generates embeddings from text besides embedding categorical features. These text encoders can be any NLP model, for example, a CNN (Park et al., 2017), RNN (Basnet and Timalsina, 2018; An et al., 2019), or self-attention (Wu et al., 2019c). The user encoder can be any model that can handle sequential data, such as a GRU (Okura et al., 2017), attention network (Wu et al., 2019a), or more complex networks that combine multi-headed self-attention and additive attention (Wu et al., 2019c). Lastly, the click predictor can be any method or model that can calculate the relevance between the user representation, $\mathbf{u}$, and the candidate news embedding, $\mathbf{c}_n$. Examples of click predictors are the inner product (Okura et al., 2017), factorization machine (Guo et al., 2017), or neural network (Wang et al., 2018).

There exist other news recommendation architectures. For example, Gharahighehi and Vens (2021) use session-based recommenders such as SKNN (Jannach and Ludewig, 2017), VSKNN (Ludewig et al., 2019b), STAN (Garg et al., 2019), and VSTAN (Ludewig et al., 2019a). However, this requires the dataset to be in a session-based format, which does not apply to the Microsoft News Dataset (MIND) dataset. Therefore, we utilize the general news recommendation architecture used by the best-performing models on the MIND leaderboard [1].

3.2 Utilizing PLMs in News Recommendation

As eluded to in the introduction, PLMs can be used as text encoders in the news encoder in the traditional news recommendation framework. PLMs can capture complex meanings and representations of textual content and are thus suitable as text encoders. Every input news article is represented by M tokens contained in the article's title and/or abstract, denoted by $[w_1, w_2, .., w_M]$. Through a series of Transformer (Vaswani et al., 2017) layers, these tokens are transformed into contextual word embeddings, denoted as $[r_1, r_2, .., r_M]$. Following Wu et al. (2021a), an attention network is used to summarize these word embeddings into a single news embedding. The PLM and attention network together form the news encoder. In the following subsections, we discuss two different PLMs and their use in news recommendation.

3.2.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a Transformer model proposed by Devlin et al. (2019) and is the most cited Transformer model. The base BERT model consists of 12 layers, a hidden dimension of 768, 12 attention heads, and 110 million parameters. BERT is well known for its text modelling capabilities and (at the time) state-of-the-art performance on numerous Natural Language Processing (NLP) tasks (Devlin et al., 2019). BERT

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

is trained using a Masked Language Modeling (MLM) pre-training method. This means that some tokens in the input are replaced with $[MASK]$ tokens, and the model is trained to predict and reconstruct the masked tokens. BERT can be used in news recommendation to create accurate representations of the article's textual content. Having accurate representations of the articles can be beneficial for providing better news recommendations (Wu et al., 2021a).

3.2.2 Fastformer

Fastformer is a Transformer model proposed to address the inefficiency of the Transformer architecture (Wu et al., 2021b). Transformer models have a quadratic complexity to the sequence length, meaning they are especially inefficient for long sequences (Wu et al., 2021b). At the core of the Fastformer architecture lies additive attention. Instead of modelling pairwise interactions using normal self-attention, additive attention first models global contexts and then further transforms each token representation based on its interaction with global context representations. Wu et al. (2021b) claim that Fastformer is significantly more efficient than many existing Transformer models and achieves new state-of-the-art performance on news recommendation.

3.3 Diversification Approach

This section covers our approach to diversifying the news recommendations. The first subsection covers the motivation and intuition behind this approach, while the second subsection elaborates on the implementation within state-of-the-art neural news recommenders.

3.3.1 Motivation and Intuition

The main idea behind our implementation is that users with diverse preferences and news consumption should not be pushed to even more diverse consumption. In contrast, users with very uniform news consumption should be nudged to read more diverse articles. Other works in diversity, such as by Wu et al. (2020a) and Raza (2021), focus on achieving better diversity as a result of architectural changes. This means that the diversity gains are generally small and the same for all users. Our approach is novel in that we treat different users differently, according to the extent of diversity in their reading behaviours.

We hypothesize that users with already diverse news consumption should not be nudged to consume even more diverse news articles, as works by Raza (2021) and Raza and Ding (2021a) prove that diversity and accuracy tend to trade off between one another. However, we want to nudge users with very undiverse reading habits to read more diverse news articles. We hypothesize that users can benefit greatly from the personalized importance of diversity in their news recommendations. Users with undiverse reading habits get nudged towards consuming more diverse news, while users with diverse reading habits do not suffer from less relevant recommendations. Therefore, we propose a personalized diversity-aware re-ranking method, called Personalized Diverse Recommendation (PD-Rec), that uses the user's history similarity to determine the importance of diversity in the recommendations. The user's history similarity can be defined as any vector similarity measure, e.g. cosine similarity or Euclidean distance, between the past articles a user has read. This weight is then multiplied by the diversity of the candidate news articles to determine the diversity scores. These diversity scores are then combined with the default relevance scores from the news recommender. The new diversity-aware news recommendation architecture is visualized in Figure 3.2.

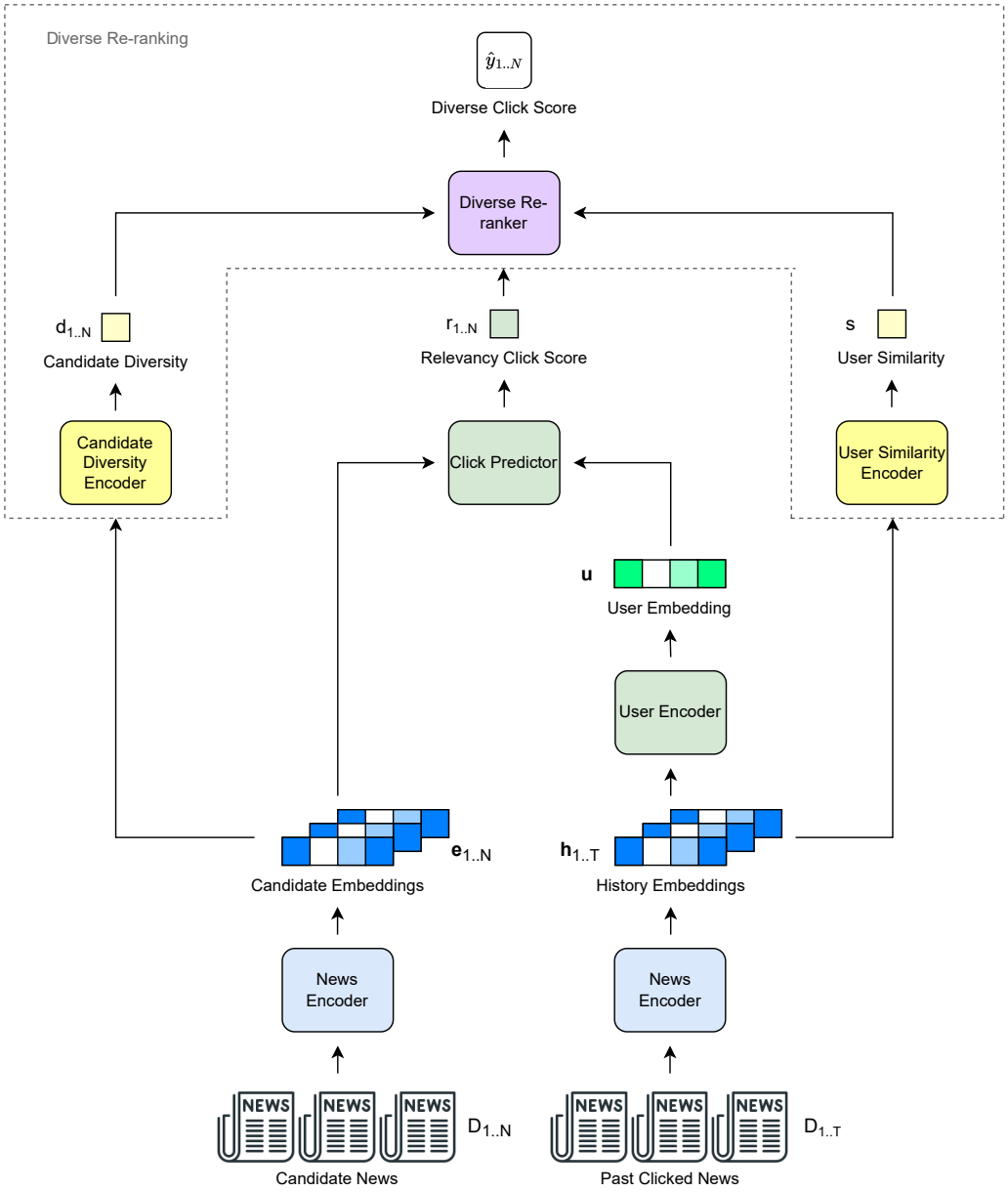

Figure 3.2: Diverse news recommendation architecture. The additional diverse re-ranking is depicted within the dashed lines, the default architecture is left unchanged. The Candidate Diversity Encoder and User Similarity Encoder use the article embeddings from an external model.

3.3.2 Implementation

We can summarize the workings of PD-Rec using Algorithm 1.

Algorithm 1 Personalized Diverse Recommendation re-ranking algorithm.

Require:
Historical news vectors: $\mathbf{h}_{1..T}$
Candidate news vectors: $\mathbf{c}_{1..N}$
Candidate relevancy scores: $r_{1..N}$
Vector similarity measure: $\text{sim}(\mathbf{x}, \mathbf{y})$
Diverse ranking function: $\text{div\_rank}(s, d_n, r_n)$

procedure DIVERSE_RERANKING($\mathbf{h}_{1..T}, \mathbf{c}_{1..N}, r_{1..N}, \text{sim}(\mathbf{x}, \mathbf{y}), \text{div\_rank}(s, d_n, r_n)$)
$s = \frac{1}{T} \sum_{i=0}^{T} \frac{1}{1-T} \sum_{\substack{j=0 \\ i \neq j}}^{T} \text{sim}(\mathbf{h}_i, \mathbf{h}_j)$ ▷ Calculate the history similarity
for $\mathbf{c}_n$ in $\mathbf{c}_{1..N}$ and r_n in $r_{1..N}$ **do**
$d_n = \frac{1}{N-1} \sum_{\substack{i=0 \\ i \neq n}}^{N} 1 - \text{sim}(\mathbf{c}_n, \mathbf{c}_i)$ ▷ Calculate the candidate diversity
$\hat{y}_n = \text{div\_rank}(s, d_n, r_n)$ ▷ Calculate diversified click score
end for
Re-rank based on $\hat{y}_{1..N}$
end procedure

PD-Rec consists of four primary parts: the external news encoder, which is different from the one used in the general news recommendation architecture, the history similarity encoder, the candidate diversity encoder, and the diverse re-ranker. The external news encoder is a PLM, which is frozen and only utilized during validation and inference, that encodes the articles for similarity calculation. There are three reasons why we use an external model to do this.

Firstly, during preliminary experiments, we found that the diversity when using the model's article embeddings can vary significantly among the datasets and optimizers. For example, Li et al. (2021) find that SGD is generally poor for embedding tasks, while Smith et al. (2020) prove that SGD tends to generalize better to unseen data. Secondly, we can use an external model to ensure that the diverse re-ranker works similarly across all news recommendation models. Thus, the diverse re-ranking step will not be affected by how the news recommendation models embed the news articles. Thirdly, using an external model makes the models and datasets directly comparable. Since we encode the news articles similarly for all models, the resulting Intra-List Distance (ILD) and Rank and Relevance sensitive Intra-List Distance (RR-ILD) can be compared across the different models and datasets.

The history similarity encoder calculates the average similarity between the vector representations of the user's read articles. To emphasize, we utilize the embeddings from the external news encoder. This similarity is then converted to a range between 0 and 1, where 0 indicates a very diverse set of read articles and 1 indicates a very undiverse set of read articles. Equation 3.1 summarizes the history similarity encoder, where the history similarity is denoted as s.

$$s = \frac{2}{T(T-1)} \sum_{i=0}^{T} \sum_{j=i+1}^{T} \text{sim}(\mathbf{h}_i, \mathbf{h}_j) \tag{3.1}$$

To provide a good intuition of how the history similarity encoder works, we sampled two users with a short history length (four articles) with either a low or a high history similarity.

Figure 3.3 shows these two users. We can see from the article titles that user $U401044$ has a very diverse reading history. The articles have no overlap in terms of category or domain. On the other hand, we can see that user $U248349$ only reads articles about the NFL and even has read the same article twice. Thus, user $U248349$ has far less diverse reading habits than user $U401044$.

Figure 3.3: Two users with a history length of 4 sampled from the MIND development dataset. Similarity is measured using the embeddings from an external news encoder.

The candidate diversity encoder is very similar to the history similarity encoder. However, instead of calculating the similarity over the user's read articles, it calculates the diversity among the candidates. Again, to emphasise, we use the embeddings from the external news encoder. This time we calculate the distance, or dissimilarity, instead of the similarity. This can be done by using the inverse of the similarity. This means that we still maintain the 0 to 1 range. Candidates with a candidate diversity of 0 are very similar to the other candidates, while those with a candidate diversity of 1 are unique among the other candidates. We denote the candidate diversity of each candidate as d_n in Equation 3.2.

$$d_n = \frac{1}{N-1} \sum_{\substack{i=0 \\ i \neq n}}^{N} 1 - \text{sim}(\mathbf{c}_n, \mathbf{c}_i) \tag{3.2}$$

Again, to provide a good intuition of how the candidate diversity encoder works, we sampled two entries with four candidate articles with either low or high candidate diversity. Figure 3.4

visualizes these two entries. We can see that entry 16269 has low candidate diversity across all candidates. This can be attributed to the fact that all articles are concerned with US politics and are thus closely related. On the contrary, the candidates in entry 205466 are very different from one another. Looking at the titles, we see that the candidates have no overlap indeed in terms of categories and domains.

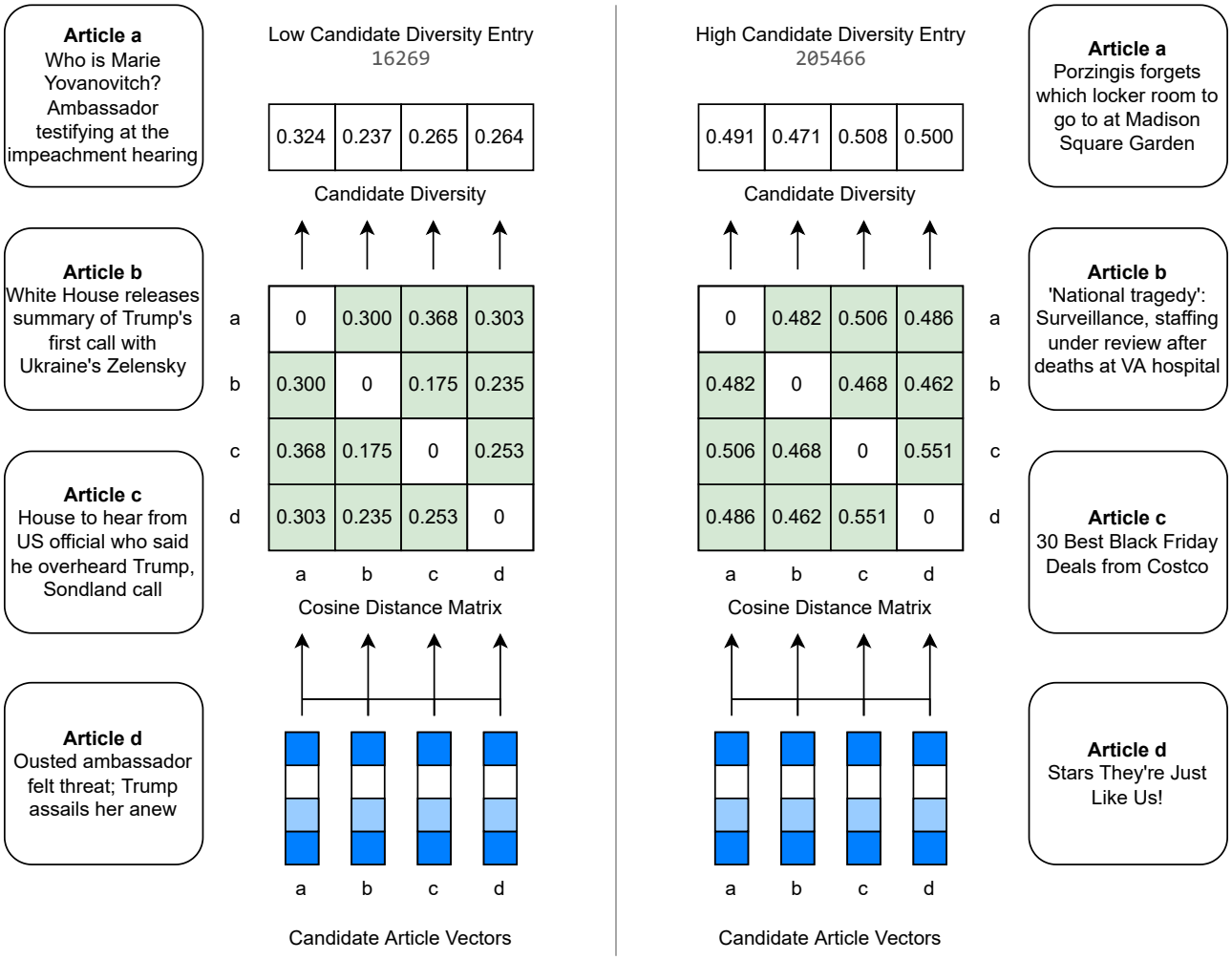

Figure 3.4: Two entries with 4 candidate articles sampled from the MIND development dataset. Diversity is measured using the embeddings from an external news encoder.

Lastly, we need the diverse re-ranker to combine the history similarity, candidate diversity, and relevancy click score into a diverse click score. The diverse ranker weights the importance of the candidate diversity by the user's history similarity and combines this with the relevancy click score (r_n). Equation 3.3 summarizes the diverse ranker, where the diverse click score for each candidate is denoted as $\hat{y}$.

$$\hat{y}_n = \text{div\_rank}(s, d_n, r_n) \tag{3.3}$$

The diverse re-ranker can be any equation combining the user similarity s, candidate diversity d_n, and candidate relevancy r_n. Therefore, we include this in our hyperparameter search. Section 4.4.5 explains the different functions we investigate as diverse re-ranker.

Chapter 4

Experimental Setup

In the chapter, we cover the setup of the experiments performed in this study. Firstly, we elaborate on the experiments that are conducted throughout this paper. Secondly, we describe and analyze the datasets used in the experiments. Thirdly, we present the evaluation metrics. Lastly, we elaborate on our hyperparameter search.

For all experiments, a *Standard_NC4as_T4_v3* cluster from Microsoft Azure Databricks [1] is used. This cluster contains an AMD EPYC 7V12 CPU, 28GB RAM, and NVIDIA T4 GPU with 16GB of video memory.

4.1 Experiments

In this section, we elaborate on the experiments performed in this paper. The first subsection covers the experiments used to determine how diverse current state-of-the-art attention-based news recommenders already are. The second subsection explains the experiments conducted to investigate the use of personalized levels of diversity in news recommendation.

4.1.1 Diversity of Attention-based News Recommenders

To find out how diverse state-of-the-art attention-based news recommender systems are, we compare NRMS (Wu et al., 2019c), NAML (Wu et al., 2019a), and PLM-NR (Wu et al., 2021a). These models all employ attention mechanisms to capture user preferences from historical clicks and use this information to recommend a new article. We compare these models among each other but also with two baselines. The first is a diversity baseline (denoted as F-DIV) that ranks candidates solely based on their diversity compared to all other candidates. The second is a random baseline (denoted as RAND) that orders the candidates randomly. We also compare the models against LSTUR (An et al., 2019), the previous state-of-the-art before the introduction of attention-based news recommenders. LSTUR (An et al., 2019) uses LSTMs to learn long- and short-term reading interests separately and concatenates these embeddings to create a user representation. We compare the models in an offline setting on the test partitions of the Microsoft News Dataset (MIND) and RTL - News Recommendation (RTL-NR) datasets. All experiments are repeated 3 times, and we report the means of the evaluation metrics. Since we use an external news encoder, we can directly compare all metrics between the different models and datasets.

[1] https://azure.microsoft.com/en-us/services/databricks/

4.1.2 Impact of Personalized Diverse Re-ranking on News Recommendation

To examine the impact of our diverse re-ranking method, we compare NRMS (Wu et al., 2019c), NAML (Wu et al., 2019a), and PLM-NR (Wu et al., 2021a) with and without the Personalized Diverse Recommendation (PD-Rec) re-ranking on the test partitions of the MIND and RTL-NR datasets. Again, we repeat all experiments 3 times and report on the means of the evaluation metrics.

Furthermore, we also investigate the impact of the personalized diversity weights. We do this by ablating the history similarity (s) score and using fixed values instead. We compare the diverse re-ranking function with three other versions that use a fixed value for s across all users. This demonstrates how personalized levels of diversity impact the accuracy and diversity of the recommendations. Specifically, we compare against three variations that use $s \in \{0.2, 0.5, 0.8\}$. We do this using the NAML model since the PD-Rec re-ranking is model agnostic and works similarly for all the different models. We evaluate both on the MIND and RTL-NR datasets.

4.2 Datasets

In this section, we discuss the datasets that are used in our experiments. The first dataset is the open-source English MIND dataset (Wu et al., 2020b), while the second is a proprietary Dutch dataset, which we refer to as RTL-NR. In the final subsection, we compare the datasets on the following characteristics: history length, title length, abstract length and the number of candidates and positives in the subsets.

4.2.1 MIND

MIND (Wu et al., 2020b) is a dataset aimed at personalized news recommendation. This dataset is constructed from click logs of 1 million users and more than 160.000 English news articles on the Microsoft News website. For each news article, the textual content such as title, abstract, and body are provided, as well as the category and subcategory. To construct the dataset, 1 million users with at least 5 news click records during the 6 weeks from October 12th to November 22nd, 2019, were sampled. The samples in the last week (week 6) are used for testing, while the samples in the fifth week are used for training. The validation set is constructed from the last day of the fifth week. The user's history is constructed from the first four weeks for the training set and the first five weeks for the test set. To train a recommender, the dataset also provides negative samples. These negative samples are impressions a user has seen but not clicked on. Note that the clicks are only gathered on the homepage, which means that clicks from articles to other articles (e.g. via links to earlier articles within an article) are not included. Wu et al. (2020b) show that MIND provides a good benchmark for personalized news recommendation through a comparative study of several state-of-the-art recommendation methods. MIND is openly available [2]. Specifically, we will use the entire dataset denoted as *large* for both the training and validation sets. Since the MIND dataset does not provide labels for the test set, we split the validation set 50-50 into a new validation and test set.

4.2.2 RTL-NR

RTL-NR is a dataset constructed from the RTL Nieuws website [3]. RTL is a major private media company with a significant presence in the Netherlands. This includes the RTL Nieuws

[2] https://msnews.github.io/
[3] https://www.rtlnieuws.nl/

website, with unbiased and unaffiliated news coverage. To construct a news recommendation dataset, we aggregated all interactions of identifiable users on the homepage in the period between January 1st 2022 and February 22nd 2022. Identifiable users are those who are either logged in or have accepted cookies. Otherwise, we are not able to log their behaviour. We follow the work by Wu et al. (2020b) as closely as possible since MIND has proven to be a good dataset for news recommendation.

The training set was constructed from the interactions in the first two weeks of February, from the 1st to the 14th, with the interactions in the month of January as the user history. The third week of February, from the 15th to the 22nd, was used for evaluation. This week was split chronologically into an evaluation and test set. Following Wu et al. (2020b), we filtered out all users with fewer than five interactions within the collection period. Furthermore, we aggregated all interactions of a user within 15 minutes into a single impression. The resulting dataset details are summarized in Table 4.1 and also compared to the MIND dataset.

Table 4.1: MIND (our version) and RTL-NR frequencies.

Characteristic	MIND	RTL-NR
Training impressions	2,186,683	1,302,232
Validation impressions	182,600	364,422
Test impressions	182,600	364,423
Total unique articles	161,013	6,698

The RTL-NR dataset differs from the MIND dataset in two major aspects. The first difference is that in the MIND dataset, the user history is not updated with interactions that occurred earlier in the training or validation set. However, we have opted to update the user's history based on all interactions for the RTL-NR dataset, as this provides a more realistic dataset for real-time news recommendation. The second difference is that at RTL, we have no way to track what a user has seen but only have records of what a user has clicked. This means that, unlike the MIND dataset, we have no known negatives (items which we know the user has seen but has not clicked on). Therefore, we sampled interactions from other users within a 3-minute window around the user's click as negatives. As a result, we were able to reconstruct what articles have been on the homepage accurately, but we do not know which articles the user has seen due to scrolling. In short, our sampled negatives are likely less representative of what the user has seen. However, they are still an accurate representation of what was displayed on the homepage at the time.

4.2.3 Dataset Characteristics

We analyzed the MIND and RTL-NR datasets on five different characteristics. The first two are the token lengths of the titles and abstracts. We use a Bidirectional Encoder Representations from Transformers (BERT) tokenizer to determine the token lengths over the entire dataset (train, validation, and test). The third characteristic is the history length, calculated as the number of articles in the users' reading history. The fourth and fifth characteristics are the number of candidate news articles and positives in the different subsets. We make this comparison since these characteristics can influence the difficulty of the dataset. For example, suppose there are many candidates and only a few positives in the validation or test set. In that case, it is harder for the model to pick a correct recommendation than when there are few candidates and relatively more positives (e.g. picking 2 positives out of 40 candidates is harder than picking 3 positives out of 30 candidates).

Table 4.2: Comparison of the MIND and RTL-NR datasets.

Characteristic	Value	MIND	RTL-NR
Title Length	Mean	15.8	14.9
	Min	4	6
	Max	132	28
	90%	22	19
Abstract Length	Mean	51.1	61.2
	Min	3	3
	Max	626	143
	90%	101	85
History Length	Mean	32.9	15.4
	Min	1	1
	Max	801	515
	90%	79	36
Num candidates training	Mean	37.4	36.7
	Std.	38.7	7.7
Num positives training	Mean	1.52	1.42
	Std.	1.17	0.86
Num candidates validation	Mean	37.5	35.5
	Std.	39.6	7.2
Num positives validation	Mean	1.53	1.39
	Std.	1.17	0.82
Num candidates test	Mean	37.3	29.9
	Std.	39.6	10.1
Num positives test	Mean	1.53	1.34
	Std.	1.17	0.75

Table 4.2 summarizes these characteristics, with exact distributions available in Appendix A. When we compare the RTL-NR dataset with the MIND dataset, we find that the token title lengths are very comparable. However, we see a big difference in the abstract length. This can be attributed to the fact that the RTL-NR dataset's abstracts are the articles' leads. Leads are short bold-faced summaries at the start of an article. These leads are much shorter than normal abstracts. One significant difference, however, is that the history length of users in the RTL-NR dataset is much shorter than those of users in the MIND dataset. This shorter history length could make the RTL-NR a much harder dataset since the models have less information on the user's reading preferences.

When we compare the candidate and positive distributions of the RTL-NR and MIND datasets, we see that the averages are very comparable. However, the standard deviations of the RTL-NR dataset are significantly lower than those of the MIND dataset. This means that there is overall less variance in the number of candidates and the number of positives in the RTL-NR dataset than in the MIND dataset. Figure 4.1 shows the distributions of MIND (left) and RTL-NR (right) side-by-side. Looking at the distributions, the difference becomes even more apparent. Whereas the RTL-NR follows an approximately symmetric distribution, the MIND dataset is heavily skewed to the left.

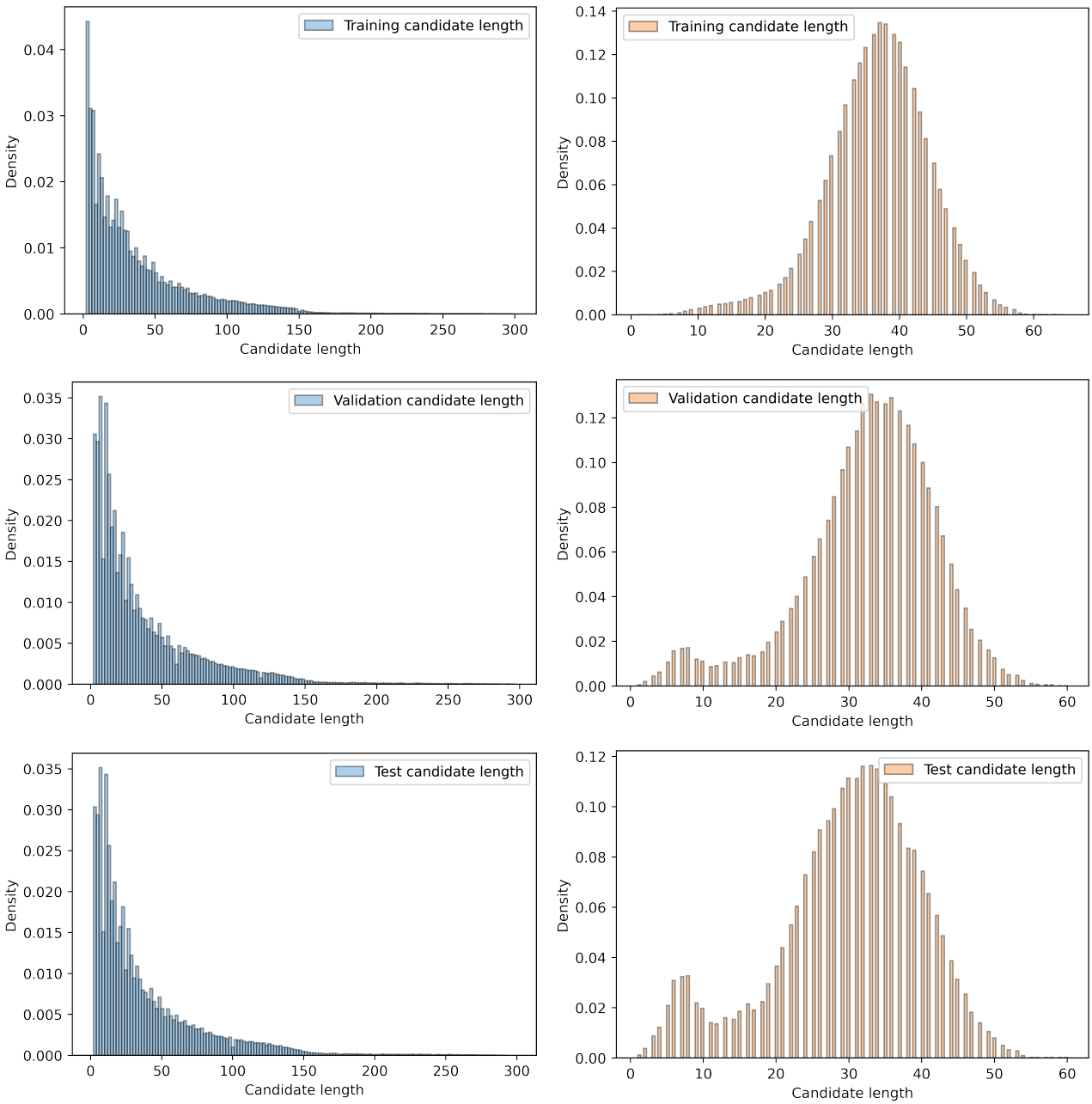

Figure 4.1: Comparison of the candidate length distributions of the MIND (left) and RTL-NR (right) datasets. The difference in distributions might be attributed to the fact that RTL-NR is simulated data, as discussed in section 4.2.2

4.3 Evaluation Metrics

This section discusses the relevant evaluation metrics. The first subsection covers recommendation accuracy metrics. The second subsection states the diversity metrics that are used in the experiments.

4.3.1 Recommendation Accuracy Metrics

Following Wu et al. (2020b), we measure the evaluation accuracy using Area Under the Curve (AUC), Mean Reciprocal Rank (MRR), and normalized Discounted Cumulative Gain (nDCG). AUC measures the area under the Receiver Operating Characteristic (ROC) curve (Bradley, 1997). The ROC curve combines the True Positive Rate (TPR) and False Positive Rate (FPR) into a single graph representing the accuracy. AUC can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example. AUC ranges from 0.0 (all predictions are incorrect) to 1.0 (all predictions are correct).

MRR (Craswell, 2009) is a straightforward metric that measures where the first relevant item in the list of recommendations appears. The higher the score, the higher the first relevant item is ranked within the recommendations.

nDCG (Wang et al., 2013) is a measure of ranking quality. Cumulative Gain (CG) is the sum of graded relevance values in the list of recommendations, as provided by truth labels. A problem with CG is that it does not consider the list's ordering. As a result, all items are weighted equally, while a list of recommendations should score higher when higher-ranked items are relevant. Discounted Cumulative Gain (DCG) addresses this problem by penalizing highly relevant documents that appear lower in the ranked list by reducing the relevance value logarithmically proportional to the position of the result. Lastly, we need to normalize for varying lengths in lists of recommendations. Therefore, normalized Discounted Cumulative Gain (nDCG) was introduced. The metric is commonly measured up to a certain cut-off point, K, where common values are $K = 5$ and $K = 10$. These are denoted as nDCG@5 and nDCG@10.

4.3.2 Diversity Metrics

To measure diversity, we employ several metrics. Similar to nDCG, all diversity metrics are measured at a certain cut-off point, K, where common values are $K = 5$ and $K = 10$. Thus, all diversity metrics are either denoted as @5 or @10, like the works by Raza (2021) and Qi et al. (2021). The first of which is the Intra-List Distance (ILD). ILD is the inverse of the cosine similarity between all pairs within a list of recommendations.

The second metric we employ is Coverage (COV). COV measures the percentage of articles that the model can recommend. A higher COV indicates that a larger portion of the total article base gets recommended to at least one user.

The third and final metric is Rank and Relevance sensitive Intra-List Distance (RR-ILD). This version of ILD also considers the rank and relevance of the items. RR-ILD assigns more weight to recommended items labelled as clicked and those who appear higher in the ranked list.

4.4 Hyperparameter Search

This section covers the hyperparameter search conducted to find the most optimal hyperparameters. First, we do an extensive hyperparameter search for the RTL-NR dataset, where we examine different optimizers, learning rates, and negative ratios. Second, we investigate

using two different text encoders within the PLM-NR architecture. Third, we examine the optimal article title and abstract text lengths regarding the performance/efficiency trade-off. Fourth, we examine the influence of the history length in user encoding on both performance and efficiency. Fifth, we investigate two different vector similarity measures within the diverse re-ranker. Lastly, we examine different diverse re-ranking functions that combine relevancy and diversity scores.

Throughout all experiments and during hyperparameter search, we use the following features: title text, abstract text, category, and subcategory. We also use the *paraphrase-multilingual-mpnet-base-v2* (Reimers and Gurevych, 2020) model from the Sentence Transformers library (Reimers and Gurevych, 2019) as our external news encoder. During both the hyperparameter search and experiments, we train for a maximum of 10 epochs with an early stopping criterion that stops training if the validation AUC has not increased with at least 0.01 combined for 3 epochs. We report the metrics of the best epoch.

4.4.1 RTL-NR Training Parameters

Finding the proper training parameters is crucial for our newly created RTL-NR dataset. The first step is to find the best optimizer and learning rate. We compare Stochastic Gradient Descent (SGD, Stochastic Gradient Descent with 0.9 momentum (SGD+M), Adam (Kingma and Ba, 2014), and AdamW (Loshchilov and Hutter, 2017). We use a learning rate of 1e-4 and a negative ratio of 4 across all optimizers as a starting point. We do this for the three different architectures; NRMS, NAML, and PLM-NR with BERT as text encoder. However, as PLM-NR with BERT is prone to returning NaN loss values during training when learning rates are set too high (as will be explained later in section 5.3.1), we use a learning rate of 1e-5 for the PLM-NR architecture on the RTL-NR dataset.

Now that the choice of the optimizer is settled, we need to find the best learning rate. We try three different values, namely 1e-3, 1e-4, and 1e-5. We use the best optimizer for each model, as discovered previously. Again, we use a negative ratio of 4.

Now that the best optimizer and learning rate are found, we need to find the optimal value for the negative ratio. The negative ratio is the number of negatives per positive sampled during training. We investigate three different values: 2, 4, and 6.

4.4.2 Choice of Text Encoder

We start by examining the choice of PLM as text encoder. The text encoder choice is based not only on performance metrics but also on execution time. Since news recommendation is a time-sensitive task, real-time recommendation means that latency is crucial. Specifically, we consider the following PLM encoder models:

- BERT: we use the *bert-base-uncased* pre-trained checkpoint on the MIND dataset and *GroNLP/bert-base-dutch-cased* on the RTL-NR dataset. Both checkpoints can be found in the Huggingface library (Wolf et al., 2019).
- Fastformer: we use the GloVe embeddings (Pennington et al., 2014) to initialise the text embeddings for the MIND dataset and the Dutch FastText embeddings (Bojanowski et al., 2017) for the RTL-NR dataset.

4.4.3 Data Lengths

As we determined in section 4.2.3, 90% of the news article title and abstract lengths are within 22 and 101 tokens, respectively. However, Wu et al. (2019c) utilize a length of 20 tokens for

the title and 50 tokens for the abstract. The benefit of shorter sequences is that it allows for larger batch sizes. Therefore, we compare the different lengths.

We also investigate the influence of history length on performance. In section 4.2.3, we found that 90% of the users' histories contain 79 or fewer articles. However, Wu et al. (2019c) utilize a history length of 50 articles. This seems to indicate that historical clicks from long ago are cut off for a large percentage of users. However, shorter histories can lead to larger batch sizes and faster training. Therefore, we investigate the impact of different history lengths on performance.

4.4.4 Vector Similarity Measure

We also investigate two vector similarity measures to calculate the user's history similarity and candidate diversity within the diverse re-ranker. Our first measure is cosine similarity, which calculates the cosine of the angle between two vectors. Since cosine similarity is a metric in the range $[-1, 1]$, we need to convert this to a $[0, 1]$ range to get a weight that can be used in a weighted sum. Equation 4.1 summarizes how cosine similarity can be used to calculate the similarity between two vectors, such that they range from 0 (very dissimilar) to 1 (very similar).

$$\text{sim}(\mathbf{h}_i, \mathbf{h}_j)_{CS} = (1 + \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{||\mathbf{h}_i|| \times ||\mathbf{h}_j||}) \cdot \frac{1}{2} \tag{4.1}$$

The second measure is Euclidean distance, which calculates the square root of the sum of squared differences between two vectors. In this case, we are using a distance metric instead of a similarity metric. Thus, to calculate the similarity, we need to take the inverse of the Euclidean distance. However, Euclidean distance is a metric with a range of $[0, \infty]$; thus, we need to convert this to a $[0, 1]$ range as well. Equation 4.2 shows how Euclidean distance can be used to measure vector similarity with a range from 0 (very dissimilar) to 1 (very similar).

$$\text{sim}(\mathbf{h}_i, \mathbf{h}_j)_{ED} = \frac{1}{1 + ||\mathbf{h}_i - \mathbf{h}_j||} \tag{4.2}$$

We compared the two different vector similarity measures within the diverse re-ranker on the validation set. However, we still calculate the intra-list distance using cosine similarity, such that the numbers are comparable.

4.4.5 Diverse Re-ranking Function

Lastly, we also investigate the effect of different diverse re-ranking functions. The first function is a simple weighted sum that uses the user similarity to assign importance to the diversity of the articles. The relevancy scores are first passed through a softmax function, so all values are within a $[0, 1]$ range. We denote this as the naive re-ranker and is shown in Equation 4.3.

$$\begin{aligned} r_n &= \text{softmax}(r_n) \\ \text{div\_rank}(s, d_n, r_n)_{naive} &= (s \cdot d_n) + ((1 - s) \cdot r_n) \end{aligned} \tag{4.3}$$

The second function is the same as the naive re-ranker, but it bounds the value of s by a certain minimum and maximum value. By giving a minimum value, we ensure that diversity has a certain importance in the final score for all users. In contrast, by giving a maximum value, we ensure that diversity is not assigned too much importance. Thus, this function takes two additional parameters: s_{max} and s_{min}, which denoted the maximum and minimum values of s. This function is denoted as the bound re-ranker and is summarized in Equation 4.4. We find that, for the MIND dataset, the mean user similarity is 0.563, with a minimum of 0.450

and a maximum of 1.000. However, the users with a user similarity of 1 are edge cases who have only consumed the same news article multiple times. For the RTL-NR dataset, the mean user similarity is 0.599, with a minimum of 0.444 and a maximum of 1.000. Therefore, we try two different versions of this bounded re-ranking function. The first function, denoted as BO_C, is a conservative function that uses $s_{max} = 0.55$ and $s_{min} = 0.45$, which means that it should generally assign less importance to diversity than the naive re-ranker. The second version, denoted as BO_D, is a more diversity-focused function that uses $s_{max} = 0.80$ and $s_{min} = 0.55$.

$$\begin{aligned} r_n &= \text{softmax}(r_n) \\ s &= min(max(s, s_{min}), s_{max}) \\ \text{div\_rank}(s, d_n, r_n, s_{max}, s_{min})_{bound} &= (s \cdot d_n) + ((1 - s) \cdot r_n) \end{aligned} \tag{4.4}$$

The third function is a normalized version of the naive re-ranker. We found through empirical testing that both the user similarity and candidate diversity do not utilize the entire $[0, 1]$ range. Again, we find that the mean user similarity is 0.563, with a minimum of 0.450 and a maximum of 1.000 for the MIND dataset. Meanwhile, the mean candidate diversity is 0.452, with a minimum of 0.177 and a maximum of 0.550. For the RTL-NR dataset, the mean user similarity is 0.599, with a minimum of 0.444 and a maximum of 1.000. The mean candidate diversity is 0.441, with a minimum of 0.245 and a maximum of 0.510. Thus, this function takes four additional parameters: s_{min}, s_{max}, d_{min}, and d_{max}. This function is denoted as the normalized re-ranker and is summarized in Equation 4.5. For the MIND dataset we use the following values: $s_{min} = 0.44$, $s_{max} = 1.00$, $d_{min} = 0.16$, $d_{max} = 0.56$. For the RTL-NR dataset we use the following values: $s_{min} = 0.43$, $s_{max} = 1.00$, $d_{min} = 0.23$, $d_{max} = 0.52$.

$$\begin{aligned} r_n &= \text{softmax}(r_n) \\ s &= \frac{s - s_{min}}{s_{max} - s_{min}} \\ d_n &= \frac{d_n - d_{min}}{d_{max} - d_{min}} \\ \text{div\_rank}(s, d_n, r_n, s_{min}, s_{max}, d_{min}, d_{max})_{normalized} &= (s \cdot d_n) + ((1 - s) \cdot r_n) \end{aligned} \tag{4.5}$$

Chapter 5

Results

This chapter summarizes the results of each of the experiments. The first two subsections address the results of our experiments, while the third subsection elaborates on the results of the hyperparameter search.

5.1 Diversity of Attention-based News Recommenders

In this section, we aim to answer the first research question; how diverse are state-of-the-art attention-based news recommenders?

Table 5.1 summarizes the results of the different architectures and baselines when evaluated on the Microsoft News Dataset (MIND) test dataset. One of the most interesting findings is that NRMS outperforms the PLM-NR model with Fastformer as text encoder, which contradicts the claims made by Wu et al. (2021b). This finding surprised us. We can only imagine the difference comes from different implementations since Wu et al. (2021b) do not provide the code for this experiment and only provide the Fastformer model itself. Furthermore, we can see that the attention-based recommenders significantly outperform the LSTUR model in terms of accuracy but are not considerably more diverse. We also find no significant differences in diversity among the attention-based recommenders, and these are also not significantly more diverse than the LSTUR model. We find that the F-DIV and RAND baselines provide the best diversity, whereas the recommender models all perform substantially lower in this regard. We hypothesize that this stems from the fact that the recommenders learn patterns in the data, and these patterns might not be diverse by themselves (i.e. people tend to read the same topics). Interestingly, we find that better diversity and better coverage do not necessarily go hand-in-hand. Whereas the F-DIV baseline provides the best diversity, its coverage is the worst of all models and baselines.

Table 5.2 summarizes the results of the different architectures and baselines when evaluated on the RTL - News Recommendation (RTL-NR) test dataset. Interestingly, we find that the PLM-NR model with Bidirectional Encoder Representations from Transformers (BERT) as text encoder significantly outperforms all the other models in terms of accuracy metrics. This is contrary to what we saw for the MIND dataset, where NRMS outperforms the PLM-NR model with Fastformer as text encoder. This means that the results on the RTL-NR dataset validate the claim made by Wu et al. (2021a). As we found on the MIND dataset, the attention-based recommenders significantly outperform the LSTUR model in terms of accuracy but are not more diverse. Again, we find that the F-DIV and RAND baselines provide the best diversity, whereas the recommender models all perform significantly lower in this regard. Furthermore, we again see that better diversity and better coverage are not tied together, with the F-DIV baseline being the worst in terms of coverage.

To answer the question of how diverse state-of-the-art attention-based news recommenders

Table 5.1: Performance of attention-based news recommenders and baselines evaluated on the MIND test set. All numbers are means across three separate runs. Bold-faced numbers are the best, while the underlined are the second best.

		Models			Baselines		
Type	Metric	PLM-NR	NRMS	NAML	F-DIV	RAND	LSTUR
Acc	AUC	0.6778	**0.6889**	0.6747	0.4303	0.4999	0.6230
	MRR	0.3239	**0.3293**	0.3228	0.1928	0.2189	0.2827
	nDCG@5	0.3593	**0.3680**	0.3583	0.1855	0.2234	0.3117
	nDCG@10	0.4239	**0.4320**	0.4230	0.2464	0.2865	0.3760
Div	ILD@5	0.4250	0.4274	0.4261	**0.4748**	0.4489	0.4219
	ILD@10	0.4354	0.4362	0.4352	**0.4673**	0.4489	0.4337
	RR-ILD@5	0.0042	0.0043	0.0043	**0.0047**	0.0045	0.0042
	RR-ILD@10	0.0043	0.0044	0.0043	**0.0047**	0.0045	0.0043
	COV@5	0.0490	0.0511	0.0501	0.0333	**0.0607**	0.0436
	COV@10	0.0629	0.0645	0.0636	0.0469	**0.0699**	0.0582

Table 5.2: Performance of attention-based news recommenders and baselines evaluated on the RTL-NR test set. All numbers are means across three separate runs. Bold-faced numbers are the best, while the underlined are the second best.

		Models			Baselines		
Type	Metric	PLM-NR	NRMS	NAML	F-DIV	RAND	LSTUR
Acc	AUC	**0.7044**	0.6534	0.6612	0.3941	0.5006	0.6120
	MRR	**0.2906**	0.2142	0.2200	0.1293	0.1534	0.1860
	nDCG@5	**0.3025**	0.2041	0.2125	0.1034	0.1343	0.1772
	nDCG@10	**0.3730**	0.2882	0.2971	0.1482	0.1950	0.2614
Div	ILD@5	0.4110	0.4055	0.4333	**0.4766**	0.4413	0.4349
	ILD@10	0.4207	0.4179	0.4347	**0.4666**	0.4413	0.4339
	RR-ILD@5	0.0041	0.0040	0.0043	**0.0048**	0.0044	0.0044
	RR-ILD@10	0.0042	0.0042	0.0043	**0.0047**	0.0044	0.0043
	COV@5	0.0916	0.0479	0.0831	0.0651	**0.0935**	0.0892
	COV@10	**0.0935**	0.0706	0.0905	0.0832	**0.0935**	0.0926

are, we can say that these models (PLM-NR, NRMS, and NAML) are not more diverse than the previous state-of-the-art models (LSTUR). Furthermore, among these models, there is very little or no difference in diversity. However, we did find that the differences in diversity were more significant on the RTL-NR dataset than on the MIND dataset.

5.2 Impact of Personalized Diverse Re-ranking on News Recommendation

In this section, we aim to answer the second research question; how can personalized levels of diversity contribute to better diversity in news recommendation?

Table 5.3 summarises the results of the different architectures and their re-ranked versions when evaluated on the MIND dataset. The personalized re-ranking step leads to a drop of 2-4 percentage points across all accuracy metrics, except for Area Under the Curve (AUC), where

we see drops of around 8 percentage points. However, we also gain 2-3 percentage points in the Intra-List Distance (ILD) and Rank and Relevance sensitive Intra-List Distance (RR-ILD). This seems to suggest that the personalized re-ranking step can improve diversity, but still at the cost of significant drops in accuracy. Interestingly, the personalized diverse re-ranking seems to hurt Coverage (COV), with a drop across all models. This means that personalized diverse re-ranking also hurts diversity, as it results in less of the total articles being recommended to users.

Table 5.3: Performance of attention-based news recommenders, with and without re-ranking, evaluated on the MIND test set. All numbers are means across three separate runs.

		PLM-NR		NRMS		NAML	
Type	Metric	Default	Re-ranked	Default	Re-ranked	Default	Re-ranked
Acc	AUC	0.6778	0.5982	0.6889	0.6100	0.6747	0.5918
	MRR	0.3239	0.3047	0.3293	0.3125	0.3228	0.3018
	nDCG@5	0.3593	0.3332	0.3680	0.3418	0.3583	0.3282
	nDCG@10	0.4239	0.3870	0.4320	0.3956	0.4230	0.3819
Div	ILD@5	0.4250	0.4539	0.4274	0.4558	0.4261	0.4570
	ILD@10	0.4354	0.4596	0.4362	0.4600	0.4352	0.4608
	RR-ILD@5	0.0042	0.0045	0.0043	0.0046	0.0043	0.0046
	RR-ILD@10	0.0043	0.0046	0.0044	0.0046	0.0043	0.0046
	COV@5	0.0490	0.0449	0.0511	0.0462	0.0501	0.0448
	COV@10	0.0629	0.0559	0.0645	0.0573	0.0636	0.0561

Table 5.4 summarises the results of the different architectures and their re-ranked versions when evaluated on the RTL-NR dataset. We find that compared to the results on the MIND dataset, the drop in accuracy is more significant, with drops across all metrics between 4-9 percentage points and AUC dropping as much as 13-16 percentage points. This seems to suggest that the diverse re-ranking step has a significantly larger impact on the accuracy on the RTL-NR dataset than on the MIND dataset. However, we also find that the increase in ILD and RR-ILD is larger than on the MIND dataset, with 3-6 percentage points. Furthermore, where we witnessed a drop in COV using the diverse re-ranking on MIND, this is not the case on the RTL-NR dataset.

To investigate how much impact the personalized diverse re-ranking has on the list of recommendations, we use Rank-Biased Overlap (RBO). RBO is a similarity measure that calculates overlap between two ordered lists and ranges from 0 (no overlap) to 1 (perfect overlap) (Webber et al., 2010). Since we perform three runs for each of the models, both diversified and non-diversified, we can calculate the average RBO of the top-10 recommendations in the test set between these runs. We also calculate the average RBO between runs of the same model (either diversified or non-diversified) to provide a baseline for the average variance between runs. Here we omit the RBO of a run with itself since this will always be 1. We then calculate the average RBO between all diversified and non-diversified (denoted as CROSS) runs to measure the impact of diversification on the eventual recommendations. The results are summarized in Table 5.5. Overall, the overlap between runs of either the non-diversified or diversified models is much higher than the overlap between non-diversified and diversified models. An interesting finding here is that the level of variance differs significantly between the different models and datasets. For example, NAML shows a very high variance (low RBO) among runs of the same model on the RTL-NR dataset, while NRMS shows minimal variance (high RBO). These results show that adding the diverse re-ranking results in top-10 recommendations that are considerably different from what the default model would recommend.

Table 5.4: Performance of attention-based news recommenders, with and without re-ranking, evaluated on the RTL-NR test set. All numbers are means across three separate runs.

		PLM-NR		NRMS		NAML	
Type	Metric	Default	Re-ranked	Default	Re-ranked	Default	Re-ranked
Acc	AUC	0.7044	0.5678	0.6534	0.4935	0.6612	0.5334
	MRR	0.2906	0.2327	0.2142	0.1769	0.2200	0.1877
	nDCG@5	0.3025	0.2310	0.2041	0.1653	0.2125	0.1785
	nDCG@10	0.3730	0.2839	0.2882	0.2137	0.2971	0.2354
Div	ILD@5	0.4110	0.4603	0.4055	0.4624	0.4333	0.4582
	ILD@10	0.4207	0.4600	0.4179	0.4607	0.4347	0.4568
	RR-ILD@5	0.0041	0.0046	0.0040	0.0046	0.0043	0.0046
	RR-ILD@10	0.0042	0.0046	0.0042	0.0046	0.0043	0.0046
	COV@5	0.0916	0.0906	0.0479	0.0639	0.0831	0.0851
	COV@10	0.0935	0.0929	0.0706	0.0788	0.0905	0.0904

Table 5.5: Average Rank-Biased Overlap over three runs of the different architectures and datasets. Columns are Architecture (Archt.), Dataset (DS), Non-Diversified (NDIV) RBO, Diversified (DIV) RBO, and Cross RBO (CROSS).

Archt.	DS	NDIV	DIV	CROSS
PLM-NR	MIND	0.8353	0.8473	0.6884
	RTL-NR	0.7436	0.8161	0.5321
NRMS	MIND	0.8484	0.8465	0.6845
	RTL-NR	0.9896	0.9951	0.4736
NAML	MIND	0.8961	0.9215	0.6882
	RTL-NR	0.5598	0.6987	0.5022

We take a closer look at the RBO between the diversified and non-diversified models by considering the different groups of users. We bin the users into 10 equal groups based on their user history similarity and plot their mean RBO. The resulting Figure 5.1 shows these distributions. We can see a pattern across both datasets where the RBO decreases with increasing user history similarity. This is expected behaviour, as the diversity is assigned more weight when the user history similarity increases, and thus results in a more considerable change in the recommendations. Interestingly we see that for the MIND dataset, the RBO starts low for the group with the lowest user history similarity and then jumps up. Furthermore, we see a jump in RBO for the eight group, which breaks the pattern of decreasing RBO.

To answer the question of how personalized levels of diversity can contribute to better diversity in news recommendation, we can say that it can help increase the ILD and RR-ILD of the recommendations. However, we find that it can also hurt diversity in terms of COV, as is apparent in the case of the MIND dataset. Furthermore, using RBO, we have shown that the personalized diverse re-ranking significantly alters the top-10 recommendations produced by the model. It also demonstrates how the personalized diverse re-ranker has a bigger impact on users with undiverse reading habits (i.e. users with a high user history similarity).

5.2.1 Benefits of Personalized Levels of Diversity

In this subsection, we aim to answer the sub-research question; do personalized levels of diversity provide benefits over fixed levels?

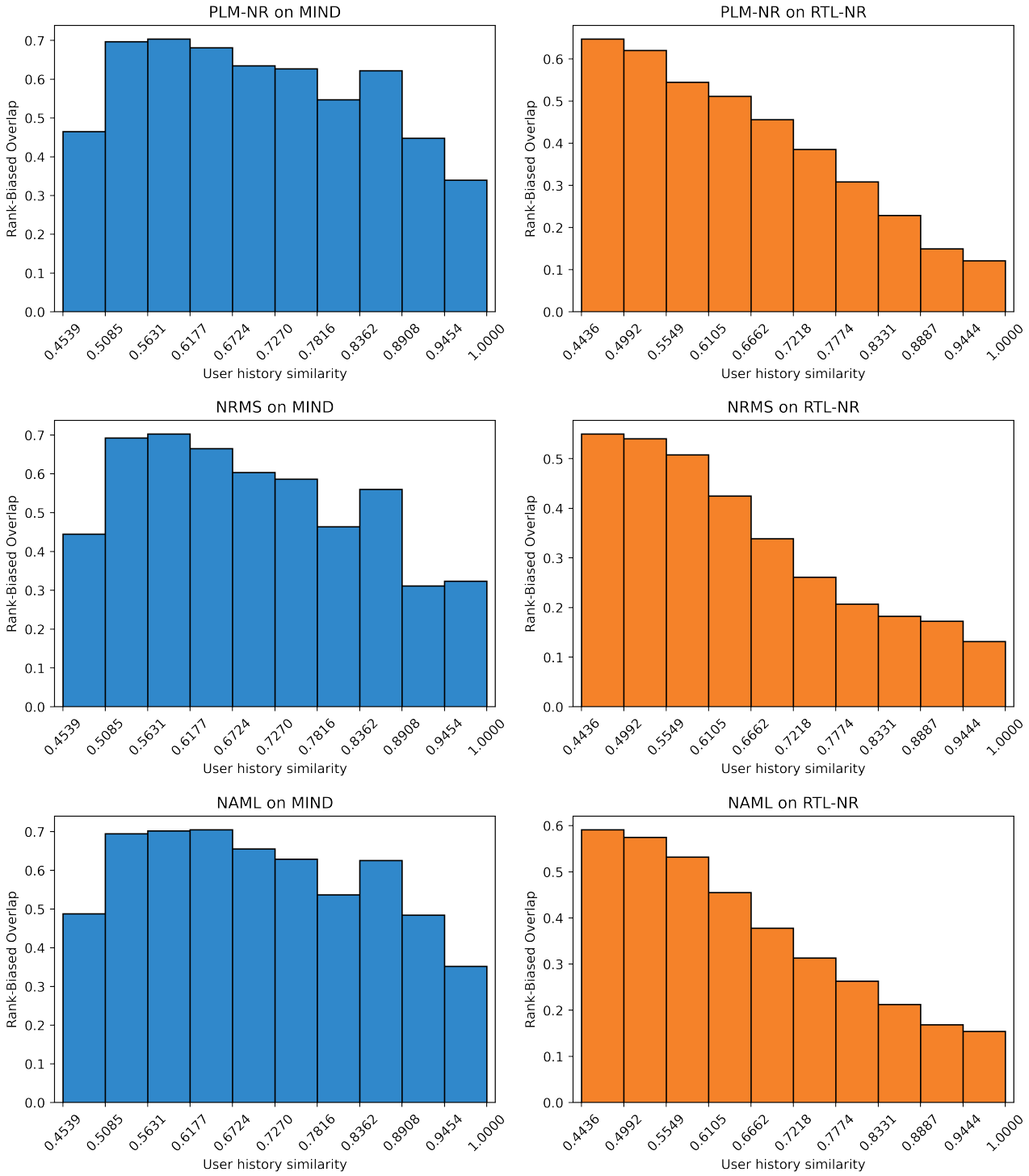

Figure 5.1: Average Rank-Biased Overlap over three runs per user group based on the user history similarity.

Table 5.6: Performance of different re-ranking functions. evaluated on the MIND test set using NAML. s=var denotes the model where s is determined per user, while the other functions use the given constant value. All numbers are means across three separate runs.

		MIND				RTL-NR			
Type	Metric	s=var	s=0.2	s=0.5	s=0.8	s=var	s=0.2	s=0.5	s=0.8
Acc	AUC	0.5911	0.6567	0.6061	0.5174	0.5334	0.6426	0.5216	0.4512
	MRR	0.3012	0.3193	0.3071	0.2629	0.1877	0.2258	0.1754	0.1529
	nDCG@5	0.3276	0.3531	0.3375	0.2692	0.1785	0.2226	0.1629	0.1323
	nDCG@10	0.3814	0.4173	0.3921	0.3250	0.2354	0.2938	0.2200	0.1811
Div	ILD@5	0.4572	0.4328	0.4502	0.4728	0.4582	0.4412	0.4589	0.4714
	ILD@10	0.4608	0.4417	0.4577	0.4664	0.4568	0.4417	0.4574	0.4650
	RR-ILD@5	0.0046	0.0043	0.0045	0.0047	0.0046	0.0044	0.0046	0.0047
	RR-ILD@10	0.0046	0.0044	0.0045	0.0047	0.0046	0.0044	0.0046	0.0046
	COV@5	0.0446	0.0496	0.0467	0.0379	0.0851	0.0831	0.0838	0.0744
	COV@10	0.0559	0.0630	0.0578	0.0500	0.0904	0.0897	0.0892	0.0864

We compared the personalized diverse re-ranker against three other versions where the user history similarity is given a fixed weight in $\{0.2, 0.5, 0.8\}$. The results are summarized in Table 5.6. For the MIND dataset, we find that the performance of the variable model is comparable to using a fixed weight of $s = 0.5$, with the variable model providing slightly better diversity and the fixed model providing slightly better accuracy. However, the variable model provides better accuracy on the RTL-NR dataset but slightly less diversity than $s = 0.5$. This indicates that personalised diversity's impact can vary across datasets. We also find that using a weight of $s = 0.2$ has little impact on the diversity while using a weight of $s = 0.8$ increases diversity significantly but also harms accuracy considerably. This holds for both datasets.

We take a closer look at these results by comparing the diversity and accuracy across different user groups. We group users based on their user history similarity and evaluate how each re-ranking function contributes to diversity and accuracy. For the diversity, we measure the gain in ILD @ 10 compared to the non-diversified model. For accuracy, we measure the decrease in Mean Reciprocal Rank (MRR) and normalized Discounted Cumulative Gain (nDCG) @ 10 compared to the non-diversified model. Figure 5.2 shows these distributions for the MIND dataset. We find that the personalized model approximately follows an increasing pattern in ILD, except for the last two groups where the ILD drops. Furthermore, the decrease in accuracy (MRR and nDCG @ 10) shows an increasing pattern. This means that the recommendations become less accurate when the user history similarity is high. This proves that our approach works, where users with a higher user history similarity get more diverse recommendations at the cost of accuracy. In contrast, users with a low user history similarity do not receive much less accurate recommendations. Interestingly, we find that the distributions of ILD also show an increasing pattern when using fixed values of s. This is especially apparent for the model that uses $s = 0.8$, where the distribution is very similar to the variable model. However, these models do not show a clear pattern in the decrease in MRR and nDCG @ 10. If we pick the $s = 0.8$ model as an example, we see that the most significant accuracy loss is actually for user groups with a lower user history similarity (between 0.5 and 0.67). For the variable model, these groups lose significantly less accuracy.

Figure 5.3 shows the same distributions for the RTL-NR dataset. For the personalized model, we see an even more evident increasing pattern in ILD compared to the MIND dataset. Furthermore, we also witness an increasing pattern in the decrease in accuracy. Like on the MIND dataset, this seems to suggest that the approach achieves its goal of providing users with

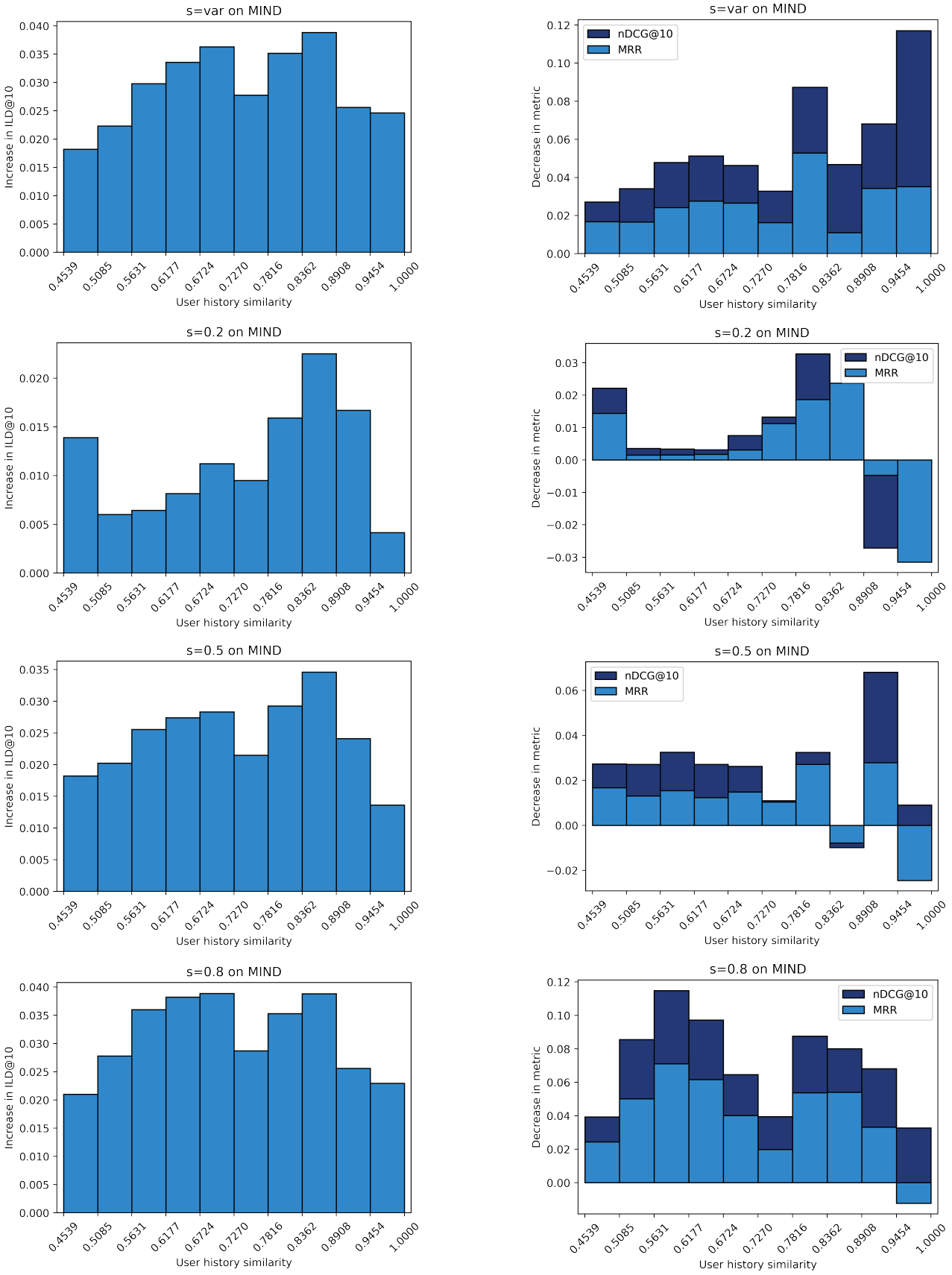

Figure 5.2: Increase in ILD and decrease in MRR and nDCG@10 of different values of s compared to the non-diversified model using NAML on the MIND dataset.

a higher user history similarity with more diverse recommendations at the cost of accuracy, while users with a low user history similarity do not receive much less accurate recommendations. Furthermore, like on the MIND dataset, we find that the distributions of ILD also show an increasing trend when using fixed values of s. However, in the case of the RTL-NR dataset, we find that the personalized model shows a stronger increasing trend. Another finding is that, unlike on the MIND dataset, the models with fixed values of s also show an increasing pattern in the decrease in accuracy measures. However, the personalized model appears to have a much bigger range than the other models in terms of both the increase in ILD and the decrease in MRR and nDCG @ 10. This suggests that the personalized model is better at diversifying based on the user history similarity and makes more difference across user groups.

To answer the question of what benefits personalized levels of diversity provide over fixed levels, we find that personalized levels are better able to provide diverse recommendations to users with undiverse reading habits while retaining accuracy for users with diverse reading habits compared to fixed levels. Even though fixed levels already exhibit an increasing pattern in ILD and on the RTL-NR dataset also in the decrease in MRR and nDCG @ 10, this pattern is much stronger when using the personalized model. We also find that the personalized model provides a more extensive range in these metrics than the fixed models.

5.3 Hyperparameter Search

This section presents the results of the hyperparameter search. The first subsection elaborates on the optimal parameters for the RTL-NR dataset. The second subsection covers the results of the search for the best text encoder. The third subsection discusses the results of the hyperparameter search for the best vector similarity measure. The fourth and last subsection covers the search for the best re-ranking function. The results for the other hyperparameters can be found in B. The optimal hyperparameters for the models can be found in Appendix C.

5.3.1 RTL-NR Training Parameters

As shown in Table 5.7, the different models behave differently with the different optimizers. Where NRMS performs best across the accuracy metrics with SGD with momentum, NAML performs best using Adam. However, both these models show that the ILD is better when using Adam instead of SGD. However, see the same pattern for PLM-NR. Interestingly, we see that the first epoch tends to be the best for the PLM-NR architecture while subsequent epochs degrade performance. This seems to suggest that the architecture on the RTL-NR dataset benefits significantly from the pre-training. However, when using AdamW as optimizer, we saw a gentle increase in performance up to the third epoch. We opted to train the model for a single epoch only due to the long training times (64 up to 68 hours per epoch) and the marginal performance gains (less than 1 percentage point across the accuracy measures between epochs 1 and 3 using AdamW). We pick the optimizers that yield the best accuracy metrics for each model. We use AdamW for PLM-NR, SGD with momentum for NRMS, and Adam for NAML.

Table 5.8 shows the result of the search for an optimal learning rate. We find that for both the NRMS and NAML architectures, the performance decreases when the learning rate increases or decreases to 1e-3 or 1e-5. This seems to suggest that 1e-4 is the optimal value for the learning rates. Interestingly, for the PLM-NR architecture, when we use a learning rate of 1e-3 and 1e-4, our model starts returning NaN loss values, which halts training. This seems to suggest that this learning rate is too high, and a lower learning rate is required for this model. On the other hand, with a learning rate of 1e-5, the learning seems more stable. Therefore, we use a learning rate of 1e-4 for NRMS and NAML and 1e-5 for PLM-NR.

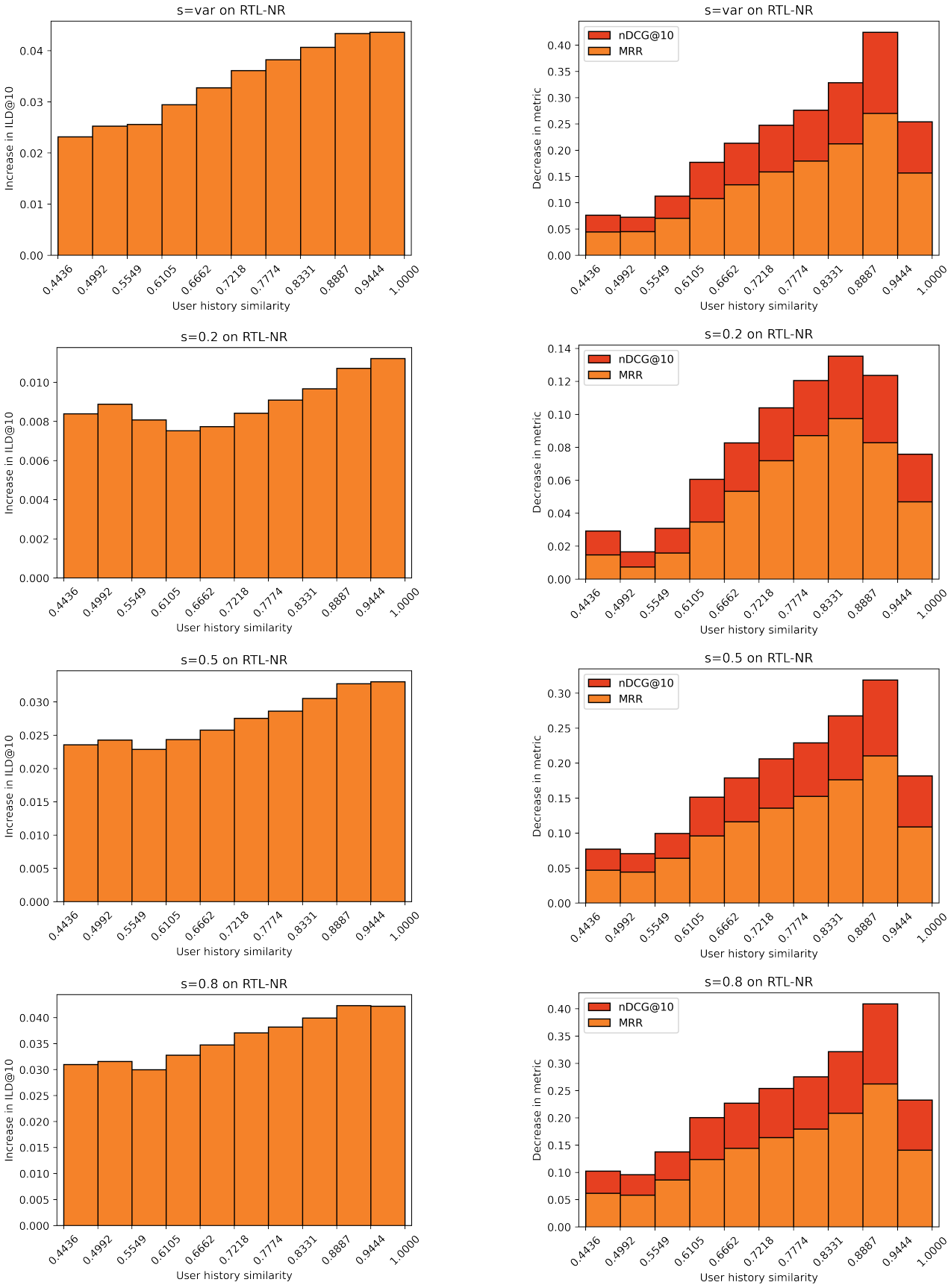

Figure 5.3: Increase in ILD and decrease in MRR and nDCG@10 of different values of s compared to the non-diversified model using NAML on the RTL-NR dataset.

Table 5.7: Performance of different optimizers evaluated on the RTL-NR validation set. Columns are Architecture (Archt.), Optimizer (OP), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 5 (nDCG@5), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Intra-List Distance @ 5 (ILD@5), and Intra-List Distance @ 10 (ILD@10).

Archt.	OP	NE	AUC	MRR	nDCG@5	nDCG@10	ILD@5	ILD@10
PLM-NR	SGD	1	**0.6889**	0.2589	0.2726	0.3401	0.3938	0.4117
	SGD+M	1	0.6580	0.2356	0.2385	0.3083	0.4105	0.4232
	Adam	1	0.6854	0.2717	0.2785	0.3430	0.4132	0.4239
	AdamW	1	0.6871	**0.2725**	**0.2792**	**0.3443**	**0.4138**	**0.4254**
NRMS	SGD	10	0.6111	0.2040	0.1934	0.2612	0.4161	0.4314
	SGD+M	4	**0.6695**	**0.2428**	**0.2436**	**0.3175**	0.4023	0.4182
	Adam	4	0.5793	0.1814	0.1757	0.2347	**0.4404**	**0.4430**
	AdamW	2	0.5900	0.1908	0.1834	0.2426	0.4303	0.4386
NAML	SGD	4	**0.6723**	0.2066	0.2075	0.2893	0.4124	0.4230
	SGD+M	3	0.6706	0.1903	0.1812	0.2732	0.4099	0.4210
	Adam	5	0.6590	**0.2279**	**0.2220**	**0.2944**	**0.4339**	0.4344
	AdamW	4	0.6306	0.1946	0.1834	0.2529	0.4333	**0.4352**

Table 5.8: Performance of different learning rates evaluated on the RTL-NR validation set. Columns are Architecture (Archt.), Learning Rate (LR), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 5 (nDCG@5), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Intra-List Distance @ 5 (ILD@5), and Intra-List Distance @ 10 (ILD@10).

Archt.	LR	NE	AUC	MRR	nDCG@5	nDCG@10	ILD@5	ILD@10
PLM-NR	1e-3	-	-	-	-	-	-	-
	1e-4	-	-	-	-	-	-	-
	1e-5	1	**0.6871**	**0.2725**	**0.2792**	**0.3443**	**0.4138**	**0.4254**
NRMS	1e-3	1	0.6597	0.2162	0.2174	0.2959	0.4092	0.4218
	1e-4	4	**0.6695**	**0.2428**	**0.2436**	**0.3175**	0.4023	0.4182
	1e-5	10	0.6111	0.2041	0.1935	0.2612	**0.4161**	**0.4314**
NAML	1e-3	1	0.6374	0.1683	0.1523	0.2434	0.4306	0.4327
	1e-4	5	**0.6590**	**0.2279**	**0.2220**	**0.2944**	**0.4339**	**0.4344**
	1e-5	4	0.6328	0.2042	0.1876	0.2574	0.4306	0.4339

Table 5.9: Performance of different negative ratios evaluated on the RTL-NR validation set. Columns are Architecture (Archt.), Negative Ratio (NR), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 5 (nDCG@5), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Intra-List Distance @ 5 (ILD@5), and Intra-List Distance @ 10 (ILD@10).

Archt.	NR	NE	AUC	MRR	nDCG@5	nDCG@10	ILD@5	ILD@10
PLM-NR	2	1	0.6753	0.2565	0.2607	0.3288	**0.4166**	**0.4255**
	4	1	0.6871	0.2725	0.2792	0.3443	0.4138	0.4254
	6	1	**0.6964**	**0.2743**	**0.2852**	**0.3515**	0.4143	0.4246
NRMS	2	5	**0.6701**	0.2413	0.2426	0.3175	0.4014	0.4175
	4	4	0.6695	0.2428	**0.2436**	0.3175	**0.4023**	**0.4182**
	6	4	0.6694	**0.2429**	0.2430	**0.3181**	**0.4023**	0.4175
NAML	2	2	0.6605	0.1892	0.1864	0.2737	0.4272	0.4290
	4	5	0.6590	**0.2279**	0.2220	0.2944	**0.4339**	**0.4344**
	6	3	**0.6734**	0.2214	**0.2264**	**0.2976**	0.4274	0.4310

Table 5.9 summarizes the search for the optimal negative ratio. We find that decreasing the negative ratio from 4 to 2 leads to significantly lower performance for NAML and PLM-NR, while there is no significant loss for the NRMS model. Furthermore, increasing the negative ratio from 4 to 6 only significantly improves the PLM-NR model. Therefore, we use a negative ratio of 4 for NAML and NRMS, while we use a negative ratio of 6 for PLM-NR.

5.3.2 Choice of Text Encoder

In Table 5.10, we can see that, for MIND, the Fastformer model outperforms BERT as text encoder while being significantly faster to train. However, we found that the training of the BERT model crashed during the second epoch due to the model returning NaN loss values. After contact with the original authors, we have been unable to resolve this issue. Furthermore, it appears that our models have an average of 1 to 2 percentage points lower performance across all metrics compared to the performance claimed by Wu et al. (2021a) and Wu et al. (2021b). We have also not been able to close this performance gap. However, these hyperparameter search results still validate the claim by Wu et al. (2021b), that Fastformer is a significantly faster model that can achieve performance as good as, or even better than, larger Transformer models, such as BERT. Therefore, we will use Fastformer as text encoder in the PLM-NR model throughout the experiments. However, when we look at the results for the RTL-NR dataset, we do not see the same trend. We find that BERT significantly outperforms Fastformer. However, this comes at the cost of significantly slower training times. We also found that when using BERT as text encoder, the first epoch is the best and performance decreases from there to a range of about 1 to 2 percentage points lower across all metrics. This seems to suggest that the model benefits significantly from BERT's pre-training. Therefore, we use Fastformer as text encoder for the MIND dataset, while we use BERT for the RTL-NR dataset throughout the rest of the experiments.

5.3.3 Vector Similarity Measure

The results in Table 5.11 show a clear distinction between the two vector similarity measures. We find that cosine similarity induces the highest ILD but at the cost of a significant drop in accuracy. On the other hand, euclidean distance introduces little ILD and maintains better accuracy metrics.

Table 5.10: Performance of different text encoders evaluated on the MIND and RTL-NR validation sets. Columns are Dataset (DS), Encoder, Learning Rate (LR), Batch Size (BS), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Average Training Epoch Time (ATET), and Average Evaluation Time (AET).

DS	Encoder	LR	BS	NE	AUC	MRR	nDCG@10	ATET(h)	AET(h)
MIND	BERT	1e-5	8	1	0.6634	0.3148	0.4113	127:35:40	0:07:15
	Fastformer	1e-4	64	5	**0.6848**	**0.3276**	**0.4288**	**9:21:39**	**0:06:53**
RTL-NR	BERT	1e-5	8	1	**0.6854**	**0.2717**	**0.3430**	67:18:34	0:26:19
	Fastformer	1e-5	64	5	0.6536	0.1954	0.2710	**4:52:58**	**0:25:43**

Table 5.11: Performance of diversification distance measures evaluated on the MIND and RTL-NR validation sets. Columns are Architecture (Archt.), Dataset (DS), Similarity Measure (SM), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Intra-List Distance @ 5 (ILD@5), and Intra-List Distance @ 10 (ILD@10).

Archt.	DS	SM	NE	AUC	MRR	nDCG@10	ILD@5	ILD@10
PLM-NR	MIND	CS	2	0.6001	0.3057	0.3875	**0.4535**	**0.4593**
		ED	4	**0.6674**	**0.3240**	**0.4233**	0.4286	0.4406
	RTL-NR	CS	1	0.5491	0.1987	0.2404	**0.4561**	**0.4595**
		ED	1	**0.6654**	**0.2614**	**0.3313**	0.4243	0.4342
NRMS	MIND	CS	6	0.6147	0.3145	0.3975	**0.4560**	**0.4601**
		ED	6	**0.6774**	**0.3296**	**0.4312**	0.4311	0.4407
	RTL-NR	CS	8	0.5207	0.1645	0.2031	**0.4640**	**0.4616**
		ED	5	**0.6391**	**0.2291**	**0.2918**	0.4191	0.4360
NAML	MIND	CS	5	0.5971	0.3052	0.3849	**0.4573**	**0.4609**
		ED	5	**0.6599**	**0.3218**	**0.4206**	0.4318	0.4408
	RTL-NR	CS	2	0.5123	0.1651	0.1981	**0.4596**	**0.4600**
		ED	3	**0.6432**	**0.1973**	**0.2685**	0.4374	0.4388

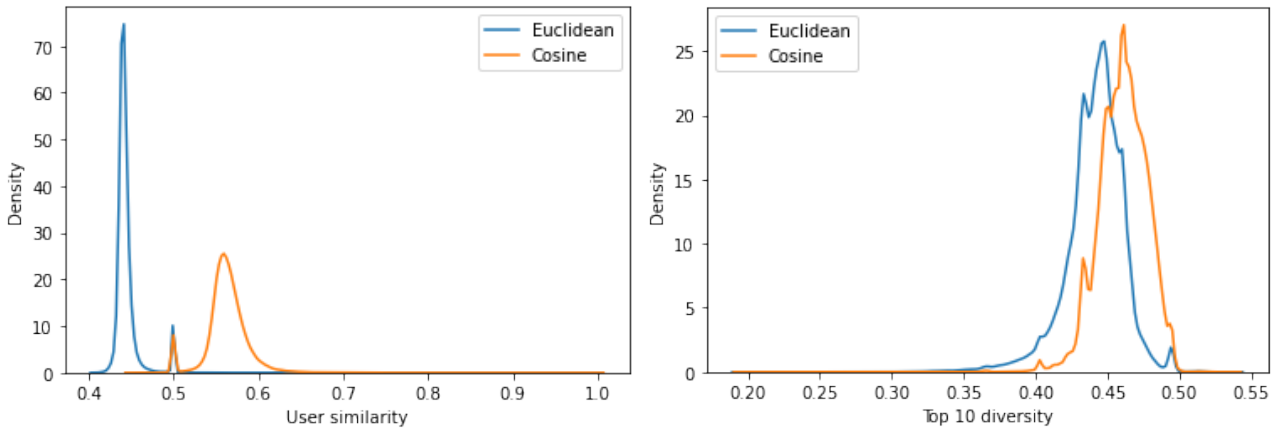

Figure 5.4: Comparison of the user similarity and top-10 diversity between euclidean distance and cosine similarity as vector similarity measures on the MIND dataset, using the NRMS model.

To explain the difference between the two similarity measures, we looked at the distributions of both Euclidean distance and cosine similarity for the user similarity of the MIND dataset. The left graph in Figure 5.4 clearly shows that overall cosine similarity returns higher values for the user similarity than Euclidean distance. The average user similarity for euclidean distance is 0.445, while this is 0.563 for cosine similarity. In practice, this means that, on average, the majority of the diverse re-ranked score is composed of the relevancy score for euclidean distance and the diversity for cosine similarity. We also see in the right graph of Figure 5.4 that this, in turn, also leads to lower top-10 diversity for Euclidean distance when compared to cosine similarity. These results are both measured using cosine similarity, such that they are directly comparable. Since cosine similarity appears to be better at driving diversity, we use this as our vector similarity measure throughout the rest of our experiments.

5.3.4 Diverse Re-ranking Function

Table 5.12 shows the results of the search for the best diverse re-ranking function. On the MIND dataset, we can see a clear pattern across all models, where the normalized re-ranker retains the best accuracy and the naive re-ranker scores the best (or second best in the case of PLM-NR) on the diversity metrics. For the RTL-NR dataset, the story is less clear. We find that the conservative bounded re-ranker and normalized re-ranker interchangeably perform the best in terms of accuracy, while the diverse bounded re-ranker and naive re-ranker interchangeably perform the best in terms of diversity. Since we find that on the MIND dataset, the naive re-ranker performs the best in diversity and achieves near or the best diversity on the RTL-NR dataset, we use this as our diverse re-ranking function throughout the experiments.

Table 5.12: Performance of diverse re-ranking functions evaluated on the MIND and RTL-NR validation sets. Columns are Architecture (Archt.), Dataset (DS), Re-ranking Function (RF), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Intra-List Distance @ 5 (ILD@5), and Intra-List Distance @ 10 (ILD@10).

Archt.	DS	RF	NE	AUC	MRR	nDCG@10	ILD@5	ILD@10
PLM-NR	MIND	NA	2	0.6001	0.3057	0.3875	0.4535	0.4593
		BO_C	3	0.6118	0.3126	0.3962	0.4521	0.4588
		BO_D	3	0.6005	0.3087	0.3901	**0.4549**	**0.4598**
		NO	3	**0.6339**	**0.3158**	**0.4078**	0.4408	0.4519
	RTL-NR	NA	1	0.5491	0.1987	0.2404	**0.4561**	**0.4595**
		BO_C	1	**0.5875**	0.2124	**0.2654**	0.4491	0.4565
		BO_D	1	0.5658	0.2051	0.2507	0.4551	0.4590
		NO	1	0.5690	**0.2146**	0.2618	0.4503	0.4563
NRMS	MIND	NA	6	0.6147	0.3145	0.3975	**0.4560**	**0.4601**
		BO_C	5	0.6180	0.3142	0.3990	0.4544	0.4594
		BO_D	5	0.6161	0.3164	0.3997	0.4555	0.4600
		NO	4	**0.6430**	**0.3220**	**0.4149**	0.4438	0.4531
	RTL-NR	NA	8	0.5207	0.1645	0.2031	0.4640	0.4616
		BO_C	8	**0.5449**	**0.1810**	**0.2248**	0.4612	0.4599
		BO_D	8	0.5189	0.1631	0.2010	**0.4645**	**0.4620**
		NO	7	0.5315	0.1716	0.2124	0.4586	0.4585
NAML	MIND	NA	5	0.5971	0.3052	0.3849	**0.4573**	**0.4609**
		BO_C	4	0.5934	0.3023	0.3827	0.4551	0.4601
		BO_D	4	0.5912	0.3017	0.3810	0.4571	**0.4609**
		NO	3	**0.6229**	**0.3104**	**0.4000**	0.4443	0.4536
	RTL-NR	NA	2	0.5123	0.1651	0.1981	0.4596	0.4600
		BO_C	3	**0.5170**	0.1667	0.2029	0.4600	0.4595
		BO_D	3	0.5029	0.1494	0.1830	**0.4649**	**0.4625**
		NO	4	0.5111	**0.1679**	**0.2032**	0.4591	0.4590

Chapter 6

Conclusions

In this chapter, we discuss the conclusions we can draw from the results of our experiments. The first three subsections summarize the conclusions from each of the corresponding experiments. In the fourth subsection, we cover the limitations of our work. In the fifth and final subsection, we provide some interesting ideas and directions for future research.

6.1 Diversity of Attention-based News Recommenders

In this section, we conclude our findings for the first research question; how diverse are state-of-the-art attention-based news recommenders?

Our experiments found that the attention-based models (PLM-NR, NRMS, and NAML) are not more diverse than the previous state-of-the-art model, which is LSTUR. We also find no significant differences between the three attention-based models regarding diversity. Whereas NAML is significantly more diverse than the other models on the RTL - News Recommendation (RTL-NR) dataset, this difference is not apparent on the Microsoft News Dataset (MIND) dataset. Thus, we can conclude that the state-of-the-art attention-based news recommenders are not more diverse than the previous state-of-the-art, and there are no significant differences in diversity among the models.

6.2 Impact of Personalized Diverse Re-ranking on News Recommendation

This section concludes our findings for the second research question; how can personalized levels of diversity contribute to better diversity in news recommendation?

The experiments have shown that personalized diverse re-ranking can help increase the recommendations' diversity in terms of Intra-List Distance (ILD) and Rank and Relevance sensitive Intra-List Distance (RR-ILD). However, the experiments on the MIND dataset showed that the diverse re-ranking could also harm diversity in terms of Coverage (COV). On the RTL-NR dataset, we did not see a significant change in the COV, which indicates that it is dataset dependent on whether diverse re-ranking hurts COV. Our experiments using Rank-Biased Overlap (RBO) have shown that the top-10 recommendations change significantly after applying the personalized diverse re-ranking. Furthermore, we also find that the personalized diverse re-ranker works as intended and has a more significant impact on the recommendations shown to users with undiverse reading habits (i.e. users with a high user history similarity) than on the recommendations shown to users with diverse reading habits (i.e. those who have a low user history similarity).

6.2.1 Benefits of Personalized Levels of Diversity

This subsection concludes the results of the experiments for the sub-research question; do personalized levels of diversity provide benefits over fixed levels?

The experiments have shown that personalized levels can better provide diverse recommendations to users with undiverse reading habits, while retaining accuracy for users with diverse reading habits compared to fixed levels. Even though fixed levels of diversity already exhibit an increasing pattern in ILD and on the RTL-NR dataset also in the decrease in Mean Reciprocal Rank (MRR) and normalized Discounted Cumulative Gain (nDCG) @ 10, this pattern is much stronger when using the personalized model. The results on the MIND dataset also show that the personalized model is better able to sacrifice accuracy for higher diversity for users with undiverse reading habits (i.e. a high user history similarity). In contrast, the models with fixed values appear to lose accuracy more uniformly across all users. Thus, using personalized levels of diversity, we are better able to provide diverse recommendations to users with undiverse reading habits while preserving accuracy for users who already consume diverse news.

6.3 Limitations

One limitation of this work is that the proposed diverse re-ranking occurs after the accuracy-optimized recommendation. Even though this makes the approach versatile, such that it can be applied to any recommender model, it also introduces additional computation and latency. We are well aware of this limitation, but this research was intended to bring up a personalized approach to diversity where users with different reading habits are exposed to different levels of diversity within their recommendations.

A limitation of our Personalized Diverse Recommendation (PD-Rec) approach is that the user history similarity (s) might depend on the history length of a user. One can argue that it is more likely that 3 articles are similar to each other than that 15 articles are similar to each other. Our approach does not consider the fact that more articles in the user history will likely lead to less user history similarity.

Another limitation is that this study only considers offline experiments on two real-life datasets. This means we do not know the impact personalized diverse re-ranking has on user click behaviour. We note that this is a severe limitation of our work, as offline metrics do not tell the whole story and cannot reflect user behaviour and preferences.

Furthermore, we cannot make the RTL-NR dataset publicly available due to privacy legislation and concerns. We note that this is a severe limitation to the reproducibility of this paper.

6.4 Future Work

We hope this research leads to more research into personalized levels of diversity in news recommendation. We think this is an exciting way to view the diversity problem in news recommendation. Future work could, for example, focus on implementing user similarity directly into a news recommender and optimize towards both diversity and accuracy per user.

Another important aspect for future research would be to investigate the impact of diversification on online user click behaviour and user preferences. This could be done using A/B testing and user studies on news websites.

We also think there are possibilities to investigate the use of personalized levels of diversity with other notions of diversity. For example, for news aggregators, it might be very relevant to ensure diversity in terms of political views or minority representation for users that read articles with only certain political views or are from a particular ethnic background.

Furthermore, it could be interesting to investigate how users actually perceive diversified lists of news recommendations. This could be done using a user study. Quantitative results do not tell the whole story in this case. People could perceive diversified lists as higher quality and prefer them over lists optimized for accuracy. Thus, future research could perform qualitative experiments like user studies to better understand the users' preferences.

Appendix A

Dataset Analysis

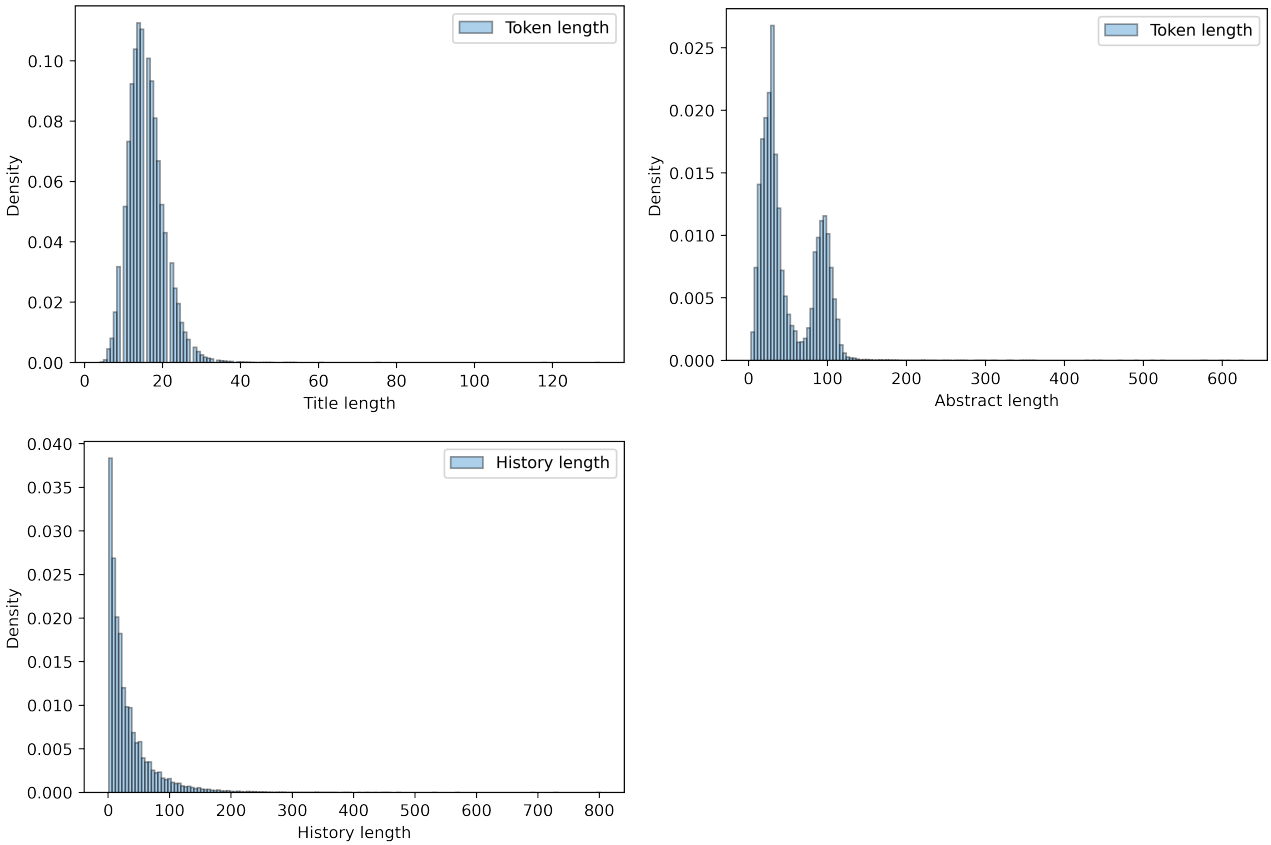

Figure A.1: MIND length analysis. Title and abstract length are in token length, as determined by a BERT tokenizer. History lengths are in number of articles.

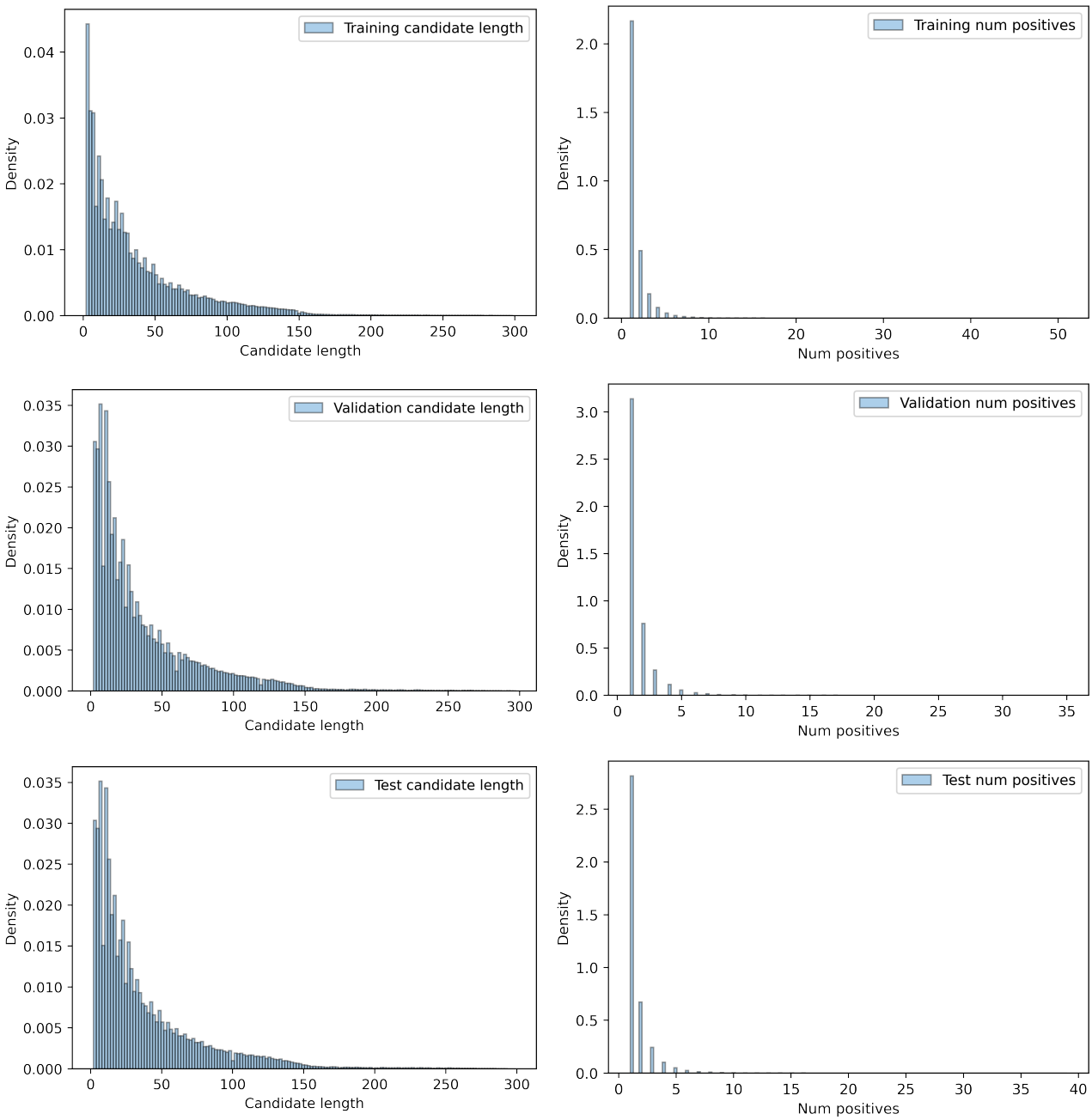

Figure A.2: MIND candidate and positives analysis.

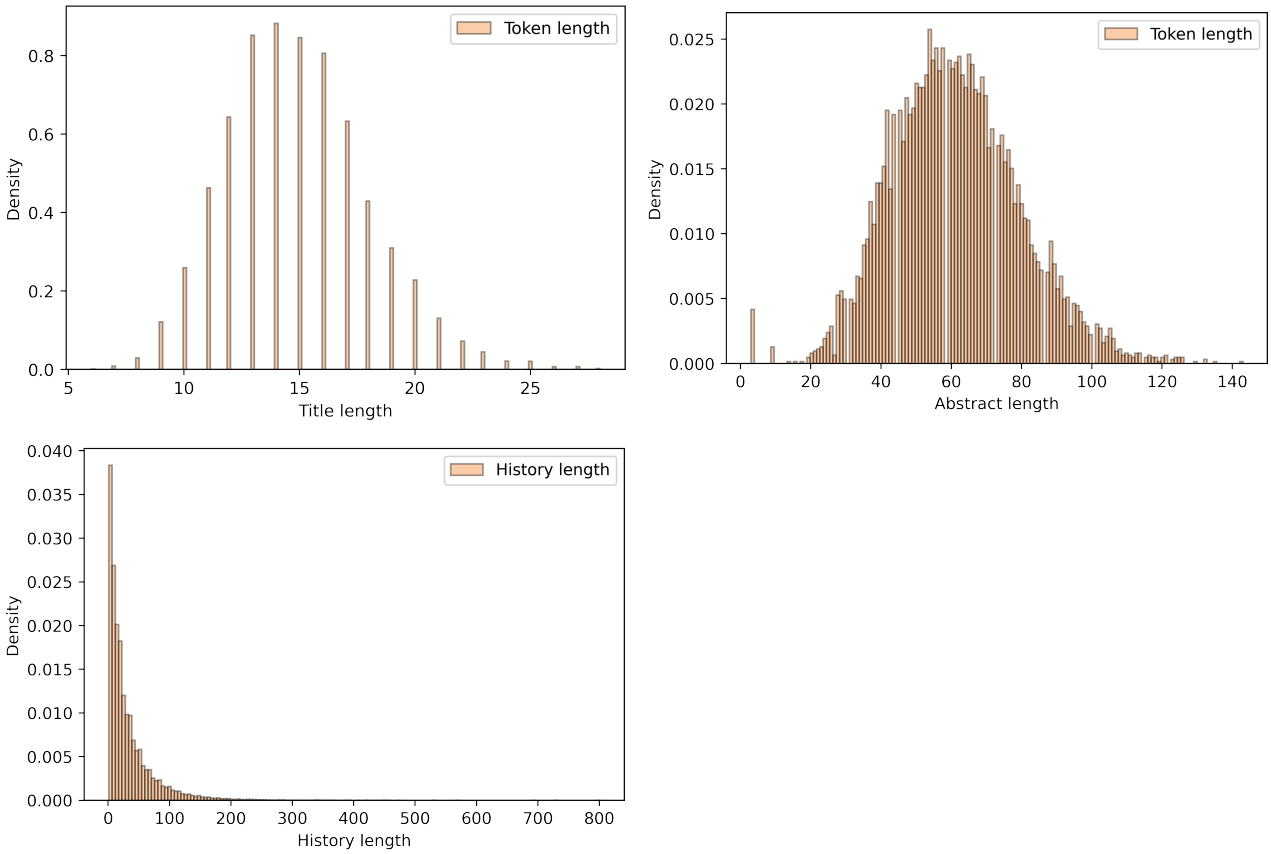

Figure A.3: RTL-NR length analysis. Title and abstract length are in token length, as determined by a BERT tokenizer. History lengths are in number of articles.

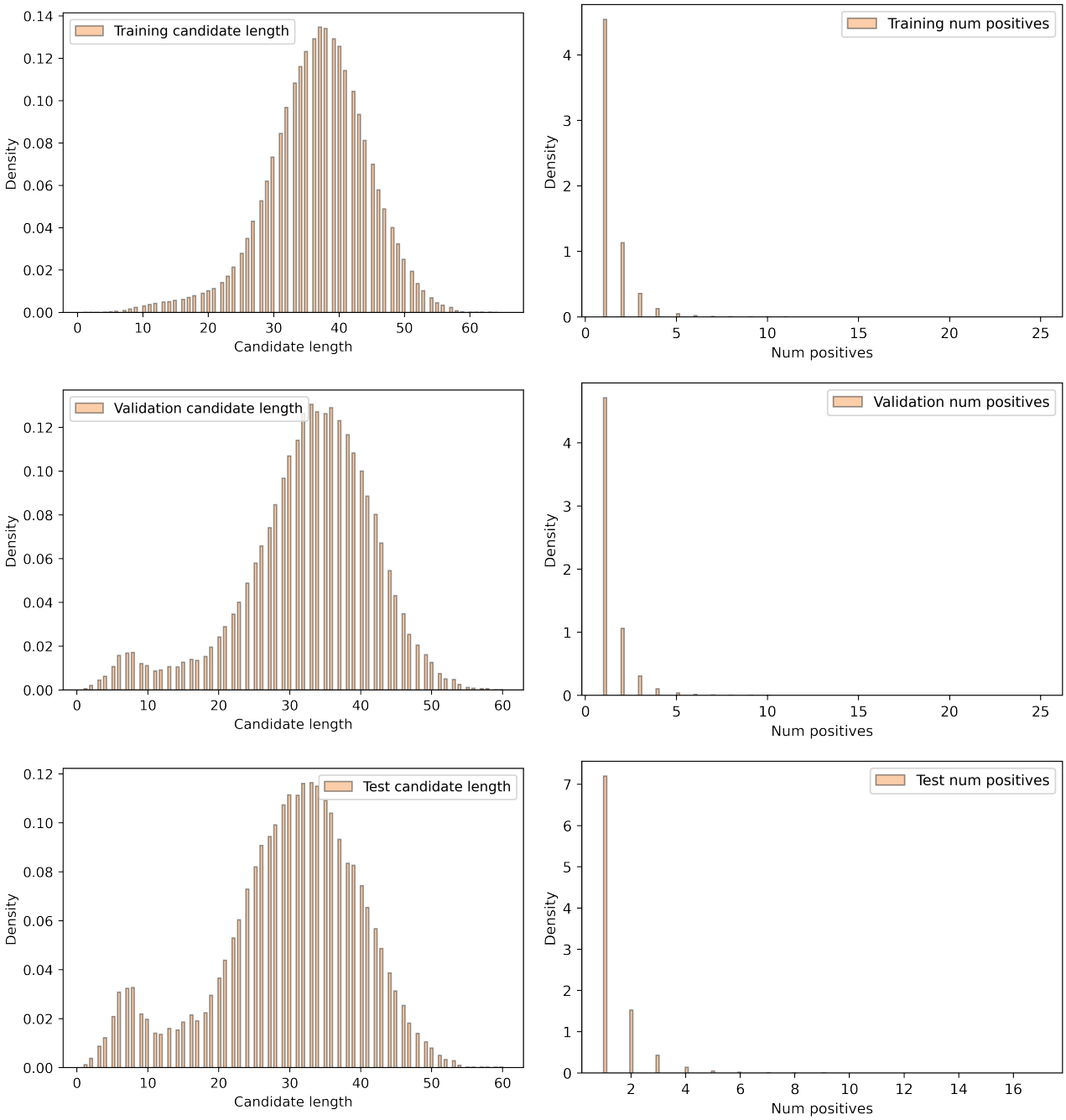

Figure A.4: RTL-NR candidate and positives analysis.

Appendix B

Data Lengths Hyperparameter Search Results

B.1 Text Length

As Table B.1 shows, increasing the title and abstract text length does not yield any significant increases in the accuracy metrics. However, it does yield very significant increasing in training times, especially for the PLM-NR and NRMS models that have to use smaller batch sizes to accommodate the increase in text length. We also witness small increases in evaluation times when using the longer text lengths. Since the longer texts do not result in any significant performance gains, but do lead to significantly longer training times and slightly longer evaluation times, we opt to use the default short text length of 20 and 50 tokens for the title and abstract respectively.

B.2 History Length

In Table B.2, we can deduce no clear pattern on the MIND dataset. We find that increasing the history length for MIND does not yield better performance for all models, while lowering the history length does yield a decrease in performance across all models. This seems to suggest that a history length of 50 is ideal for the MIND dataset, since it provides the optimal balance between training times and performance. We find that there are only marginal differences in evaluation times across the different history lengths. However, for the RTL-NR dataset we surprisingly find that decreasing the history length of 50 to 30 yields slightly better results for the NRMS and NAML models. Meanwhile, increasing the history length from 50 to 70 yields better results on the PLM-NR model. However, these increases are not very significant and vary per model. Furthermore, the significant increase in training times when increasing the history length is not offset by the small gains in performance. Thus, for both datasets, we have decided to use a history length of 50 articles throughout our experiments.

Table B.1: Performance of different text lengths evaluated on the MIND and RTL-NR validation sets. Columns are Architecture (Archt.), Dataset (DS), Text Length (TL), Batch Size (BS), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Average Training Epoch Time (ATET), and Average Evaluation Time (AET).

Archt.	DS	TL	BS	NE	AUC	MRR	nDCG@10	ATET(h)	AET(h)
PLM-NR	MIND	Short	64	5	**0.6848**	**0.3276**	**0.4288**	**9:21:39**	**0:06:53**
		Long	16	1	0.6671	0.3186	0.4184	21:28:21	0:06:55
	RTL-NR	Short	8	1	**0.6964**	0.2743	0.3516	**68:18:10**	**0:26:27**
		Long	4	1	0.6838	**0.2811**	**0.3547**	159:50:06	0:27:16
NRMS	MIND	Short	64	6	**0.6926**	0.3317	0.4346	**5:20:18**	**0:07:05**
		Long	16	3	0.6920	**0.3385**	**0.4392**	14:38:55	0:07:19
	RTL-NR	Short	64	4	**0.6696**	**0.2428**	**0.3178**	**3:24:11**	**0:26:34**
		Long	16	3	0.6656	0.2360	0.3078	6:53:06	0:27:31
NAML	MIND	Short	256	3	0.6726	0.3198	0.4204	**0:29:00**	**0:07:02**
		Long	256	3	**0.6734**	**0.3206**	**0.4207**	0:31:52	0:07:41
	RTL-NR	Short	256	1	0.6623	0.2180	0.2786	**0:14:12**	**0:28:52**
		Long	256	3	**0.6667**	**0.2205**	**0.2961**	0:15:05	0:29:01

Table B.2: Performance of different history lengths evaluated on the MIND and RTL-NR validation sets. Columns are Architecture (Archt.), Dataset (DS), History Length (HL), Batch Size (BS), Number of Epochs (NE), Area Under Curve (AUC), Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain @ 10 (nDCG@10), Average Training Epoch Time (ATET), and Average Evaluation Time (AET).

Archt.	DS	HL	BS	NE	AUC	MRR	nDCG@10	ATET(h)	AET(h)
PLM-NR	MIND	70	32	2	0.6835	0.3289	**0.4295**	12:54:50	0:06:53
		50	64	5	**0.6848**	0.3276	0.4288	9:21:39	0:06:53
		30	64	4	0.6834	**0.3291**	0.4289	**6:18:13**	**0:06:38**
	RTL-NR	70	4	1	0.6923	**0.2837**	**0.3562**	91:53:10	0:26:58
		50	8	1	**0.6964**	0.2743	0.3516	68:18:10	0:26:27
		30	8	1	0.6704	0.2593	0.3292	**45:18:33**	**0:25:52**
NRMS	MIND	70	32	3	0.6881	**0.3317**	0.4335	7:10:09	0:07:04
		50	64	6	**0.6926**	**0.3317**	**0.4346**	05:20:18	0:07:05
		30	128	6	0.6901	0.3297	0.4324	**03:22:05**	**0:06:56**
	RTL-NR	70	32	3	0.6724	0.2407	0.3175	4:46:25	0:27:40
		50	64	4	0.6696	**0.2428**	0.3178	3:24:11	0:26:34
		30	128	7	**0.6699**	**0.2428**	**0.3196**	**2:08:55**	**0:26:18**
NAML	MIND	70	256	4	**0.6758**	**0.3216**	**0.4223**	0:35:51	0:07:51
		50	256	3	0.6726	0.3198	0.4204	0:29:00	**0:07:02**
		30	256	4	0.6724	0.3211	0.4213	**0:20:53**	**0:07:02**
	RTL-NR	70	256	3	0.6567	0.2081	0.2799	0:17:50	**0:27:28**
		50	256	1	0.6623	0.2180	0.2786	0:14:12	0:28:52
		30	256	3	**0.6668**	**0.2220**	**0.2947**	**0:10:02**	0:28:12

Appendix C

Model Hyperparameters

Table C.1: Hyperparameters used for the different models on MIND.

Parameter	PLM-NR	NRMS	NAML
Batch Size	64 (Fastformer) / 8 (BERT)	64	256
Negative Ratio	4	4	4
Filter Number	0	0	0
News Attributes	T, A, C, S	T, A, C, S	T, A, C, S
Optimizer	Adam	Adam	Adam
Learning Rate	1e-4 (Fastformer) / 1e-5 (BERT)	1e-4	1e-4
Max Epochs	10	10	10
Num Words Title	20	20	20
Num Words Abstract	50	50	50
User Log Length	50	50	50
Word Embedding Dim	256 (Fastformer) / 768 (BERT)	300	300
Num Last Layers Unfrozen	All (Fastformer) / 2 (BERT)	-	-
News Dim	64	400	400
News Query Vector Dim	200	200	200
User Query Vector Dim	200	200	200
Num Attention Heads	20	20	20
User Log Mask	True	True	True
Dropout Rate	0.2	0.2	0.2
Lowercase	True	True	True

Table C.2: Hyperparameters used for the different models on RTL-NR.

Parameter	PLM-NR	NRMS	NAML
Batch Size	64 (Fastformer) / 8 (BERT)	64	256
Negative Ratio	6	4	4
Filter Number	0	0	0
News Attributes	T, A, C	T, A, C	T, A, C
Optimizer	AdamW	SGD+M	Adam
Learning Rate	1e-5 (Fastformer) / 1e-5 (BERT)	1e-4	1e-4
Max Epochs	1	10	10
Num Words Title	20	20	20
Num Words Abstract	50	50	50
User Log Length	50	50	50
Word Embedding Dim	256 (Fastformer) / 768 (BERT)	300	300
Num Last Layers Unfrozen	All (Fastformer) / 2 (BERT)	-	-
News Dim	64	400	400
News Query Vector Dim	200	200	200
User Query Vector Dim	200	200	200
Num Attention Heads	20	20	20
User Log Mask	True	True	True
Dropout Rate	0.2	0.2	0.2
Lowercase	True	True	True

References

G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, Florence, Italy. Association for Computational Linguistics.

Yonata Andrelo Asikin and Wolfgang Wörndl. 2014. Stories around you: Location-based serendipitous recommendation of news articles. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014*, volume 1181 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ashok Basnet and Arun K. Timalsina. 2018. Improving nepali news recommendation using classification based on lstm recurrent neural networks. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, pages 138–142.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Andrew P. Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Sanxing Cao, Nan Yang, and Zhengzheng Liu. 2017. Online news recommender based on stacked auto-encoder. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 721–726.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 659–666, New York, NY, USA. Association for Computing Machinery.

Nick Craswell. 2009. *Mean Reciprocal Rank*, pages 1703–1703. Springer US, Boston, MA.

Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 153–162, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Evgeny Frolov and Ivan Oseledets. 2018. Tensor methods and recommender systems.

Florent Garcin and Boi Faltings. 2013. Pen recsys: A personalized news recommender systems framework. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 469–470, New York, NY, USA. Association for Computing Machinery.

Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and time aware neighborhood for session-based recommendations: Stan. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1069–1072, New York, NY, USA. Association for Computing Machinery.

Alireza Gharahighehi and Celine Vens. 2021. Diversification in session-based news recommender systems. *CoRR*, abs/2102.03265.

Wanrong Gu, Shoubin Dong, Zhizhao Zeng, and Jinchao He. 2014. An effective news recommendation method for microblog user. *TheScientificWorldJournal*, 2014:907515.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1725–1731. AAAI Press.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 173–182, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, page 1–9, New York, NY, USA. Association for Computing Machinery.

Lucien Heitz, Juliane A. Lischka, Alena Birrer, Bibek Paudel, Suzanne Tolmeijer, Laura Laugwitz, and Abraham Bernstein. 2022. Benefits of diverse news recommendations for democracy: A user study. *Digital Journalism*, 0(0):1–21.

Natali Helberger. 2019. On the democratic role of news recommenders. *Digital Journalism*, 7(8):993–1012.

Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 306–310, New York, NY, USA. Association for Computing Machinery.

Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar Shalom, Haggai Roitman, and Bracha Shapira. 2016. Recsys'16 workshop on deep learning for recommender systems (dlrs). pages 415–416.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Know.-Based Syst.*, 111(C):180–192.

Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 210–217, New York, NY, USA. Association for Computing Machinery.

Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications*, 41(7):3168–3177.

Yan Li, Dhruv Choudhary, Xiaohan Wei, Baichuan Yuan, Bhargav Bhushanam, Tuo Zhao, and Guanghui Lan. 2021. Frequency-aware SGD for efficient embedding learning with provable benefits. *CoRR*, abs/2110.04844.

G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

Felicia Loecherbach, Judith Moeller, Damian Trilling, and Wouter van Atteveldt. 2020. The unified framework of media diversity: A systematic literature review. *Digital Journalism*, 8(5):605–642.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Feng Lu, Anca Dumitrache, and David Graus. 2020. Beyond optimizing for clicks: Incorporating editorial values in news recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2020, Genoa, Italy, July 12-18, 2020*, pages 145–153. ACM.

Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019a. Empirical analysis of session-based recommendation algorithms. *CoRR*, abs/1910.12781.

Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019b. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 462–466, New York, NY, USA. Association for Computing Machinery.

Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting online performance of news recommender systems through richer evaluation metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 179–186, New York, NY, USA. Association for Computing Machinery.

Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, page 1097–1101, New York, NY, USA. Association for Computing Machinery.

Nic Newman, Richard Fletcher, Anne Schulz, Sigme Andı, Craig T. Robertson, and Rasmus Kleis Nielsen. 2021. Reuters institute digital news report 2021 10th edition.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1933–1942, New York, NY, USA. Association for Computing Machinery.

Javier Parapar and Filip Radlinski. 2021. *Towards Unified Metrics for Accuracy and Diversity for Recommender Systems*, page 75–84. Association for Computing Machinery, New York, NY, USA.

Eli Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group , The.

Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2255–2258, New York, NY, USA. Association for Computing Machinery.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. PP-rec: News recommendation with personalized user interest and time-aware news popularity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5457–5467, Online. Association for Computational Linguistics.

Shaina Raza. 2021. A news recommender system considering temporal dynamics and diversity.

Shaina Raza and Chen Ding. 2020. A regularized model to trade-off between accuracy and diversity in a news recommender system. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 551–560.

Shaina Raza and Chen Ding. 2021a. Deep dynamic neural network to trade-off between accuracy and diversity in a news recommender system.

Shaina Raza and Chen Ding. 2021b. News recommender system: A review of recent progress, challenges, and opportunities.

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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

K. G. Saranya and G. Sudha Sadasivam. 2017. Personalized news article recommendation with novelty using collaborative filtering based rough set theory. *Mob. Netw. Appl.*, 22(4):719–729.

Samuel L. Smith, Erich Elsen, and Soham De. 2020. On the generalization benefit of noise in stochastic gradient descent. *CoRR*, abs/2006.15081.

Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 909–912, New York, NY, USA. Association for Computing Machinery.

Gabriel de Souza Pereira Moreira. 2018. Chameleon: A deep learning meta-architecture for news recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 578–583, New York, NY, USA. Association for Computing Machinery.

Cass R. Sunstein. 2001. *Republic.Com*. Princeton University Press, USA.

Saúl Vargas. 2014. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *Proceedings of the 37th International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '14, page 1281, New York, NY, USA. Association for Computing Machinery.

Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 109–116, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. 2021. *Recommenders with a Mission: Assessing Diversity in News Recommendations*, page 173–183. Association for Computing Machinery, New York, NY, USA.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1835–1844, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Shaoqing Wang, Benyou Zou, Cuiping Li, Kankan Zhao, Qiang Liu, and Hong Chen. 2015. Crown: A context-aware recommender for web news. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1420–1423.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. 2013. A theoretical analysis of NDCG type ranking measures. *CoRR*, abs/1304.6480.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 3863–3869. AAAI Press.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 2576–2584, New York, NY, USA. Association for Computing Machinery.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. 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)*, pages 6389–6394, Hong Kong, China. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020a. SentiRec: Sentiment diversity-aware neural news recommendation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 44–53, Suzhou, China. Association for Computational Linguistics.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021a. Empowering news recommendation with pre-trained language models.

Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021b. Fastformer: Additive attention can be all you need.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020b. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, page 153–162, New York, NY, USA. Association for Computing Machinery.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 167–176, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Hua Zheng, Dong Wang, Qi Zhang, Hang Li, and Tinghao Yang. 2010. Do clicks measure recommendation relevancy? an empirical user study. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 249–252, New York, NY, USA. Association for Computing Machinery.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5973–5980.

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA. Association for Computing Machinery.