University of Zurich^UZH

# Diverse Political News Recommendations

## Design and Implementation of an Algorithm for Diverse Political News Recommendations

Thesis August 21, 2019

**Lucien Heitz**
of Bonstetten ZH, Switzerland

Student-ID: 12-737-979
lucien.heitz@uzh.ch

Advisor: **Suzanne Tolmeijer**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

# Acknowledgements

I would like to express my deep gratitude to Professor Abraham Bernstein for giving me the opportunity to work on this research topic and the valuable and constrictive suggestions he has given me. I would like to offer my special thanks to my advisor Suzanne Tolmeijer for her guidance, great kindness and invaluable support. Assistance provided by Dr. Bibek Paudel was greatly appreciated and I benefited from his expertise on recommender systems. My grateful thanks are extended to Professor Natali Helberger for her help with the political aspects of this thesis. Assistance provided by Dr. Juliane Lischka and Alena Birrer was greatly appreciated and their help was vital for creating the surveys featured here. My thanks are extended to *Blick*, *Neue Zürcher Zeitung* and *Tages Anzeiger* for taking part in the research experiment and granting access to their news archives. I would also like to thank Markus Göckeritz, Christof Schwizer and Luca Weibel who all codeveloped the DDIS News App and provided support for the integration of the algorithm implemented here.

# Zusammenfassung

Viele Personen lesen heutzutage Nachrichten im Internet. Die angebotenen Artikel sind dabei oftmals personalisiert und auf die Interessen der einzelnen Personen abgestimmt. Hierfür werden sogenannte Empfehlungsdienste eingesetzt. Eine primäre Fokussierung dieser Dienste auf die Interessen der einzelnen Personen kann jedoch dazu führen, dass diese einseitig über das aktuelle Geschehen informiert werden. Filterblasen können eine mögliche Folgeerscheinung hiervon sein. In der vorliegenden Arbeit wird ein Algorithmus für einen Empfehlungsdienst entwickelt, welcher auf Diversität optimiert ist, um dieser Entwicklung entgegenzusteuern. Der Fokus hierbei liegt auf dem Erstellen einer Lektüreliste, welche auf eine politische Vielfalt von Zeitungsartikeln ausgelegt ist.

# Abstract

Many people nowadays read news on the Internet. The selection of available articles is often personalized and matches the interests of their respective readers. So-called recommender systems are used for this. When primarily focusing on the interests of their readers, however, these systems can lead to people receiving only one-sided news about recent events. Filter bubbles are a possible consequence of this. An algorithm for a recommender system is developed in this thesis, one that optimizes for diversity, in order to counteract this development. The focus lies on creating recommendation lists, which focus on political diversity of news articles.

# Table of Contents

**1 Introduction 1**

**2 Related Work 5**

2.1 Personalized News Recommendations . . . 5

2.2 Diverse Recommender Systems . . . 6

2.3 Personalized Diverse News Recommendations . . . 7

2.4 Multi-Objective Recommendation Optimization . . . 8

2.5 News Datasets for Training and Evaluation . . . 8

**3 Theory and Algorithm Design 11**

3.1 General Theory on Recommender Systems . . . 12

3.1.1 Collaborative Filtering and Content-Based Filtering . . . 12

3.1.2 User Feedback . . . 15

3.1.3 Optimization Objectives . . . 16

3.2 Diverse Recommender Algorithm Design . . . 17

3.2.1 Defining and Measuring Diversity . . . 17

3.2.2 News Articles Distribution . . . 19

3.2.3 Accuracy-Diversity-Recency Trade-Off . . . 23

**4 Implementation of Recommender Algorithm 25**

4.1 In-App Survey and User Scores . . . 25

4.2 Reading Metrics and Article Scores . . . 27

4.3 Recommender Algorithms . . . 29

4.3.1 Baseline Algorithms . . . 29

4.3.2 Diversity Algorithm . . . 30

**5 Preliminary User Study 35**

5.1 Methods and Experiment Setup . . . 35

5.2 Scrapers and News Outlets . . . 36

5.3 News App Overview . . . 37

5.4 Data Collection . . . 38

5.5 Evaluation Strategies . . . 39

**6 Processing Results 41**

6.1 Reading Metrics and Indirect User Feedback . . . . . . . . . . . . . . . . . . 41

6.2 Article Ratings and Direct User Feedback . . . . . . . . . . . . . . . . . . . 45

6.3 General Feedback on User Experience . . . . . . . . . . . . . . . . . . . . . 46

**7 Discussion 49**

**8 Limitations 51**

**9 Future Work 55**

**10 Conclusions 59**

**A Appendix 65**

A.1 Intake Survey . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

A.2 Recruitment Survey . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

A.3 Exit-Survey . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

A.4 Reading Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67

# 1

# Introduction

Recommender systems help filter information that is relevant to a user from the information that is irrelevant to them. They do so by trying to predict what the particular items are that are closest to a user's preferences and their needs. [Aggarwal, 2016] These systems are widely used by online search engines, e-commerce platforms, social media sites, streaming services and online news aggregators to name but a few of the most prominent applications. [Aljukhadar et al., 2013, Karimova et al., 2016, Ziegler et al., 2005] With the help of recommender systems, search engines can display personalized search results based on the user's search history. E-commerce platform can analyze the purchase history of their customers and recommend new products. And online news aggregators allow their readers to customize the news feed such that they get only articles and opinion pieces on the specific topics they are interested in.

The goal of this thesis is to create a such recommender algorithm. The target domain is news recommendations. The focus lies on designing and implementing an algorithm that optimizes for political diversity of news articles. This is in contrast to the many accuracy-optimized algorithms currently in use. The recommender system and algorithms implemented here are used in combination with the DDIS News App to conduct further research and analyze how diverse news recommendations influence people's reading behavior. To assess the influences of the diversity algorithm on participants' reading behavior it will be compared to two baseline algorithms, focusing on accuracy as well as chronological ordering of news articles. Their design and implementation is included in this thesis. Furthermore, the algorithm designed here need to be embedded into the existing back end of the DDIS News App.

There are two distinct parts to this thesis. The first part is rather theoretical in nature. Its main objective is to design and then to implement a recommender algorithm for diverse news recommendations (see chapters 3 and 4). The second part is more empirical. The diverse news recommendation algorithm developed in the first part will be embedded into the DDIS News App and tested in a preliminary user study (see chapters 5 and 6). This includes setting up the study, inviting participants and distributing the app. The preliminary user study is followed by a thorough evaluation of the gathered metrics on reading behavior of the participants in order to further improve the performance of the app as well as the diverse recommendation algorithm.

The motivation behind introducing diverse political news recommendations is threefold. The first reason has to do with negative effects that can arise from recommender systems that are solely focused on optimizing for accuracy. One major drawback of non-diverse recommendations is that they could give raise to so-called 'echo chambers' or 'filter bubbles.' [Tintarev, 2017] Echo chambers and filter bubbles describe situations in which users receive one-sided recommendation. In the domain of news articles, for example, users can be deprived of stories or opinion pieces that question the beliefs held by them or expand their horizon of knowledge, had it not been for a non-diverse recommender algorithm. Being only exposed to opinions already held creates a vicious cycle or reaffirming of one's own views, beliefs and prejudices, without knowing other people's deliberations on a particular subject matter.

Ignorance like this can be very problematic in the context of political topics and political news that one receives from their media outlets of choice. This is especially true if one considers that the majority of people in Switzerland already use online news outlets as their primary source of information and thus most likely are already exposed to a number of recommender algorithms.[1] In the context of news recommendations, accuracy-optimized algorithms can lead to situations where individual users are deprived of articles expressing opinions not aligned with their own. This can be detrimental to their ability of deliberative decision making, which is a key element of any liberal democracy. [Habermas, 1962] One way of counteracting this development is to implement new types of recommendation algorithms. Ideally, these are recommendation algorithms no longer optimize solely for accuracy but for also take diversity into consideration.

The second reason why it is crucial to advance the research and development of diverse recommenders is that in some areas there is a distinct mismatch between what operators of platforms using accuracy optimized recommender systems program the algorithms to recommend and what the customers' real needs are. On e-commerce platforms, for example, recommender systems are optimized to create an exact user profile such that complementary or related products can be recommended; the goal of which is that users buy more items. [Szlavik et al., 2011] The actual needs of the customers, however, are irrelevant in terms of the deployed recommendation strategy.

The same holds true for news outlets and news aggregators. Many focus on clicks and optimize recommendations in such a way that the new content one receives closely matches the past reading history. [Liu et al., 2010] For news outlets, the main motivation here is to capture the user's attention for as long as possible with the goal of getting more clicks and them staying longer on the site. These are the most important metrics to optimize, for they directly influence how much ad revenue is generated by an online outlet. As a result, if user activity is tracked over different news sites, the diversity of read articles tends to be larger than what any particular outlet would recommend to them. [Flaxman et al., 2016] In other words, recommender systems of particular news sites create reading lists narrower than users' actually reading interests.

[1] Study on news consumption in Switzerland, conducted in 2015 by researchers of the University of Zurich: https://www.media.uzh.ch/de/medienmitteilungen/archive/2015/schweizer-informieren-sich-hauptsaechlich-ueber-das-internet.html

One explanation for this narrow range in which recommendations fall is that broader recommendations do come with a higher risk of a potential mismatch, since they do not match as closely the user's profile. But this mismatch also indicates that there is potential room for improvement in terms of trading accuracy for diversity. If implemented properly, a more diversity focused algorithm could match users' interests more closely, creating a more engaging experience. It is precisely for this reason that there is an accuracy baseline algorithm included in the experiments setup. If diverse recommendations create a more engaging user experience than accurate recommendations, this might even result in a change of how commercial recommender systems operate.

The third and final reason for political diverse news has to do with ethical and legal considerations related to transparency and data privacy when it comes to how recommender systems use accumulated user data for the purpose of creating accurate customer profiles. One major problem that users have with many commercial recommender systems have is that they are opaque. [Mittelstadt, 2016] These problems of transparency and privacy became even more prevalent in the wake of the introduction of the GDPR, the General Data Protection Regulation of the European Union, which guarantees each user the right to know what data is stored on them and how/for what purposes it is being processed.[2] This includes any use of personal data for the purpose of creating personalized recommendation lists, such as accessing recent purchases, the reading history and other personal data. The algorithm proposed here is designed with privacy in mind. For the purpose of diverse political news recommendations there is but one value stored for each user, assigning them a political score. No additional personal information is required. This way, the user knows at any given time what data is processed. In addition to that, the system can be setup in such a way that all users enjoy full anonymity.

[2] See GDPR article on 'Transparent information, communication and modalities for the exercise of the rights of the data subject': https://gdpr-info.eu/art-12-gdpr/

# 2

# Related Work

Recommender systems are a vast area of research. The domains most relevant to this thesis, however, can be narrowed down to the work done in the field of diversity optimization and news recommendations. The following sections will provide an overview of the current research conducted in these areas. The first section will discuss recommender systems that specifically focus on personalized recommendations of news articles and the different techniques available for doing so. The second section will then provide an insight into the work that has been done in the field of diverse recommendation objectives. Here, the focus lies on the questions of what algorithms are available today, where are they used and to what extent can they be integrated into the algorithms implemented in this thesis. The third part then takes a closer look at the intersection of diverse recommender systems and news recommendations. It takes into account both previously discussed elements, combines them and further elaborates on the work that has been done in this area of research.

After providing an overview of the progress in this field, the two subsequent sections will focus exclusively on open challenges. Multi-objective recommendation optimization is the first one, dealing with the optimization of two or more objectives at a time. The second problem then is related to datasets available for training and evaluation of algorithms in the context of diverse political news recommendations.

## 2.1 Personalized News Recommendations

The first topic looked at in greater detail is the topic of personalized news recommendations. There is an ever-growing variety of how the news articles are custom-tailored to the needs of a specific user. However, one of the oldest yet still widely used examples is customization by category selection. [Liu et al., 2010] Users selects a number of topics they are interested in from a given selection when they setup their account with on news sites. They then get a customized feed of articles from the chosen topics. Approaches based on such a selection focus solely on accuracy. Diversity is not considered in any way. This approach to of how news articles are presented to users, however, can lead to them receiving but one-sided information.

More advanced implementations of news recommendations include collaborative filtering, which result in a longer average reading/visiting duration for news articles. [Garcin et al., 2012, Liu et al., 2010] The idea behind collaborative filtering is that a user's recommendations get influenced by the reading behavior of other users. In the previous example, whenever a user browses their news feed, articles are typically displayed in chronological order. Using collaborative filtering, the system now takes a closer look at all the users that have a particular category in their news feed. It then counts the number of users who read a given article. Articles with more readers subsequently get ranked higher in their respective categories.

Both of the above presented approaches require initial user interaction in order to start recommending news articles. However, there are ways to fully automate this process. One option is to observe user behavior over a certain period of time, monitor what the categories are of which they read articles and then provide the users with a selection of news stories from related, similar or identical categories. [Garcin et al., 2013] This approach is also known as information filtering. [Liu et al., 2010] The recommender system makes use of implicit user feedback, i.e., users' reading histories, and tries to extract a number of meaningful properties to look for in other articles.

One major drawback of all these approaches is that they rely on pre-established news categories. [Epure et al., 2017] And in case there are none or more than one news category available, then an article might not be processed correctly by the recommender system. In addition to that, the news aggregators must make sure that there is a consistent labeling of categories across all outlets they feature. However, even if all of this is a given, there still remains a problem that prevents the adoption here of these kinds of systems. They are unable to assess the political position expressed in an article on the basis of, e.g., analyzing the semantic information contained in it.

The best result these approaches can deliver is labeling whether or not a given article belongs to the 'Politics' category. Categories are readily available metadata of an article. However, in order to know the political positions expressed in an article, some semantic processing would be required. Unfortunately, there has been little research done in this area of content-based filtering, for only a very limited range of items would even lend themselves to such an approach. [Li et al., 2011b]

## 2.2 Diverse Recommender Systems

The first problem one faces when dealing with diversity is the question of how to establish a meaningful metric thereof. The last section introduced the idea of collaborative filtering for accuracy optimized recommendations. By keeping track of what items were purchased by a user, what news articles they read etc., the recommender system is able to create a user profile. For each user, there is a list of items related to them. Given two users, the system can now rate their similarity in terms of how many items they have in common on their lists. The more items appear in both lists the more similar two users are. And the fewer items they have in common, the more diverse their consumption habits are. [Ziegler et al., 2005]

This approach can also be used to check for item similarity. The similarity or dissimilarity of any two items is established by looking at the similarities among their respective lists of users who purchased or read the items in question. A diverse recommendation list for a user now only contain items that stand in no relation with a user similar to the particular user in question. Although this is but one possibility of how to establish diversity, it is by far the most prominent one since it is closely related to the widely used collaborative filtering.[Han and Yamana, 2017, Vargas and Castells, 2011]

As popular as this approach is, as unfit is it for diverse political news recommendations. The reason for that has to do with the method's roots in collaborative filtering. Similar to the disadvantages of the previously discussed approaches to personalized news recommendations, this method also lacks any context or semantic information about the items it recommends. [Javari and Jalili, 2014] For example, if there is a first edition printing and a second edition printing available of one and the same book, then the recommender system would classify the books as dissimilar for it is very unlikely that if one purchased the first printing they would purchase the second printing as well. Purchase histories do not allow for meaningfully establishing categories of items. Similarly, reading histories do not allow for establishing categories of news articles, let alone political positions expressed in a particular news article.

## 2.3 Personalized Diverse News Recommendations

While there is an ever growing number of publications on personalized news and diversity-optimized recommender system, there is little to no work done at the intersection of both topics. As outlined in the first section of this chapter, personalized news recommender systems primarily focus on providing users with a highly personalized list of news articles that focus primarily on accuracy. In contrast to that, research on diverse recommender systems do generally avoid diversity of news articles or diverse political news. Diversity is very item-specific and each type comes with its particular set of challenges. Dealing with news items is especially difficult. It does come as no surprise then that the research conducted in the area of item diversity primarily focused on books, movies and music recommendations. A recent survey paper found that but two of 72 papers published on diverse recommender systems since 2001 made use of collaborative filtering in the context of on news articles. [Han and Yamana, 2017]

The way the two studies who considered news article circumvented the problem of there being no reliable context/semantic information with collaborative filtering is by looking for specific keywords in the news articles to establish article categories. [Abbar et al., 2013, Li et al., 2011a] What these methods achieved was a diverse recommendation in terms of article topics, e.g., politics, sports etc. However, these models are not fit to be adopted for diverse political news recommendations. Simply looking at keywords without context is generally not detailed enough assign a political label to an article. Semantic processing would be required for that. While semantic labeling models do exist, the resulting categories are not specific enough to determine what political opinion a given article expresses. [Irsan and Khodra, 2019]

## 2.4 Multi-Objective Recommendation Optimization

The objective of a recommender system is what it optimizes for. Apart from optimizing for accuracy and diversity, it is also possible to have recommender systems that focus on novelty, serendipity, confidence and trust to name but a few examples. [Aggarwal, 2016] The optimization objective usually does not pose any substantial problem as long as there is but one objective. If there are multiple non-contradicting ones, then it is also not a problem. Unfortunately, when it comes to diverse political news articles, there are two relevant optimization objectives and they are in conflict with one another. They are diversity and recency. Previous studies suggested that an article has but a very limited lifetime after which they are no longer of any particular relevance to readers. If a news article is not recommended during its lifetime, then it cannot be recommended at all. This is unlike, e.g., books or movies, which usually have no lifetime associated with them. For example, if someone like a certain genre of music, the age of particular piece is generally irrelevant for the recommendation process.

This is different for news articles. Empirical studies have shown that the lifetime of a news article is around one day. [Garcin et al., 2012, Garcin et al., 2013] And after one day is over, readers tend to ignore any articles, even if they might agree with the political thought expressed in these articles. This very short time span means that it is crucial to put a heavy emphasis on the temporal dimension when dealing with news articles. There are already a number of optimization techniques available to find the optimal trade-off between contradicting objectives. [Rodriguez et al., 2012] Unfortunately, there are no studies or findings available that apply these techniques to the domain of news articles and the recency trade-off. [Chakraborty et al., 2019]

## 2.5 News Datasets for Training and Evaluation

There is a large number of readily available resources for training and offline evaluation of recommenders that focus on media objects such as books, movies, music etc.[1] Unfortunately, there are currently no data sets dedicated for training and offline evaluation of diverse political news recommendations found in the literature that also emphasize the element of recency. [Karimi, 2018] Datasets that would have been relevant, such as the *Yahoo! News Feed* dataset or the *Bing Toolbar* dataset have either been deleted or are not publicly available.[2] [3] One of the only available dataset is the *Adressa News Dataset*.[4] However, this dataset does not lend itself well to testing for political diversity for it contains news articles from only one newspaper. Furthermore, there are no labels given for political orientations of news articles. Only broad categories labels are available and no fine-grained evaluation of expressed opinions.

[1] Overview of the largest and most well-known datasets for the purpose of testing and offline evaluation of recommender algorithms: https://github.com/daicoolb/RecommenderSystem-DataSet

[2] Information on the *Yahoo! News Feed* which is still features on their website, yet no longer available for download: https://github.com/danijar/semantic/wiki/Recommendation-Data-Sets

[3] The *Bing Toolbar* dataset was featured in [Flaxman et al., 2016] but is not publicly available.

[4] *SmartMedia Adressa News Dataset* of news articles: http://reclab.idi.ntnu.no/dataset/

It is not impossible to use the resources available for the purpose of offline recommender evaluation. However, there are three major drawbacks to such an approach. First, unsupervised means for establishing the performance of diverse recommendations aggregate article diversity and calculate averages. [Hijikata, 2014] There are currently no methods proposed in the literature for evaluating the distribution of articles across a given dimension in terms of their diversity.

Second, and this ties back to the problem of multi-objective recommendations, books, movies and other media objects are not close enough to news articles for them to be a good measurement to compare against. The temporal aspect cannot be accounted for. Book and movie choices in these datasets do not feature the important element of having a short lifespan that is crucial for news consumption. [Lathia et al., 2013]

As a third and last point, it is important to mention that measuring how good a diverse recommender algorithm performs in the setting of diverse political news can only be partially evaluated by means of offline tests with a dataset. One of the techniques available is to see how many of the recommended news articles were actually read by the user in question. [Hijikata, 2014] The goal of diverse political news recommendations is to inform people of what happens across the entire political landscape. The quality of read recommended articles must be assessed and not the quantity of accessed articles over the total of recommended ones. In order to collect the data relevant for doing this kind of evaluation, a user interview or exit-survey is needed instead.

# 3

# Theory and Algorithm Design

Recommender systems differ from one another in terms of the technique they use for collecting the necessary data on the users, the feedback for collecting data as well as the specific optimization objective they feature to match users and items. These three characteristics outline and define a recommender system. Accordingly, the following sections on general recommender system theory will provide a discussion of each of these three elements. It elaborates on the decisions made and techniques chosen to be implemented in the final recommender algorithm proposed here. This first part of the chapter is more theoretical in nature. It introduces as well as defines the various concepts used when discussing recommender systems.

The second half of this chapter focuses on more specific questions regarding diverse recommendations in the domain of news articles. It continues the discussion by emphasizing the problem of diversity of recommendations. The aim here is to outline the different types of diversity that a recommender can optimize for and what the most useful approach is in terms of diverse political news recommendations. The next step of designing the algorithm consists of establishing what a diversity distribution of news articles ideally looks like from a normative perspective. This goes hand in hand with defining the distributions for the baseline algorithms that are used for evaluation purposes. The third section will then focus on the item-specific trade-offs in the domain of diverse news articles. The goal is to establish an optimization objective that can be used for the purpose of diverse news recommendations.

The goal of recommender systems, as outlined in the introductory part, is to help filter information and provide users with a selection of relevant items, here news articles. The upcoming sections of this chapter all lend themselves to two slightly different models or interpretations of the recommendation problem: the prediction version and the ranking version. [Aggarwal, 2016] In the prediction version of the problem, the recommender calculates for each user-article pair a value of how relevant this article is for a given user. The rating version of the problem focuses on determining only the top-$k$ items that are with a certain probability to a user's liking.[1]

[1] Please note that in the context of the topic of news recommendations discussed here, the terms 'news article' or simply 'article' will be frequently used in later parts as opposed to the more general and broader term 'item.'

In the context of news articles with a short lifetime and a news app with only limited space available to display them, it is favorable to select the ranking approach over the prediction version. News articles having a short lifetime means that there is no need to have a user-article score for all but the most recent items. Furthermore, there is a limit on how many articles can be displayed inside the app. This limit is determined by the available system resources in the back end and directly influences the frequency with which users get new recommendations. By focusing only on the top-$k$ items, experimenters can tweak the limit of articles depending on the number of participants to directly control the frequency of the recommendations. It is for this reasons that final implementation of the recommender algorithms will make use of the ranking model of the problem. The upcoming sections, however, will focus on the prediction model, for it is more general and allows for the ranking model to be derived from it.

## 3.1 General Theory on Recommender Systems

### 3.1.1 Collaborative Filtering and Content-Based Filtering

On the internet, recommender algorithms are ubiquitous. Each platform that deploys a recommender system can make use of a wide range of data available on each of their users or customers. Recommender systems calculate probabilities for items and predict how relevant they are for a given user. They do so be using a utility function $u(c, s)$ in order to compute how viable an item recommendation $s$ is to a user $u$. [Aggarwal, 2016, Adomavicius and Tuzhilin, 2005] The recommender system calculates item scores for a particular user $c \in C$, where $C$ is the set of all users given an item $s \in S$, with $S$ being the set of all items. [Adomavicius and Tuzhilin, 2005] What these particular scores are depends on the recommender system in question. With movie, for example, these can be star-based rating. With products on e-commerce sites it can be a purchase recommendation as simple as *Yes/No*. [Aggarwal, 2016]

There are two distinct techniques of how recommender systems can process data to calculate the prediction score of items relevant to a user. The two available approaches are collaborative filtering technique and content-based filtering technique.[2] [Adomavicius and Tuzhilin, 2005, Isinkaye et al., 2015]

Collaborative Filtering: Collaborative filtering makes use of previously collected user data. For example, on e-commerce platforms, the purchase history of a customer is analyzed and compared with the purchase history of the other customers. In this particular case, collaborative filtering recommenders look for pairs or groups of users with a similar purchase history. They then recommend items that a particular user has not yet purchased, but that were purchased by their respective peers. [Aggarwal, 2016]

[2] It is worth mentioning here that the way the data on users and items is collected and accessed is outside of the scope of the topic of recommender systems. The way a recommender accessed data depends on the particular architecture and design of a given back end. It is for this reason the general outline presented here lacks any particular detail on how data is collected or accessed.

Collaborative filtering, however, cannot only be used to group customers, but it also works from the perspective of items. [Aggarwal, 2016, Isinkaye et al., 2015] Instead of looking for users with a similar purchase history, it is possible to look for products that share a common customer base. The following sections, however, will only focus on finding similar users. In this context, an unknown item rating for a user $u(c, s)$ must necessarily be determined by a known item rating that one of their similar peers has given. In other words, a rating for user $c_i$ is derived from $u(c_j, u)$, where $c_j$ is a user similar to $c_i$. Another way of expressing this is to look for the similarity $sim(c_i, c_j)$ between a user pair. [Adomavicius and Tuzhilin, 2005]

User similarity for collaborative filtering is usually expressed in terms of cosine similarity. [Aggarwal, 2016] Given a pair of users $sim(x, y)$ where $x = c_i$ and $y = c_j$, their similarity can be expressed as the cosine angle between the two in an $m$-dimensional space, where $m = S_{xy}$ and $S_{xy}$ being the number of items both $x$ and $y$ have previously rated. [Adomavicius and Tuzhilin, 2005, Aggarwal, 2016] With $r_{c,s}$ being a particular rating given to $s$ by $c$, the following equation can be used for calculating user similarity:

$$sim(x, y) = \frac{\sum\limits_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum\limits_{s \in S_{xy}} r_{x,s}^2} \sqrt{\sum\limits_{s \in S_{xy}} r_{y,s}^2}} \tag{3.1}$$

Please note that $r_{c,s}$ can be any kind quantifiable rating as long as it is consistently applied over all users and for any of their items. Furthermore, it is possible to add additional details to how this rating is taken into consideration. Empirical studies have shown that there are certain users that tend to give higher ratings across the board than other users, which can be an argument for normalizing the scores. [Adomavicius and Tuzhilin, 2005, Aggarwal, 2016] And in order to now get the users that are most similar to one another, it is necessary to pair-wise compare each and every user. Collaborative filtering works best if there is readily available data on users.

In the absence of such data, the technique struggles to provide meaningful recommendations. The most challenging aspect is dealing with new items where there is only sparse data available. This is referred to as the cold-start problem. [Aggarwal, 2016, Isinkaye et al., 2015] It is a problem that is especially prominent in the domain of news recommendations, where the items that should be recommended are generally speaking the newest ones with the sparsest data.

In addition to problems related to the cold-start, there is the fact that when applying collaborative filtering, all the differences or similarities of the recommended items are ultimately rooted in the behavior of the users. Collaborative filtering on its own does not allow for, e.g., topic specific recommendations, for it does not look at any particular item attributes. It comes as no surprise then that there has never been an experiment using only collaborative filtering to optimize for diversity of news articles. [Kunaver and Pozrl, 2017]

Content-Based Filtering: The idea behind content-based filtering is to establish a number of item properties and assign them a value based on the content of the item. [Isinkaye et al., 2015] The major advantage this method has over collaborative filtering is that there is no need for extensive data on user interactions to the recommendation process. However, what is needed is a detailed user profile, listing preferences that can be matched to item properties. In this context, $u(c, s)$ specifies the utility or rating of the content $s$ of an item to the user profile $c$. This can again be expressed as a problem of determining the cosine angle, but now between a given user-item pair. [Adomavicius and Tuzhilin, 2005, Aggarwal, 2016] The equation used to calculate content-based is as follows:

$$u(c, s) = \frac{\sum_{i=1}^{K} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}^2} \sqrt{\sum_{i=1}^{K} w_{i,s}^2}} \tag{3.2}$$

Please note that it is crucial for all users and items to have set of properties $K$ that they share amongst one another and that needs to be of the same dimensionality. [Adomavicius and Tuzhilin, 2005] The different dimensions in the user and item set at $w_i j$ can be individually weighted when calculating the overall sum. This makes it possible to either increase or decrease the contribution of a particular property to the overall cosine similarity of the two. Having a set of dimensions at one's disposal that can be freely modified not only allows for this kind of fine-tuning of certain weights, but also for introducing a use case specific sets of attributes.

In the case of collaborative filtering, one is restricted to properties that can be derived from, e.g., the item purchase history. Here, any number of dimensions an experimenter can think of are possible to be included. An additional advantage a content-based approach offers over collaborative filtering in the context of news is that even recently introduces items can be reliably recommended. The remaining challenge, however, is how to reliably assess and calculate the $K$-dimensional values for all items. [Aggarwal, 2016] Even sophisticated deep learning approached struggle with calculating the relevant scores for items in real-time if content analysis goes beyond knowledge graph or named entity recognition in the domain of news articles. [Wang et al., 2018]

Collaborative filtering and content-based filtering, however, are not mutually exclusive. Both techniques can be used in combination with one another to overcome their respective weaknesses. [Li et al., 2011a] A great number of so-called hybrid approaches have been proposed. [Abbar et al., 2013, Aggarwal, 2016, Isinkaye et al., 2015, Li et al., 2011b] It is for this reason that the implementation of the recommendation algorithm here will feature elements of both techniques.

### 3.1.2 User Feedback

In order to create recommendation, there first needs to be a set of metrics or feedback data that the recommender can make use of. Generally speaking, recommender systems make use of either implicit or explicit feedback. [Isinkaye et al., 2015] Feedback is collected for the purpose of both collaborative filtering as well as content-based filtering. The goal is to create accurate user profiles and/or item profiles. In the case of collaborative filtering, the feedback is required to create a user history of their interactions with items, e.g., a purchase history in an e-commerce setting or a viewing history in the case of a streaming service. Feedback gathered in the context of content-based recommenders is usually provided by the user themselves when they, e.g., select a range of topics for their feed from a news aggregator or when they create a user profile for a music or movie service, where they list their favorite genres.

More specifically, there are two ways of how feedback can be gathered in order to influence the results the recommender systems produced. The first approach is to have explicit or direct user feedback. One common way of how this data is collected is to present the users with the option to like/dislike the recommendation a system presents them. [Aggarwal, 2016] Another way is to offer them the option to create and customize a user profile where they can list their interests. The second way of gathering metrics is by implicit or indirect feedback. Implicit feedback is generally associated with monitoring practices of all the users' actions. [Adomavicius and Tuzhilin, 2005]

On e-commerce platform, for example, all the site interactions of a user can be monitored, their browser finger print analyzed and their third-party cookies of the browser accessed. That way, relevant information about, e.g., their location, can be stored and processed. Recent approaches even take into account the context of the feedback and all the previous interactions with the system in question. [Peska, 2016] Implicit and explicit feedback are not mutually exclusive and commonly used in combination with one another. [Isinkaye et al., 2015] They can even be seen as complimentary and enable the system to capture more meaningful user interactions.

In addition to there being different methods of how to collect user feedback, there are also two different feedback types, namely positive feedback and negative feedback. Examples of positive feedback are purchase histories of users and data gathered by monitoring. Examples for negative feedback are recommended items that were skipped and/or otherwise rejected. Unfortunately, negative feedback is often not considered despite findings that show a performance increase when doing so.[3] [Zhao et al., 2018] Similar to implicit and explicit feedback, positive and negative feedback are not mutually exclusive. For the news app, is it planned to capture negative as well as positive feedback, both in a direct as well as indirect fashion. The feedback gathered this way can be used for two purposes. First, it allows to the recommender system to improve and develop a more detailed user profile. Second, by capturing positive and negative feedback, it is possible to make adjustments to the recommender algorithm itself, e.g., resulting in more refined weights for calculating content-based item scores. [Isinkaye et al., 2015]

[3] Even standard solution like *MyMediaLite* do offer but positive-only feedback for algorithm training and evaluation, derived from implicit or explicit feedback: http://mymedialite.net/

### 3.1.3 Optimization Objectives

Processing user feedback allows for collaborative filtering and content-based filtering in order to calculate user and item scores. However, defining how the data for recommendations is gathered is but one part of building a working recommendation algorithm. On their own, these scores do not yet allow for recommending any items to users. In order to do so, an optimization objective is needed. The optimization objective is what formulates the particular recommendation strategy, i.e., a strategy for saying what user score should be matched with what item.

The most popular optimization objective is the one of accuracy. [Aggarwal, 2016]. The previous section introduced pairwise user-user similarity for collaborative filtering and pairwise user-item utility for content-based filtering. In this context, an accuracy objective can be stated in terms of either maximizing pairwise user-item utility/rating for content-based filtering or minimizing pairwise user-user similarity for collaborative filtering as is shown below. [Adomavicius and Tuzhilin, 2005]

$$\forall c \in C, s' = \arg\max_{s \in S} u(c, s) \tag{3.3}$$

$$\forall x \in C, y' = \arg\min_{y \in S, x \neq y} sim(x, y) \tag{3.4}$$

Accuracy lends itself well to be an objective, function for it is relatively easy to formulate and to implement. In the above example, *s'* and *y'* are the item and user that match closest the interest or profile of a given user *c*. These implementations of accuracy-optimized objective have been thoroughly studied in the past. [Aggarwal, 2016] Using equation 3.3, creating a recommendation list using the content-based technique can be achieved by calculating all pairwise values and then sorting them in descending fashion. When using equation 3.4 for the purpose of collaborative filtering, the ordering is reversed and the two closest matches are optimal.

Research in the area of news recommendations, however, nowadays focuses heavily on diversity optimization. [Karimi, 2018] One reason for doing so are findings of studies suggesting that the inclusion of a diverse selection of news stories is perceived by users to add substantial value to the recommendations. [Castells et al., 2011] Looking now at diversity, one way of formulating this objective is to look at a set $R$ of recommendations and then calculating the average dissimilarity between item pairs. [Zhang and Hurley, 2008] $f_D$ can be a function calculating the diversity of this set. By adding, removing or replacing articles, the diversity subsequently changes and can be fitted to a given target. A formalization of which could look as follows:

$$f_D(R) = \frac{2}{p(p-1)} \sum_{i \in R} \sum_{j \in R, j \neq i} d(i, j) \tag{3.5}$$

In this example, $p$ is equal to the length of |R|. Furthermore, for establishing dissimilarity between individual items, a distance function $d(i, j)$ is needed. It depends on the particular use case how exactly this distance is established. One way doing do is to, again, calculate feature vectors and determine the cosine angle between all item pairs contained set $R$. Unfortunately, there is one major drawback with this way of assessing the diversity of recommendations. When looking at average item dissimilarities, the exact distribution of article scores over a given dimension is lost. The diversity of, e.g., three loosely related items can be equal to the diversity of two closely related items and one item with a value vastly different from the other two.

And when using this definition of item diversity in combination with collaborative filtering, then it is not possible to establish political diversity of a set of recommendations. The reason for this is that, again, collaborative filtering is ignorant of the information an item carries and looks only at the user-item interaction history. It is for this reason that diversity and collaborative filtering are generally avoided in the context of news articles; in cases where they are used in combination with one another, it is only for the purpose of, e.g., content pre-filtering, as part of a multi-stage recommendation process. [Karimi, 2018] This approach of diversity, however, is very popular in combination with content-based or hybrid recommender systems. [Aggarwal, 2016, Karimi, 2018]

Diverse recommendations in combination with content-based filtering primarily focuses on a selection of diverse categories of news; there has only been one study that diversified content within a given category, by making use of knowledge graphs for labeling articles based on their content. [Wang et al., 2018] The open challenges for the algorithm implemented here are establishing and assessing diversity of news articles in terms of their content as well as finding a way for reliably measuring this article diversity across a given number of different political dimensions that does not rely on making use of calculating average distance values for recommendatino lists.

## 3.2 Diverse Recommender Algorithm Design

### 3.2.1 Defining and Measuring Diversity

Diversity in recommender systems in by no means a new topic. Extensive work has already been done in this area. [Kunaver and Pozrl, 2017, Vargas and Castells, 2011, Ziegler et al., 2005] However, the adoption of diversity optimized recommenders is not as easy as adopting accuracy optimized solutions. The reason for this is that there are a number of problems related to the definition and measurement of diversity, which are very much dependent on the particular item in question. [Chen et al., 2016] When implementing an algorithm focusing on recommending diverse news article, it is important to first clarify the concept of 'diverse' exactly means here. Unlike accuracy, for example, there is no general maximization or minimization approach available to achieve the desired item recommendations. To do so, there are two important aspects that need to be specified when optimizing for a diverse selection of news.

First, it is important to define the type of diversity in question. There are two types available that a recommender system can focus on: individual diversity and aggregated diversity. [Metla et al., 2014] Individual diversity focuses on a diverse selection of items or news articles within the recommendation list of a particular user. Aggregated diversity on the other hand focuses on a diverse selection of news articles across all recommendation lists. The goal of the DDIS News App is to analyze the effect of diverse news recommendations on users' reading behaviors. When using aggregate diversity, the level of diversity among recommendation list can vary. It is crucial, however, that the level of diversity among all participant is equal. If that is not the case, then the experiment setup does not allow for an assessment of the impact of diverse recommendations, for some users might not even receive well-diversified news to begin with. It is for this reason that the algorithm implemented here focuses solely on individual diversity.

The second question that needs answering when dealing with diverse recommendations is related to the particular aspects of the recommended items that should be diversified. It is the definition of the above-mentioned distance function $d(i, j)$ that is needed to specify what particular items properties are relevant for establishing item diversity. In this context, there is no general definition of what item diversity is that fits in all situations. What the exact item properties are that need diversification depends on the particular research question. In the case of the DDIS News App, it is political diversity among news articles that should be diversified. The relevant item dimensions related to the property of being politically diverse have to be further specified. Ideally, they are fully dependent on the semantic information that a given news article provides.

Another important aspect when dealing with diverse item recommendations is to consider the normative nature of this area of research. Unlike with other optimization objective, e.g., accuracy optimization, there is a much broader range of possibilities to consider when determining how a diverse distribution of items should look like. There being no default approach to diversity available means that research in this area must be conducted in a more normative fashion. [Aggarwal, 2016, Wang et al., 2018] In other words, the specific distribution chosen express more the underlying assumption researcher have about, e.g., the optimal diverse reading behavior study participants should exhibit, rather than being a diversity measurement everyone can agree upon.

There are two difficulties here that need to be tackled before discussing any particular experiment setup. The first issue has to do with making explicit the underlying assumptions of the researchers motivating a given normative approach. All the assumption need justification for they provide the foundation the diverse recommender system is built on. The second challenge is related to the fact that diverse recommendations influence the behavior of users in such a way that the initially assumed reading behavior is no longer visible. In order to make sure that these claims about user habits were shown to be accurate, however, there needs to be an additional baseline group of users that are not subjected to diverse recommendations to compare against. Both of these aspects are discussed in the upcoming section in the design of a diversity-optimized recommendation algorithm for news articles.

### 3.2.2 News Articles Distribution

The recommender algorithm implemented here will be features in the DDIS News App and provide users with a diverse selection of news articles. In order to program the algorithm, however, it is first necessary to define what the diversity distribution of news articles should look like. The article distribution presented in this section will be an idealized version of what a diverse reading list of news articles for a well-informed user in terms of political topics should look like. The assumptions necessary to create such a distribution are ultimately rooted in the understanding of how democracy works, what its key elements are and how civic virtue necessary for a liberal society are best promoted. It is for this reason that the first part of this section will be a more theoretical part, outlining the underlying democratic theory used here and providing justification for the selected diversity model. The second part then outlines a possible way of how this underlying assumptions can be turned into a possible diversity distribution that can be featured in the DDIS News App.

Public discourse is a crucial element for all members of a society. [Habermas, 1962] This includes topics located in the political domain. Sharing and arguing for the advantages and disadvantages of a given policy is the essence of political discourse. [Mittelstadt, 2016] These claims are supported by empirical studies that show exposure to dissonant political views increases tolerance, i.e., the ability to see and follow the arguments of the counterparty. [Mutz, 2002] Promoting a more diverse political discussion becomes especially important in the context of increasing usage of personalized media outlets. [Möllera et al., 2018] Studies found that these platforms often have can feature echo chamber-like structures. [Colleoni et al., 2014]

A possible way of counteracting this development is to create a common ground for people to engage in public discourse. One approach to doing so it to recommend news in such a way that there is a large overlap of articles that all people read regardless of their particular political orientation. The articles recommended this way are preferable located in between the extreme political viewpoints. The reason for this is that news articles expressing with less of an extreme position generally attract a larger audience. [Flaxman et al., 2016] For this recommendation strategy to work, it is necessary to have data available on the political orientation expressed by a given news article.

This directly leads over to the technical challenges and the questions of what article scores are used to capture the political opinion express in an article as well as how the score is assigned to a news article in the first place. The DDIS News App currently focuses on the German-speaking part of Switzerland. Future experiment will be conducted here and the news outlets that provide articles focus on the Swiss political landscape. This context motivated the adoption of a political survey tool that is available nationwide and frequently used during general election.[4] It allows for a reliable assessment of political scores through a way that not only rests on a sound theoretical foundation but that is also field-tested over many years and regularly updated.

[4] The political survey adopted here was created and published by *Politool*: https://politools.net/ (The complete list of the survey questions adopted here is added under section A.1 to the appendix.

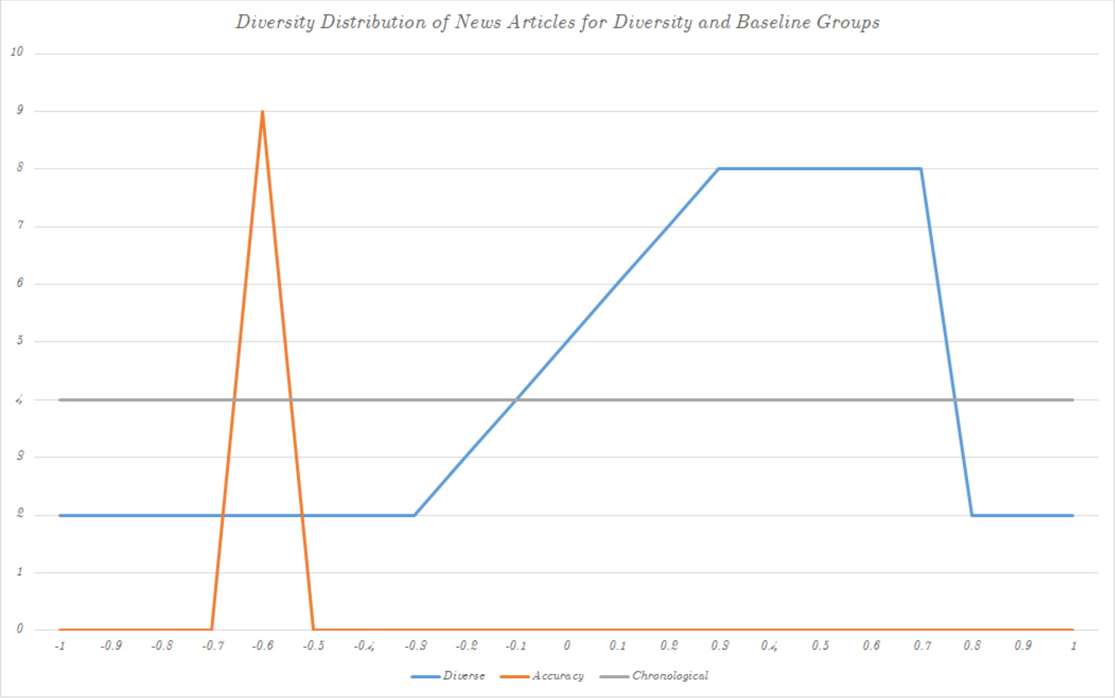


Figure 3.1: Sample distributions of news article for the three user groups.

The survey will be used here in order to assess a two-dimensional political scores. One dimension describes how far left/right the opinion expressed in a given article is. The second dimension looks at how conservative/liberal the ideas are. The news articles, however, are not directly rated. Instead, a hybrid approach of collaborative and content-based filtering is used to assign the news articles a political score on the basis of their readership. Their readership, i.e., the participating users, take the survey when signing up for the app. They are then given a two-dimensional political score. Whenever they read a news article, the recommender system processes the reading metrics and assigns a political score to the articles that were accessed by the users.

In order to circumvent the cold-start problem and any undesired rating feedback loops, the users base will be split into different groups. One group it the chronological reading group. This group does not receive diverse recommendations. Its recommendations are orders by the date an article was published. The purpose of this group is not only to serve as a baseline to compare against, but it is also the group whose members' reading metrics are used for an assigning political scores to news articles. By having a dedicated group rating the articles, this approach circumvents the cold-start problem. The next group it the one receiving diversity-optimized news recommendations. Their reading metrics are not taken into consideration for rating news articles. The same holds true for third experiment group, the group of users receiving accuracy-optimized news recommendations, mimicking a situation where users are inside an echo chamber. Both chronological and accuracy-optimized groups serve as baselines.

Figure 3.1 shows an examples of how the different idealized distributions for these three groups looks like across one political dimension. The figure quantifies on the y-axis how many news articles a user receives that are of a given political score, as indicated on the x-axis. The political scores on the x-axis range from -1.0 to +1.0 and are used for assessing both the left/right dimension as well as the conservative/liberal dimension of an opinion expressed in a news article. The numbers used here on the y-axis are but sample values to better illustrate the different recommendation strategies. The gray curve is the distribution for the chronological baseline. The blue one for the diversity group and the orange curve for the accuracy group. The political score or users is analogous to the score of news items. The score includes for each of the two political dimension in it a value ranging from -1.0 to +1.0, which gets assigned to the users after completing the in-app intake-survey when signing up for the DDIS News App.

In Figure 3.2, the accuracy-optimized orange curve shows what a news article distribution looks like for a user with the political score -0.6. There is a spike at this particular political score on the x-axis, indicating an increase number of news articles recommended to this sample user. There is a sharp drop off to the left and to the right of -0.6, meaning that no other articles get recommended to users of this group but the ones that are perfectly aligned with their respective political preferences.

The blue curve for the diversity-optimized user shows a different article distribution. It shows how a distribution looks like for a user with a score +0.6. This score lends itself well to a comparison with the distribution of the user in the accuracy group having the opposite score of -0.6. As shown in the figure, the distribution is not simply mirrored at value 0 on the x-axis. What is done instead is that a slightly skewed normal distribution is introduces that extends well over the middle. The reason for this is to create an overlap of news articles that everyone reads regardless of their political scores.

One last point worth mentioning in this regard is that the distribution curve of the gray group receiving chronological news recommendations is completely dependent on the news outlets. It is only flat in the case that a.) news outlets all favor a particular political viewpoint, b.) are evenly distributed across the political dimension and c.) all publish the same number of news articles every day. By gathering some data on the average political score of a news article published by a given outlet and its daily output, it is possible to remove or impose certain filter restrictions. Doing so allows to increase or decrease the number of articles a news outlet can contribute, with the goal of keeping the curve as flat as possible. The benefit of having a flat curve is that the difference in reading behavior across the different groups are more pronounced this way.

This approach being a normative one entailed long discussions over what a reasonable article distribution should looks like. One particular important aspect was whether or not a user should receive any news article recommendations that express a political opinion that is more extreme than their own political viewpoints. Should a user with moderate left-or right-wing viewpoints get recommended articles located at the extreme positions of the political spectrum? One argument against this type of recommendations is that users might be at risk of being radicalized. There are, however, three arguments against limiting the scope of item recommendations.

- First, the news outlets featured in the experiment do not publish any news articles that contain radical anarchistic or even illegal content to begin with. The exposure to extreme political views is generally limited to the ideas promoted by political parties in Switzerland. Furthermore, exposure to ideas promoted by political parties not aligned with one's own preferences is generally something each and every voters encounters on a regularly basis when they have to cast their vote in an upcoming election and gather information on the topic at hand.

- The second reason has to do with the fact that it is generally preferable for people to be aware of what goes on at the fringed of the political spectrum. Even if they are not sympathetic to the ideas expressed there, it is nevertheless crucial for the deliberative process to have been exposed to ideas across the spectrum. From time to time, users should be forced out of their 'political comfort zone' when reading the news. Political theory does backup this approach; exposure to ideas dissimilar with one's preferences usually benefit the inhabitants of the public sphere. [Habermas, 1962, Mutz, 2002]

- And lastly, with this adjustment in place, it would no longer be possible to recommend diverse article to users at the center of the political dimensions. Being located at the middle in between the extremes would entail that one only gets articles recommended that are neutral in terms of the political opinion they express. As a result of that, their article distribution comes close to accuracy optimized recommendations with a peak near to the 0 value within a given political dimension. If that is the case, the user can no longer be said to have receives a diverse selection of news articles.

In summary, the normative distribution given here roots in the assumption that in order for a democracy to work, it requires consensus-driven decision making. Advantages and disadvantages of each policy have to be discussed and collectively deliberated upon. Echo chambers and filter bubbles pose a threat to this way of political decision making, the creation of which are often supported by accuracy-based recommender systems. This is especially true in times where a growing number of people get their news from online sources, which make use of such algorithms for personalizing user recommendations.

Being exposed to but a narrow set of ideas and opinions can lead to prejudices and to a lower level of tolerance for alternative viewpoints. This can be a serious obstacle when trying to work out solutions with peers of a different mindset, for alternative viewpoints were never considered or are unknown to begin with. The algorithm designed here aims at enriching political discourse by focusing on providing people from across the political spectrum with a diverse selection of news articles not limited to their respective preferences. It does so by putting a heavy emphasis on creating a large set of overlapping news articles that each and every person gets to read. The goal behind doing so it to create a common basis for political discourse to emerge. To do so, a distribution similar to a skewed normal distribution is calculated, where the center is located over at a user's political score. This distribution always extends across the center of the political distribution, into the opposite political area that a particular user is located in.

### 3.2.3 Accuracy-Diversity-Recency Trade-Off

The above outlined diversity distribution is an idealized version of how news consumption of individual users should look like. However, before implementation can begin, there are a few practical issues that must be considered first. As a user of a news app, the content presented should be engaging, i.e., motivate to make use of the app for an extended amount of time. If the app does not offer any content that is to the liking of a user, it may result in a reduced level of activity. In return, fewer reading metrics can be collected. In the case of news, the most important aspect to consider here is to account for the temporal dimension of news articles. [Garcin et al., 2013, Saranya and Sadhasivam, 2012]

Each news article has but a limited shelf life. This is something that standard applications of, e.g., book or movie recommender system ignore. With such media items, temporal aspects play only a minor role, if at all. In other words, the recommenders used for these items work under the assumption that if the preferences of a user remain constant over time, then so do the accuracy and diversity scores of the recommendation lists they calculate. This basic assumption, however, does not hold in the case of news articles. Old news is old news. Even if a reader's preferences remain the same over time, their actual interest in a given news story may change heavily depending on how long ago a news article was published.

It is for this purpose that an additional time window is introduced when calculating user recommendations. This time window defines what the maximum age of an articles can be in order for still being considered by the recommender algorithm. Previous studies suggest the time window is smaller than 72 hours, ideally 24 hours [Garcin et al., 2012, Garcin et al., 2013, Karimi, 2018, Wang et al., 2018] In other words, news articles are no longer relevant to a user after three days and thus are dropped for recommendation purposes. This limitation can lead to a situation where recency consideration compete with accuracy- and diversity optimization. In order to get around this problem, the chosen implementation must provide a sound strategy of how to resolve this conflict.

# 4

# Implementation of Recommender Algorithm

The focus of this thesis lies on developing a recommender algorithm that provides users with a diverse selection of news articles. This algorithm must be integrated into the existing back end of the DDIS News App in order for it to be used in a preliminary user study planned here and subsequent research experiments. Accordingly, all the components implemented here must be developed with the existing app architecture in mind. There are three main parts to the implementation of the diverse news recommendation algorithm. To better understand the details of the algorithm as well as its integration into the back end, it is beneficial to focus on each of the parts separately and provide an in-depth description of how they are implemented here.

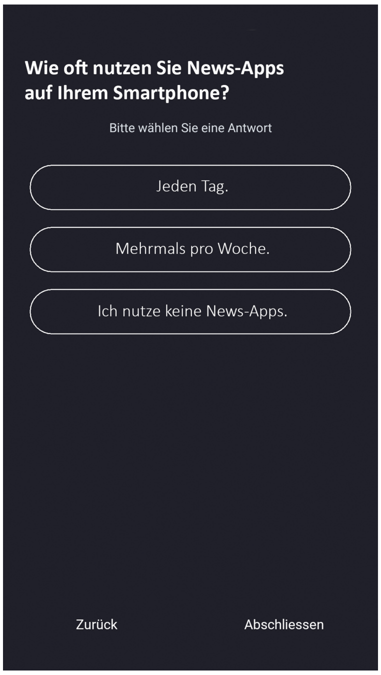
To setup and start any experiment with the DDIS News App, it is necessary for the users to take a survey so that they can be assign political score. Setting up the user survey, the algorithms to calculate the political scores and the scripts to create and store user profile will be the first part looked at in more detail.

The second part that needs to be implemented is the exact rating procedure of news articles. Here, the focus lies on what the reading metrics are featured in the recommender system and how they are subsequently used for the purpose of assign a political label to the available news articles based on their readership.

The third and final part of the implementation then focuses on the algorithms for the actual process of recommending news articles. This section highlights the details for both the baseline algorithms for chronological and accuracy-optimized recommendation as well as the algorithm for diverse political news recommendations.

## 4.1 In-App Survey and User Scores

In order to determine the political orientations of app users a nation-wide survey is adopted. It features 23 question to reliably assess the political views of participants both in terms of how far left/right as well as how conservative/liberal their viewpoints are. The survey will be presented to the user after they have created a DDIS News App account. They are automatically rerouted to the in-app survey and are asked to complete the questionnaire in order to begin using the app.



**Wie oft nutzen Sie News-Apps auf Ihrem Smartphone?**

Bitte wählen Sie eine Antwort

Jeden Tag.

Mehrmals pro Woche.

Ich nutze keine News-Apps.

Figure 4.1: Sample screen of the in-app survey tool.

Figure 4.1 illustrates how the survey is presented to the users. Please find the full list of questions featured in the intake-survey in the Appendix under A.1 (German-only). A user can get rewarded a total of 100 points per question. For each of the available dimensions there are separate point counts the system keeps track of. The maximal number of points one can receive in any given dimension is $N * max$, where $N$ is the total of questions a given dimension and $max$ is the maximum amount of points available per question. There are question-specific weights available in order to calculate the political score for both how far left/right positioned a user is as well as how conservative/liberal their viewpoints are. The weights are either $+1.0$ or $-1.0$. This means the user score for each dimension ranges from $-N * max$ to $N * max$. The starting total is always 0, which marks a neutral point for each of the dimension. For each survey question there are four possible answers available that serve as a point multiplier: yes (x1.0), rather yes (x0.25), rather no (x$-$0.25) and no (x$-$1.0).

A sample question with weights of $+1.0$ for the left/right dimension and a weight of $-1.0$ for the liberal/conservative dimension is assumed to illustrate how the political scores are calculated. If a user answers this question with 'rather yes', they get 25 points added towards their overall total in the left/right dimension $(100 * 0.25 * 1.0)$. In the liberal conservative dimension, however, 25 points are being deducted from the total $(100 * 0.25 * -1.0)$. Final scores for the left/right dimension with a negative value mean the participant has a more left-leaning position. And positive values mean their viewpoints generally align right-wing viewpoints. In the liberal/conservative dimension, negative values cover the conservative part and positive values the liberal part.

## 4.2 Reading Metrics and Article Scores

Once the users completed the intake-survey they are then able to read news articles. The political score calculated during the intake-survey gets assigned to their profile. Before doing so, however, it gets normalized to fit in between the range of −1.0 to +1.0 for. Not dealing with absolute numbers is especially useful in case there are minor tweaks to the survey, where questions are added or their weights slightly tweaked. After storing the political value, the users then get assigned to a randomly chosen group. Participants get either assigned to the chronological reading group, the accuracy-optimized or diversity-optimized group.

The recommender is setup in such a way that 60% or all users are sent to the chronological baseline group. 20% are selected for the accuracy-optimized group and the remaining 20% will be part of the diversity-optimized group. The reason for having a baseline group of three times the size of the recommender-specific ones has to do with its specific task of rating and assigning scores to news articles. One common issue that recommenders face is the cold-start problem where there is simply not enough data available to start recommending newly introduces items. This is especially true for diversity-optimized recommender systems. [Aggarwal, 2016, Karimi, 2018] With over half the user base dedicated to reading and rating of news article, however, this problem should become much less prevalent.

Users do not need to explicitly assign a political score. The rating process happens automatically based on the collected reading metrics. Table 4.1 provides an overview of data that the recommender system collects for any given user-article pair. The underlying assumption at play here is that people's reading behavior are closely aligned with their political views. In other words, if a user from the far-left corner of the political spectrum reads a given news article, then that news article is assumed to express a view commonly associated with left-wing politics or at least it features viewpoints that people from this parts of the spectrum are sympathetic towards.

| Metric | Values |
|---|---|
| Number of times an articles was selected. | INT |
| Amount of time in msec. spent reading an article. | INT |
| The date an article was first accessed. | DATE |
| The date an article was last accessed. | DATE |
| The date an article recommended to a user. | DATE |
| The date an article was added to the reading list. | DATE |
| The date an article was removed from the favorite list. | DATE |
| Whether or not an article is in the archive. | TRUE/FALSE |
| 'Like'-status of an article. | TRUE/FALSE |
| 'Dislike'-status of an article. | TRUE/FALSE |

Table 4.1: List of available reading metrics for each user-article pair.

It is unavoidable, however, that there is a number of outliers in the user base that prefer reading news articles from across the political spectrum. There are threshold values in place to lower the risk of there being a news article mislabeled by a user with an explorative reading behavior. These threshold values determine the minimum number of users that need to have accessed an article before it can be used for accuracy- or diversity-optimized recommendations. Threshold values are but one instrument trying to make the process of assigning a score based on reading behavior more reliable. The following list provides an overview of the additional heuristics that the recommender system makes use of when computing at reading metrics:

- Access/Reading Duration: It is necessary that a user has read an article for more than ten seconds before taking their metrics into consideration. One reason for this restriction is that it is possible for a user to access an article by accident. When doing so they then immediately close the article again. This interaction should be disregarded by the recommender system.
- Like/Dislike Status: Users are able to like and dislike articles. This direct user feedback is used to either boost or nullify a particular user's reading metric in terms of influencing the articles score. If a user liked an article, then their reading metrics will triple, i.e., this user-article pair will be counted three time. And in case an article was dislike, the recommender will no longer consider this particular user-item metrics for purpose of rating the article.
- Archive/Reading List: Users have the option bookmark news article and put them on a reading lists. They also have the option to archive an article so that they can access it at any time, even if it is no longer in their recommendation list. Both are interactions that provide valuable feedback that the recommender makes use of. The recommender system counts this user-article pair double if the news articles are on either one of these lists.

After applying this small set of heuristics, the recommender system assigns each article a political score equal to the average of its readership. This simple way of calculating article scores was purposefully chosen. The reason behind doing to is that it is the diversity-optimization algorithm and its distribution that needs to be implemented and evaluated. Finding or developing a reliable way of how to infer article scores from reading metrics is not the primary objective here. Furthermore, adopting a more sophisticated approach to article rating comes at the risk of introducing a number of additional uncertainties related to properly tweaking and setting up its parameters.[1]

Please note that the values chosen here are by no means absolute. These strategies are but a starting point and it is up to future studies to further tweak and add new rules to this list. In addition to that, it is important to mention that the exact parameters are very much dependent on how active an average user is and on the number of news outlets/daily published news articles that the recommender system processed.

[1] Based on a discussion with Dr. Bibek Paudel, a researcher at Stanford University, who in the past had contributed to the back end of DDIS News App.

Chapter 2.3 introduced a number of approaches currently deployed in the field of diverse news recommendations. Strategies making only use of either collaborative filtering or content-based filtering often face the cold-start problem and issues related to data sparsity when applied to the domain of news articles. [Saranya and Sadhasivam, 2012] It was for this reason that a number of alternative hybrid approaches were experimented with. These hybrid approaches to recommendations of news articles do so by implementing a two-stage process where, e.g., collaborative filtering and content-based filtering are applied in succession. [Abbar et al., 2013, Li et al., 2011b]

The approach outlined here can be seen as a direct adoption of such a two-stage hybrid approach to news recommendations. In this case, the first step is having users label articles, which can be seen as a mix of content-based and collaborative filtering. Articles then get recommended based on their assigned labels to individual users, which is a direct adoption of content-based filtering. The upcoming section now takes a closer look at how the idealized distribution of news articles is translated into an algorithm making use of content-based filtering as wella s the accompanying baseline algorithms.

## 4.3 Recommender Algorithms

### 4.3.1 Baseline Algorithms

The goal of the recommendation algorithms developed in this thesis is to analyze how diverse recommendation influence the reading behavior of users. And in order to analyze the effects of the proposes diversity algorithm on reading behavior of user, it is compared against two baseline algorithm. The two baseline algorithms implemented here focus on accuracy-optimized as well as an unbiased chronological ordering of news articles. The upcoming sections take a detailed look at the details of their implementation and how they are setup in the context of the DDIS News App back end.

Accuracy-Optimized Recommendations: The goal of having an accuracy-optimized user group is first and foremost to see what the potential effects are on reading behavior that arises on the basis of a narrow recommendation scope. This should closely mimic echo chambers or filter bubbles. Doing so required a recommendation algorithm that recommends only those news articles to users that are aligned with their political viewpoints. The previous section outlined how article scores are calculated on the basis of user scores. It insofar deviates from classical approaches to recommender system as they either have user scores or item scores available, but not both of them at the same time before the recommendation process. [Aggarwal, 2016]

Having both score available, however, immensely benefits the development of recommendation algorithm. This setting allows for developing an accuracy-optimized algorithm that achieves optimal results. To do so, the recommendation strategy implemented here simply assigns news articles to users where the difference in terms of their respective political score is minimal. All that is required is to iterate through the list of available articles and find user-article pairs that have the closes match.

The implementation is light-weight, no model needs to be trained, no computation intensive server-side calculations are required. There is, however, one additional aspects that need to be considered. It is the accuracy-recency trade-off. The time window is set to 72 hours. Whenever the recommender system creates a list of news articles for a user in the accuracy-optimized group, it must filter out the articles that were published more than 72 hours ago. As a result of this approach, it could happen that from time to time the closest matching article removed from the recommendation list.

Chronological Recommendations: The chronological baseline ideally features a flat distribution, without any peak at a particular value. Because the closer it resembles any of the other two distribution featured here, the less pronounced the differences in the reading history of the participants become. Unfortunately, the curve of the chronological recommendation group can only be controlled indirectly. The reason for this is that when the group members receive the articles, it is not yet known what the respective article scores are. In the current setup, the score becomes only available once the baseline users have actually read the news articles recommended to them. Ensuring a flat distribution in a direct manner is not a possible option.

One indirect way of controlling the distribution, however, is to calculate average scores of news articles published by a given news outlet. By doing so, it is then possible to either increase or decrease the number of articles forwarded to the chronological baseline group. Complementary to this approach, it is possible to simple try to find more news outlets to collaborate with. In the context of this thesis, however, the actual distribution of the baseline group was of no concern and subsequently not adjusted in any way. It was not required to be flat, for in the preliminary user study conducted here all the user scores were randomized. No further filtering was done and the users simply received all the articles where the ranking of the articles was ties to their publishing date.

The users in all of the three groups receive automated updates. The back end of the DDIS News App scraped the servers of the news outlets every 20 minutes to see if there were new articles published. It then processes the new articles and stores them to the database. The recommender automatically sends updated reading lists to the users in the chronological group. The same holds true for the accuracy and diversity group, although there is a small delay for the recommender system first needs to calculate the user-article pairs based on the most recent reading metrics available.

### 4.3.2 Diversity Algorithm

The task of this section is to now translate this idealized distribution into an algorithm that runs in the back end of the DDIS News App and that works in tandem with the scoring systems for users and articles. The idealized article distribution outlined in chapter 3.2.3 closely resembled a skewed normal distribution with the user's political score its center. The upcoming section now takes a closer look at how to translate this distribution into article recommendations using the previously calculate article scores.

Diversity-optimized recommendations are calculated using a content-based filtering approach, as was the case for the accuracy-optimized recommendations. Unlike the case of the accuracy algorithm, however, there was a lengthy trial period of adopting and testing a number of different approaches to content-based diversity found in the literature. Algorithms featuring random walks and the HeatS approaches were among the possible candidates, both of which provided good results in terms of diversity during earlier studies. [Nikolakopoulos and Karypis, 2019, Liu and Zhou, 2012] Unfortunately, adoption of the techniques was not successful. There are three main reasons for that: data sparsity, algorithm runtime or extensive recommendation adjustment. The following three section will highlight each problem individually. The reason of doing so is not so much explaining why a given technique could not be adopted, but rather to look at what the challenging aspects are that a suitable algorithm needs to overcome.

- Data Sparsity: A recommender needs readily available data on reading metrics of users. For this purpose, recommenders typically create a user-article matrix. In the case of news recommendations, the value of each cell in this matrix is, for example, either 1 or 0, depending on whether or not a given user has read a particular article. The problem of data sparsity describes a situation where there are too few non-zero cells in the matrix for articles that should be recommended. If that is the case, reliable recommendations become difficult, if not impossible. The number of users varies over the time of day. It might be the case that no new accuracy- or diversity- optimized recommendations are output, simply because there are too few users of the chronological baseline currently online reading news articles.

- Algorithm Runtime: News recommendations are very time-sensitive. The time it takes between when an article is published and when it gets recommended to a user should be as short as possible. Unfortunately, many of the currently available recommendation strategies do not put a heavy enough emphasis on the element of time and perform poor under heavy load. Part of this has to do with the fact that the problem space for calculating user-article pairs is of size $n * m$, where $n$ is the number of users and $m$ the number of articles scraped within the time window of three days. The runtime can grow very rapidly when conducting a large experiment with hundreds of participants a hundreds newly scraped articles. However, in order for the DDIS News App to work properly, an ideal algorithm makes sure that the infrastructure currently in place can handle small-scale experiments and large-scale experiments alike, without impacting runtime too much.

- Recommend Adjustment: Recency plays an important role when it comes to diverse political news recommendations. Creating a reliable multi-objective recommender that considers both diversity and recency is challenging. Especially if one considers that many diversity algorithms were developed with an accuracy-diversity trade-off in mind and not a diversity-recency one. As a result of this, there were a number of post-recommendation steps involved for tweaking some the of above-mentioned algorithm outputs to match the desired distribution outlined in chapter 3.2.2. Ideally, however, there are no such steps required.

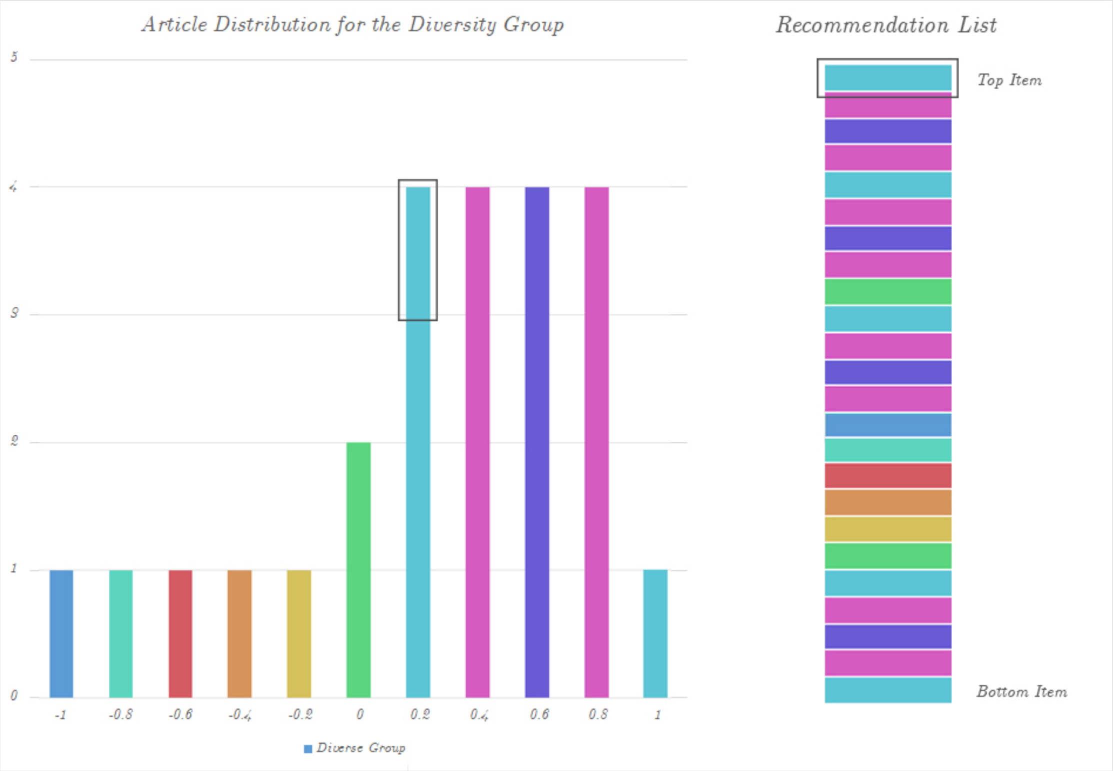


Figure 4.2: Translation of a sample distribution into a recommendation list.

The final algorithm implemented here takes care of all three problems. The problem of data sparsity is tackled by making use of thresholds. The algorithm can, in theory, recommend any news article if at least one user read it. However, the lower the number of users who read a given article, the less reliable the rating. A threshold value of ten was now added, which can be adjusted to make sure the algorithm works despite sparse data while preventing recommendation of articles with unreliable scores.

The problem related to the runtime was solved by creating non-personalized recommendations for groups of users. The political score of a user can have any value in the range between -1.0 and +1.0. However, it might not be very meaningful to calculate separate recommendation lists for users with nearly identical scores. Since it is only the political value that determines what the articles are that one receives, it is possible to put users into broader user groups. There are currently eleven user groups. The score of the first group is -1.0 is then continues in 0.2 increments. The second group has value -0.8 and so forth. A user is assigned to a particular group by rounding their user scores. This makes sure the runtime is decoupled from the number of users.

Discussing how the third and last problem of recency trade-offs was solved directly leads over to the detailed explanation of how the diverse news recommender algorithm work. This was by far the most challenging aspect and deviates in many areas from what are considered more typical approaches recommender systems.

Figure 4.2 illustrates the general approach to how a given distribution is translated into a one-dimensional reading list that gets shown to a user inside the DDIS News App. Please note that the left side of Figure 4.2 again shows the same diversity distribution that was previously features in Figure 3.1. The accuracy-optimized and chronological distribution were deleted from the chart. And the line graph was replaced by columns in order to better illustrate how the reading list is calculated by the algorithm. The implementation proposed here makes use of what can be best referred to as 'spectral slices.' These slice have two purposes. First, they offer a way to of how to resolve the diversity-recency trade-off problem. Second, they allow to fit two-dimensional distribution inside a one-dimensional reading list.

- Adjusting for Time and Diversity: A spectral slice is defined by a value $D$, which is a $n$-dimensional array that captures the article distribution of the slice. The value of $n$ is equal to the number of score groups displayed on the x-axis of the diagram in Figure 4.2. Taking this figure as an example, $D$ would need to contain the following values $D = [1, 1, 1, 1, 1, 2, 4, 4, 4, 4, 1]$. The algorithm now accesses the database and selects for each score group the $m$ most recent articles that are closest to the user's political position, where $m$ is the value stored in $D$ for the corresponding score group. There is no general function that defined $D$ in terms of a given user score. Instead, the recommender system allows for each score group to have its distinct distribution. In the example shown in Figure 4.2, the score group of articles with a political score of $+0.2$ needs to contains two articles. This is indicated by the value 2 at $D[6]$, which is that group's corresponding value in $S$. All articles are assigned to score groups. Their political scores are rounded in the same fashion as users scores are when being put into a particular user group.

- Creating a Reading List: Once all articles are collected from the database, the algorithm then continues creating the recommendation list. It looks for the largest value in $D$ and recommends the articles retrieved earlier, starting with the highest value that is the furthest away from the user's original position. Figure 4.2 has colorized columns for the purpose of visualizing how the diversity algorithm distributes them across the reading list. The colors indicate how far away a particular column from the user score is. By selecting the largest value in $D$, the algorithm finds value 4. This value is located at four different places. The user score being $+0.6$ means that articles of score group 0.2 are selected first, for they have the greatest distance among all columns featuring value 4. This being the starting point, the algorithm then creates recommendations by going through the graph from left to right, top to bottom. Please not the square the is drawn around the top left article in the distribution. This article is now placed on the top of the newly calculated recommendation list, again marked with a square around it. This process works for one or multiple dimensions alike. Furthermore, a recommendation list can have any number of subsequent slices

# 5

# Preliminary User Study

A preliminary user study was conducted in the context of this thesis. The motivation behind doing so was to make sure that all the algorithms work as intended. It also allowed to check that there are no problems related to the integration of the new algorithms into the existing DDIS News App back end. Furthermore, the user study served as a benchmark to test the performance of the current infrastructure and allowed for gathering usage date of the app. This data is useful to further enhance and improve the diversity algorithm, especially for tweaking the parameters for article rating. The upcoming sections serves both as a description of how the experiment was conducted as well as an instruction of how to setup future DDIS News App experiments.

## 5.1 Methods and Experiment Setup

Participants recruited for the survey were asked to use the app on a daily basis during a four-week time window. Both Android and iOS builds of the DDIS News App were distributed via personal invitations. Invites and downloads were all anonymous. In addition to that, users were asked to mask their identity by signing up to the app using a fictitious email address. No verification of ownership of the email address was required. This was to make sure that no personal was processed during the experiment.

And to completely anonymizing user profiles, the political user scores were randomized and not assigned on the basis of survey answers. As a result of that, the scores assigned to news articles were affected by this randomization as well. While it is possible to calculate an article score on the basis of reading behavior of users during the experiment, this score subsequently says nothing about the content of the article itself. This setup was necessary due to legal restrictions and ethical considerations that prevented processing answers on political questions and recommending politically one-sided or diverse news articles to individual participants. It is insofar not a major limitation, as it is still possible to test whether all algorithms work as intended; the values of the scores influence their performance. However, the one thing that could not be evaluated in this setting is what the influences of diverse news recommendations on the reading behavior of user is. There must not be any distinct differences between any of the groups' reading lists.

## 5.2 Scrapers and News Outlets

The goal of this thesis is to write recommendation algorithm optimized for diverse political news. In order for these algorithms to work properly during an experiment, it is a prerequisite that there are enough diverse news articles to recommend. When work on the algorithms started, there was but one news outlet included in the back end of the DDIS News App that was scrapping news articles. One news outlet, however, does not provide enough diverse political articles to test the algorithm with. This meant that the first task of setting up the experiment was to write additional scrapers for a number of new outlets. The news outlet already considered in the back end was *NZZ - Neue Zürcher Zeitung*.[1] Two new media outlets joined the experiment during writing this thesis, namely *Blick*[2] and *Tages Anzeiger*.[3] This allowed for a more diverse selection of news. For both new outlets, a customized version of the existing article scraper was implemented and integrated into the DDIS News App back end.

The scrapers go through the RSS feed of the respective newspapers and copy over the news articles for which there is not yet an entry in the database available. The database used here is the schema-free MongoDB. In case of *Blick*, direct access to their JSON file of each of their news articles was provided. There was no direct access available for accessing the data of *Tages Anzeiger*, making it necessary to parse the news articles' HTML files. However, as previous experience with *NZZ* showed, parsing the HTML files comes with a major drawback. If the news outlets changes even one small title or lead tag in the HTML file, then the scraper needs to be readjusted.

It is important for the experiment that all the articles are white labeled. For that purpose, the app displays only the title, lead, publication date and the actual article itself. News source, author name and other information will be scrapped, but it is not shown to the participants inside the app. The reason behind doing so is to avoid any potential bias towards a specific outlet or author that a participant might have. Unfortunately, there are some limitations to this white labeling. For one, there are a number of prominent columnists that are associated with a specific news outlet. The authors' names may appear in the title, lead or the body of the news article and thus allows for identifying which outlet published the article. There was only a small number of articles affected by this during the preliminary study. However, for large experiments it might be necessary to filter such content that allows for inferring the article publisher.

Article filtering is a general concern and needs careful consideration in small- as well as large-scale experiments. During testing sessions prior to the user study, there were a number of remarks regarding article duplication. News story duplication to be more exact. One specific story or event can be covered by a number of news outlets. As a result of this, there can be number of closely related articles on one and the same event that took place. This is not necessarily to the liking of participants. One way of solving this problem is to introduce additional filter criteria for news articles.

[1] Website of NZZ - Neue Zürcher Zeitung: https://www.nzz.ch/

[2] Website of Blick: https://www.blick.ch/

[3] Website of Tages Anzeiger: https://www.tagesanzeiger.ch/

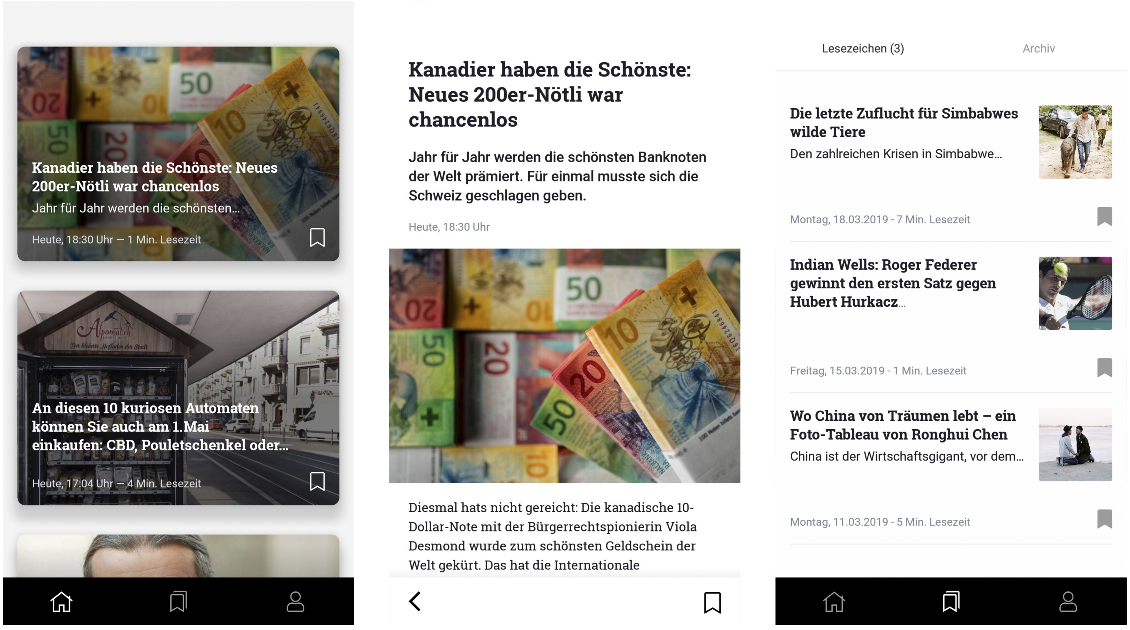
One filter adopted here focuses on sorting out articles that are based on press agency reports. Many of the above-mentioned duplication of reports are caused by news outlets importing what the national agency has written. The filtering is done via the author labels that news articles have. If a label mentions a press agency as author/source of an article, then it gets sorted out. The current solution only accepts news agency articles if they are published by *NZZ - Neue Zürcher Zeitung* and ignores any imports received from either *Blick* or *Tages Anzeiger*. The reason behind this choice was that the two new scrapers were already developed with filter criteria in mind and thus could easily accommodate for this extensions. By having a dedicated filter in place, it allows for some control of duplication. This is by no means a perfect technique. If two or more news outlets write and publish their own articles on an event, then this escapes the filter implemented here. A semantic-based approach would be necessary to do so, requiring content pre-processing of news article.

There are a few other filter criteria in place. For example, sponsored content is completely ignored. No advertising whatsoever is featured inside the app. This extends to certain news categories such as, e.g., *Cars*, where many newspapers regularly post reviews of new car models. Another category that was equipped with a filter was *Sports*. Some news outlets publish regular updates on the newest standings during sport events. As a result of this practice, the app can get flooded with sports articles, especially on weekends. To prevent this, sports news is sorted out. This is done via a category label that each article comes with when they are being scrapped from the news outlets. Again, since there is no semantic processing of articles, it may happen that a sports article gets passed this filter and recommended if, e.g., it was published under a different category like *National* or *International News* instead.

## 5.3 News App Overview

The algorithms developed in this thesis are tested using the DDIS News App. The DDIS News App was developed prior to starting this thesis. The app serves as a framework that allows for plug-and-play of different recommender algorithms. The news app is written using the React Native framework for the front end. The user experience remains the same across all platforms and versions. Platform-specific features were purposefully omitted in order not to introduce any bias. The following section will provide a brief overview of the app and explains how the recommendation list is presented to the users.

Figure 5.1 provides an overview of how news articles are presented to users. The recommendation list is shown on the left-hand side. This is the home screen of the app. User can scroll down and see what the current recommendations are. Each news article features either an image or a gray placeholder background if there is no graphic available, the article title and a short lead. In addition to that, the publish date is visible, as well as an estimate of the reading duration. The home screen gets update automatically in intervals of 20 minutes. All screen looks the exact same across all platforms and versions, save for the aspect ratio as determined by the device's screen.



Kanadier haben die Schönste: Neues 200er-Nötli war chancenlos

Jahr für Jahr werden die schönsten...

Heute, 18:30 Uhr — 1 Min. Lesezeit

An diesen 10 kuriosen Automaten können Sie auch am 1.Mai einkaufen: CBD, Pouletschenkel oder...

Heute, 17:04 Uhr — 4 Min. Lesezeit

Kanadier haben die Schönste: Neues 200er-Nötli war chancenlos

Jahr für Jahr werden die schönsten Banknoten der Welt prämiert. Für einmal musste sich die Schweiz geschlagen geben.

Heute, 18:30 Uhr

Diesmal hats nicht gereicht: Die kanadische 10-Dollar-Note mit der Bürgerrechtspionierin Viola Desmond wurde zum schönsten Geldschein der Welt gekürt. Das hat die Internationale

Lesezeichen (3) Archiv

Die letzte Zuflucht für Simbabwes wilde Tiere

Den zahlreichen Krisen in Simbabwe...

Montag, 18.03.2019 - 7 Min. Lesezeit

Indian Wells: Roger Federer gewinnt den ersten Satz gegen Hubert Hurkacz...

Freitag, 15.03.2019 - 1 Min. Lesezeit

Wo China von Träumen lebt – ein Foto-Tableau von Ronghui Chen

China ist der Wirtschaftsgigant, vor dem...

Montag, 11.03.2019 - 5 Min. Lesezeit

Figure 5.1: The app's home screen (left), article view (center) and reading list (right).

The center section of Figure 5.1 shows the detailed reading view of an article once the user has made their selection on the home screen. They are able to read the full article in this view and give a like or dislike at the bottom. In addition to that, there is a bookmark icon in the bottom right-hand side corner. When bookmarking an article, it will be put onto a reading list where users can save interesting news that they want to read at a later point in time. The article in the bookmarked reading list will not be lost once the recommendation list gets updated. The right-hand side of Figure 5.1 shows this bookmark list in more detail. Here, users are able to browse and remove items from the list. In the top, right next to the bookmark list, the archive view can be selected. It is presented in the same way as the bookmark list. By browsing the archive, users get an overview of previous items features in their bookmark list.

## 5.4 Data Collection

Chapter 4.2 outlined the reading metrics that the recommender systems stores. These metrics will be collected whenever the DDIS News App is used. When conducting an experiment, there are a number of other datapoints that the app can collect. Using the DDIS News App requires an account. Each participants creates their account when launching the app for the first time. And whenever a user fills in an in-app survey, all the answers and questions get stored under their account name. This allows for processing a wider range data. For example, one piece of information that is assigned to a user account is political store calculated on the basis of the intake-survey.

The intake-survey, however, is only one of potentially many surveys that experimenters can ask users to fill in over the course of an experiment. One such additional survey that was created in the context of this thesis is a recruitment-survey or a screener for possible participants. It can be used for recruiting people from across the political spectrum. During the preliminary user study, user scores were randomized. No screening of participants was required. However, during a normal experiment, it becomes important to have participants from across the political spectrum. A diversity of political user scores is a prerequisite for diverse political article scores. If that is not a given, then the recommender system will have suboptimal performance.

Appendix A.2 features the complete list of questions asked to potential participants in the screener. It can be implemented as an in-app addition to the intake-survey or a separate paper-based form. By making use of the screener, information on the age, gender, media consumption habits, political position etc. are collected. Not only is this useful for selecting a user base representative of the population, but it also allows for a more detailed look at the dependencies that might correlate with a change in reading behavior caused by diverse news recommendations.

## 5.5 Evaluation Strategies

Evaluation of the algorithm's performance is challenging. There are automates means available for doing so. [Han and Yamana, 2017, Hijikata, 2014] They are often very limited in scope. Two popular ways of assessing the quality of diverse recommendations is to calculate the diversity level of items for a given list and then check how many of the recommended items were accessed by a user. [Hijikata, 2014] There are, however, two problem related to automated approaches for offline evaluation in this setting. First, methods that operate in this way disregard the temporal aspects that users have but limited time available for using the app. Second, there is no point in measuring the item diversity for a given list when using the algorithm outlined here. The level of diversity will always match the distribution that was input. Measuring the level of diversity makes only sense in a setting that does not allow for direct control of the final distribution. It is for this reasons that no automated means of algorithm evaluation are adopted.

The best way to evaluate the algorithm is by interviewing participants and asking them to assess the quality of the recommended. Interview protocols for doing so were already created, see A.3 in the Appendix for more details. User satisfaction and their experiences with the app are documented in detail. However, satisfaction is not all there is to a successful diverse news recommender. Diverse recommendations should ideal provoke the users to a certain extent. But this comes at the risk of there being a number of recommendations that a given user might dislike. This is why there is also a more quantitative way of how the algorithm is evaluated. Since the main goal is to look at changes in behavior, it is first and foremost metrics like time spend reading news, the number or articles accessed etc. that matter most and capture best the user experience. It is also these metrics that can ultimately be used to convince platform owners to shift their focus from accuracy- to diversity-optimized algorithms.

# 6

# Processing Results

Due to legal restrictions and ethical considerations is was not possible to use the recommender algorithms in its full capacity. User scores were randomized during the preliminary user study. As a result of this, the political scores assigned to users and news articles do not carry any meaningful information. They are completely ignored here. Despite this limitation, however, the data gathered during the preliminary user study is crucial to further improve the performance of the diversity algorithm. The datapoints allows for a fine-tuning of the parameters used to calculate the recommendations lists. Furthermore, general user feedback was gathered during the study to improve the user experience. The study also allowed to check whether there are any problems related to a specific platform or version of Android/iOS. The data is based on the activity of a total of 17 participants, details can be found under A.4 of the Appendix.

## 6.1 Reading Metrics and Indirect User Feedback

The first part of the evaluation of the results looks at the reading metrics and the indirect user feedback. This includes reading time and login time. Reading time stores how long a user read an article as well as a timestamp. Login time stores at what point in time users accessed the app. Both reading and login time are analyzed separately. The reason for this is that not every time users open the app they actually reading articles. Testing prior to the user study showed that it is a common habit for users to simply open the app, glance over the headlines without selecting any of the recommended news articles. This information could not be captured if only the reading metrics were considered during the evaluation.

Reading Time: Figures 6.1 shows at what times the participants read the most articles. In general, they read in the morning between 08:00 - 10:00 and between 16:00 - 20:00 in the evening. Users reported that they were frequently using the app when commuting. No user activity was recorded between 03:00 - 06:00. During the day, the lowest activity can be found between 11:00 - 12:00. Figure 6.2 shows reading activity by day of week. When looking at reading behavior across the week, it is Tuesday when users are most active, followed by Saturday and Monday. The weekly low is on Sunday, where participants show barely any activity.

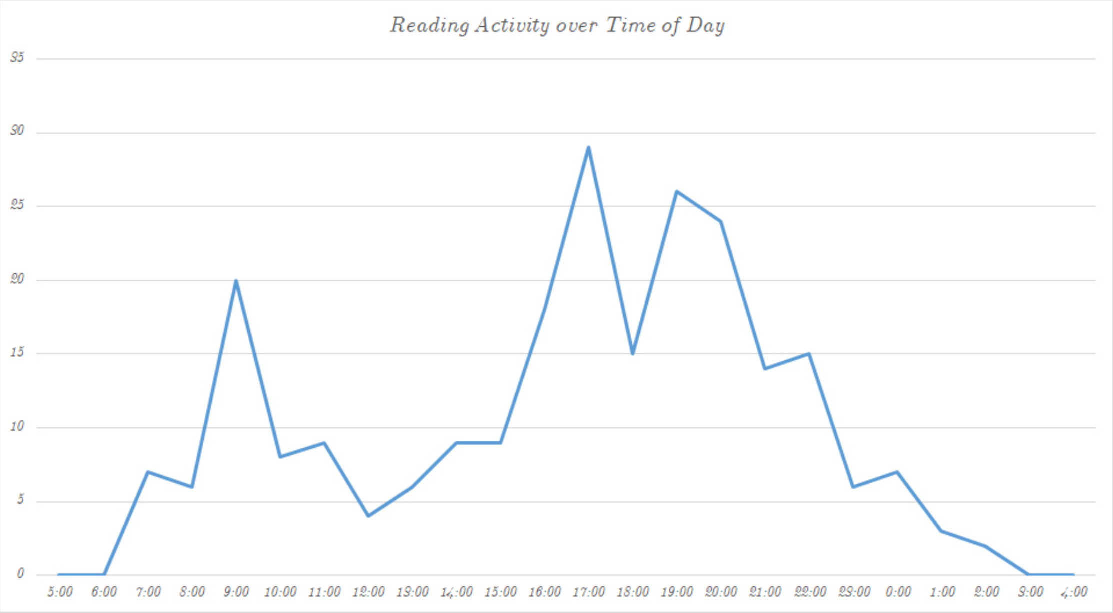


Figure 6.1: Total number of read articles (y-axis) for each hour of the day (x-axis).

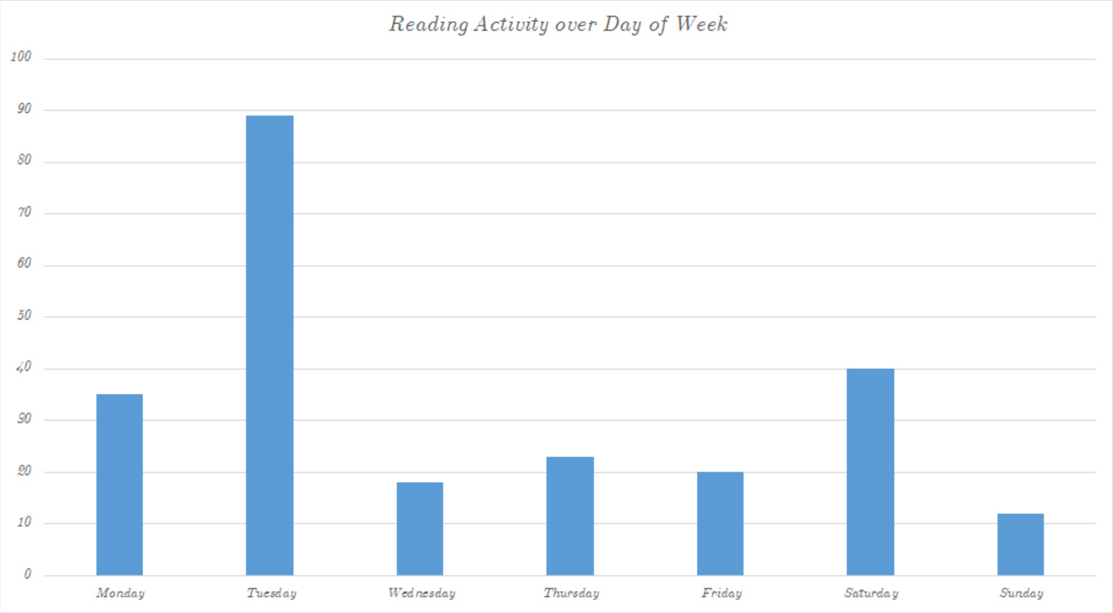


Figure 6.2: Total number of read articles (y-axis) for each day of the week (x-axis).

One possible explanation of why Monday and Tuesday show the highest activity of all business days is that participants wanted to catch up to news they missed during the weekend. A lack of news articles to read can be excluded as having an influence on the activity that Figures 6.1 and 6.2 show. The scrapers were active at all times, so was the recommender system updating the reading list. In addition to that, users had full access to the entire news archive featuring several thousand news articles.

Login Times: Figure 6.3 shows the number of logins the app registered for a given hour. Looking at login times across the day show a number similar features as well as a few distinct differences between the login and reading metrics. The first thing that is worth mentioning is that there is less of a pronounced peak in activity between 08:00 - 10:00 in the morning. Peak activity during the early to late evening remains relatively unchanged; again lasting from 16:00 - 20:00. However, there is a difference in activity during the night. It looks like many participants kept checking the app late in the night up until 01:00 o'clock. However, almost no one read articles during that time, as indicated by Figure 6.1 showing the reading activity across the day.

Figure 6.4 shows the login activity over the day of week. Looking at the activity over the entire week draws a picture almost identical to what was the case for reading time in terms of Tuesday being the most active day and Sunday the least active one. There is, however, one difference worth highlighting. The second busiest day in terms of logins is Wednesday, which before was the least active business day with reading metrics. Unfortunately, this difference may have been primarily caused by the recruitment process. A number of participants were asked to join the experiment after it had already been running for two weeks. Their recruitment happened on a Wednesday where they all downloaded the app and then created new user account.

During the experiment, users read a combined total of 237 news articles. The DDIS News App stores for each of the accessed articles how much time a user spent reading it. Figure 6.5 shows the distribution of articles sorted by reading time. The graph mimics a long-tailed distribution. Most of the articles that user read capture their attention for 20-50 seconds. The average reading duration per article is 84 seconds. One thing that was dropped from this figure is the articles that users accessed for less than ten seconds. The reason for ignoring these metrics is that when looking at an article for less than ten seconds it is highly probable that the article was selected simply by mistake. Taking these measurements into account would not benefit the overall evaluation.

Looking at this data combined with the login time reveals a particular challenge that later experiment may face. Users seem to have a general aversion against long reads. In the current version, the reading time is listed next to the lead of each article. If an article takes more than three minutes to read, however, it is less likely that it gets selected. The current diversity algorithm puts a heavy emphasis on the ordering on articles. The underlying assumption is that the higher up an article is in the recommendation list the higher its chances of being selected. However, the time estimate might be something that is more relevant to users than the ordering of the article.

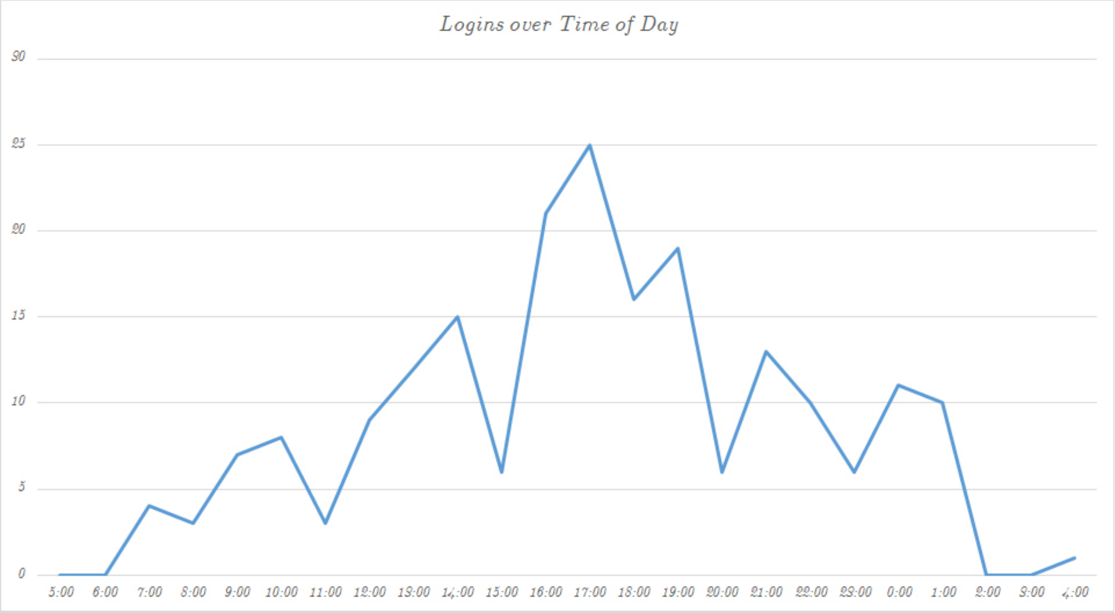


Figure 6.3: Total number of logins (y-axis) for each hour of the day (x-axis).

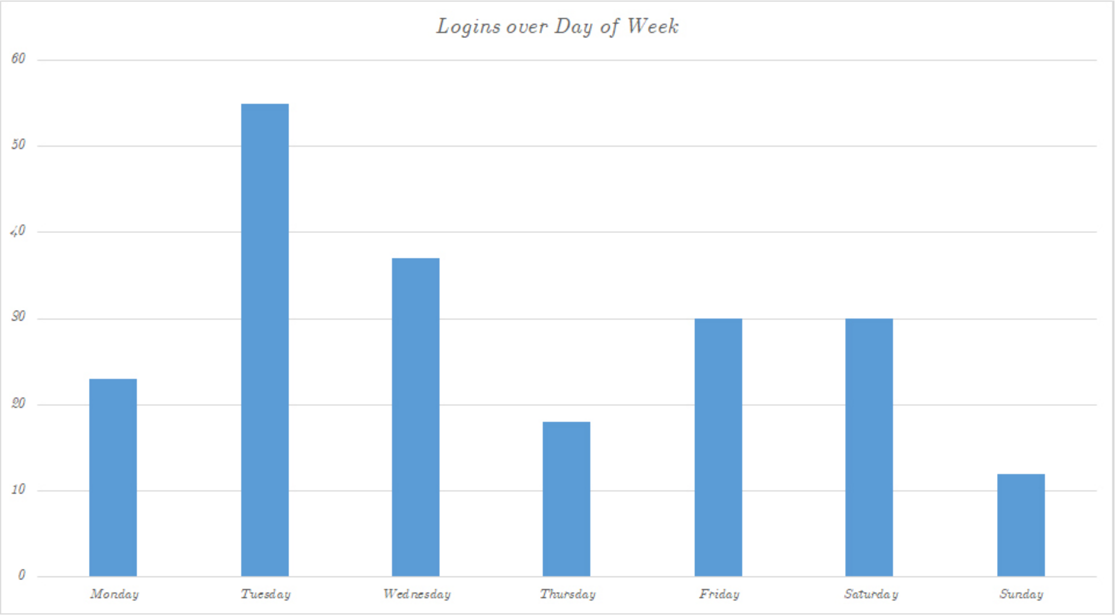


Figure 6.4: Total number of logins (y-axis) for each day of the week (x-axis).

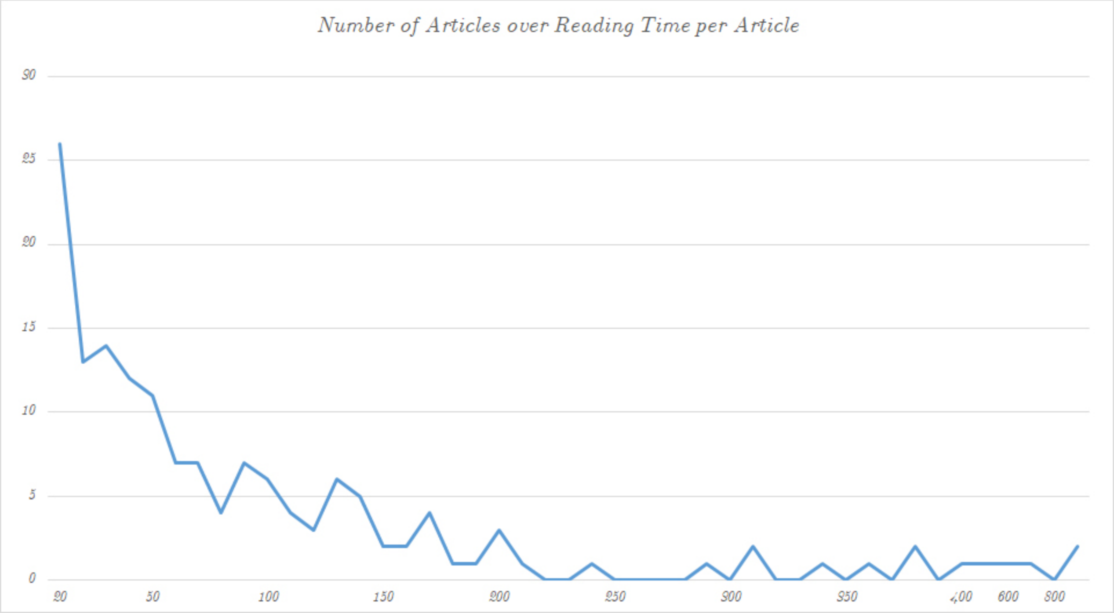


Figure 6.5: Number of read articles (y-axis) over reading duration in seconds (x-axis).

## 6.2 Article Ratings and Direct User Feedback

Likes, dislikes, reading list and article archive are the types of direct user feedback that the app keeps track of. While the rating system, the reading list and article archive were explained to participants, they were not specifically asked to make use of these features. The goal was to see to what extent they themselves make use of these features and see them as offering a certain benefit or improving the user experience. Unfortunately, the participants made only modest use of lists and article ratings.

The most popular feature was giving a like to an article. A total of 73 interactions were recorded. Adding articles to the reading list was the second most prominent interaction. However, only a total of 35 articles were added to reading lists. Adding or removing articles from archive and giving dislikes were the least popular features that users made use of. Only 18 articles were accessed in the archive and only seven news articles received a dislike over the course of the entire experiment.

The reason for why there were hardly any dislikes can be found in the reading metrics. There is a large number of articles that users had open for less than ten seconds. In other words, instead of disliking an article, users are more likely to simply close the article again and look for another article to read. And the reason for why the article archive way not used more frequently can be found in the access frequency of articles. There were only 19 cases where users have accessed articles multiple times. In general, there is no interest in reading an article multiple times

## 6.3 General Feedback on User Experience

Creating an enjoyable user experience is key when it comes to creating an app that participants should be motivated to use on a daily basis. It is for this reason that gathering information on the usability and other non-technical aspects presents a crucial part of the user feedback. While the experiment was ongoing, participants suggested a number of possible changes to the app in order for it to be more user-friendly.

One feature that was frequently requested is being able to filter certain types of news articles. This is currently not possible as the app uses user groups as the basis of its recommendations. It created non-personalized recommendations. For any two users sharing the same political score, the algorithm will return the exact same recommendation list. Users also complained about way in which articles are displayed within the app. Due to copyright reasons, it is not possible to store any pictures in the database together with the scraped news articles. Only the URL to the image is stored. When using the app, text and image information arrive from two different sources. Unfortunately, this can result in situations where text and image information arrive at different times. News articles can appear in the app while the image is still loading. If that is the case, then the article preview will feature a grayed out background image. Some participants of the preliminary user studies remarked that this prevented or stopped them from scrolling further down, because they thought the recommendation list has ended. They usually focused only on reading the first 30-50 news articles recommended to them.

Closely related to this is the general issue of news article images. While outlets like *Blick* and *Tages Anzeiger* provide an image for all the articles, *NZZ* articles may lack an image. And with the inclusion more news outlets in the future the number of news articles without a picture assigned to it will likely grow. Users commented that while scrolling through the news list they are more likely to read an article where there is an image. Some even categorically ignore articles where there is no image available. This poses a problem, for it introduces bias towards certain news articles that have nothing to do with their content, place in the reading list or their associated political score. To solve this problem, there are three different solutions that were discussed with participants that had an issue with how the app handles the lack of available article images.

- First, the scraper can be rewritten to simply stop scraping news articles where there is no image present. While easy to implement, it is preferable not to reduce the number of articles as their diversity could be affected. Another negative consequence is that this approach entails that news outlet where not every articles has an image assigned tends to be underrepresented in the app.

- The second option is to remove the images from all articles. That way there is no bias introduced regarding what articles users are more likely to read. Removing the images altogether, however, was an idea generally not well-received by the participants. While this solution is relatively easy in terms of its implementations, users very much like that pictures are an included feature of the app.

- The third and last approach is to extend the scraper in such a way that it adds a fitting image to news articles if there is none present. One way of doing so is to assign a generic image depending on the categories of an article or depending on its content. The information on the category is already stored in the database. However, doing would also entail that a number of articles all share the same image. This in return could also lead to them being simply ignored. Alternatively, the scraper could browse the Internet, looking for images related to the articles content. In order for this to work, however, an extension of the current solution is needed. There is currently nothing implemented that would allow for automated content-based processing of news articles.

# 7

# Discussion

The inputs received from the preliminary will be used to tweaked the app and make it more user friendly. With students generally representing a younger and more tech-savvy part of the population, however, there might still be a number of potential improvement for future versions when the app gets used by a broader audience. The selection of the participants for the preliminary study was limited in scope and conducted with the help of fellow students of the university as well as friends and family members. This is insofar not a problem as the goal of the study was to test the core functionality of the recommender system, scraper and app. Any future deployment of the recommender system, however, must make sure to select participants in such a way that their political viewpoints are from across the entire political spectrum.

Articles being rated on their readership implies that the range of political article scores is limited by the available political scores of users. Having no good distribution of participants across the entire political spectrum might introduce a problem to the article recommendations. For well-distributed user scores without clustering around a certain value, it is necessary to recruit people with a plurality of viewpoints using a screener. Ideally, the participants recruited for a large-scale experiment differ not only in their political views, but also in terms of their age, gender and income to name but a few of the most relevant criteria.

Further improvements can be made regarding the setup of the experiment. Something that is relevant for all upcoming participants is a proper introduction to the app. In the preliminary user study featured here, some participants simply received a link or invite to install the app together with a description of the app's feature. This might have led to a situation where some were unsure or even unaware of the possibility the app offers. It would also explain the modest use that direct user feedback saw during the experiment. Alternatively, it might become necessary to make it mandatory for people to rate article during the experiments. Direct feedback immensely benefits the rating process of articles and is crucial for when non-randomized scores have to be calculated.

Participant might also have to be asked to use the app more regularly and at different times during the day. The user study showed that there are large activity peaks during the morning and later afternoon/early evening. Users reading outside these daily peaks, however, might receive recommendation lists that feature too few new articles as a result of low levels of activity.

In addition to that, users show a habit of not reading longer articles and simply browsing headlines, ignoring the ordering of the recommendation list. Unfortunately, this may have something to do with there being but a limited time window within which participants make use of the app. Many reported using the app during commuting instead of purposefully allocating some time during the day to read news. People also show a habit of quickly closing a selected article. This might be caused by the article not being to their liking or too long of a read. The reasons behind doing so were not further investigated in the context of the evaluation of the preliminary survey. Ideally, however, they read all recommendations presented to them.

# 8

# Limitations

The current setup can only answer the question of how diverse, chronological and accuracy-based news recommendation influence the reading behavior of users. The recommender system is currently unable to answer the question of what a good diverse distribution of news article across the political spectrum looks like. In order to answer this question, two or more diverse recommendation algorithm need to be compared with one another; it is not sufficient to simply compare one diverse recommendation algorithm against two baseline algorithms. Fortunately, while there is currently but one diverse algorithm implemented, the way the recommender system is set up allows for any number of different recommender algorithms to be added. The easiest way of doing so would be copying and modifying the current diversity algorithm and tweaking its parameters to match a new distribution of news articles across the political spectrum.

While adding another algorithm to the recommender system is in large parts fully automatized, there still remains the challenge of adjusting all the algorithm's parameters. As previously discussed in the Chapter 4.3, it is crucial to adopt the recommender system to the particular political system it is used in. The reason for this is that the system is highly dependent on the preliminary survey successfully capturing the intricacies of the political landscape. Similarly, the key to meaningful recommendations is to have a good selection of news outlets and participants from across the entire political spectrum. The following sections further discuss this high dependency on the political landscape as well as the dependency on news outlet and participants in greater detail, focusing on the difficulties and challenges associated with them.

High Dependency on Political Landscape: As far as the adoption of the proposed system is concerned, it is important to mention that the rating procedure, i.e., the survey questions and evaluation of answers for calculating a political position for each user, is very specific to a particular political landscape. The information that goes into creating such a survey must be gathered with a particular political system or landscape in mind and the different questions rely heavily on past elections and official statements of political parties. In other words, the survey from *Polittools* used here is only suitable for assessing the political score of people in the context of Swiss politics. If this system were to be adopted for use in a different country, then the questionnaire would need revision in order for the political scores to be reliable.

In addition to taking into account the different political systems the recommender system can be used in, it is also crucial to keep in mind the time window in which the survey was created and then later used. The agendas of political parties change over time. So do the definitions of what, e.g., a left-wing or right-wing viewpoint are regarding a particular subject at hand. This is especially true around the time of parliament elections, where parties try to incorporate new goals into their programs in order to attract more voters. Without going into the details of how these positions are assessed in the first place, it is important to mention that *Politool* does indeed look at the different endorsements and recommendations political parties make over time and adopts the survey accordingly. Not only is this necessary for any recommender adoption outside Switzerland, but this also implies that the current solution needs to feature a new or slightly updated survey at a particular point in the future.

While on the topic of assigning political scores, it is worth mentioning how to meaningfully interpret the scores if one were to make use of them in subsequent research. It is important to keep in mind that the political score assigned to an article is not comparable to the score assigned to a user. Article scores do not generally allow to make a statement about the content of an article. The score only indicates what the average political preferences are of users that have read the article in question. The political score of an article says something about the reader and not the article, as opposed to the user score, which is directly derived from the participants' answers.

This is a crucial and important distinction. The score saying something about the content itself would only be possible in a situation where participants in the baseline group would exclusively read articles that align with their political preferences. As previous studies suggest, however, this is not the case. Users can show a tendency to read news articles even if they are not in line with their political preferences or even articles that oppose their own political views. [Flaxman et al., 2016] The extent to which this is the case depends on the particular user, how novelty seeking or explorative their reading behavior is. This reading behavior distorts the political score assigned to news articles and in turn limits what can be derived from article scores.

In general, there might be a highly subjective element present in terms of what a reader likes or dislikes, an element that is not correlated with the political score assigned to them. In preparing the preliminary study and setting the diversity level, i.e., defining the distribution of articles, it was suggested to establish a baseline of how diverse, explorative or novelty seeking users' individual reading habits are before they take part in the experiment.[1] Having this data available, it would allow for personalized adjustments of the respective reading list of the users by tweaking the distribution of news articles across the political spectrum. The current system, however, does not feature any option of inputting such data recorded prior to the experiment. In addition to that, it does currently now allow for individual tweaking of recommendation list. Users with the same political score get the same recommendations.

[1] Based on a discussion with Natali Helberger, professor of Information Law, with a special focus on the use of information, at the University of Amsterdam.

High Dependency on the Selection of News Outlets and Participants: All three newspapers that were part of the preliminary user study, *Blick*, *Neue Zürcher Zeitung* and *Tages Anzeiger*, publish exclusively for the German speaking regions of Switzerland and they are all located in the city of Zurich. French, Italian and Romansh speaking parts of the population are not yet covered. Furthermore, the three news outlets are not fully representative of the entire media landscape of Switzerland. With over 1,400 different newspapers, magazines and other publications available this is but a small fraction of the news outlets that people have access to and use on a daily basis.[2]

In order to provide users a diverse selection of news articles from across all the political dimensions, it is crucial to have a wide range of news outlets participating the any given experiment, outlets that are all sympathetic to different political ideas or parties. The greater the variety of news outlets, the broader a diverse distribution of news articles can be. The benefit of a broader distribution is that the effects of diverse news recommendation are likely to be more pronounced. For if there are only a few news outlets that all cater towards the same political audience, there is only little room to diversify recommendation lists. The reason for this is that all articles scores are likely to be centered around a handful point in the political spectrum. As a result of the articles not being well-distributed, the diverse recommendations and baseline recommendations will be relatively similar to one another.

Having a selection of news outlets from across the political dimensions is but one part of conducting a successful experiment. Bearing in mind that news articles eventually get rated on the basis of their readership, a selection of participants from across the entire political spectrum is of equal importance. Any future deployment of the proposes recommender system ideally makes sure to sample their participants in a manner that ensures the resulting list of participants is representative of the population in terms of age, gender, ethnicity and income to name but a few attributes. Without casting a net as wide as possible when trying to recruit participants for the study, problems similar to a limited selection of news outlet might arise. Despite there being a diverse selection of news outlets, if all participants have similar political preferences then article scores again are likely to center around one specific point in the political spectrum. This will again result in the diverse and baseline algorithms creating recommendation lists that are too similar to one another for there to be pronounced differences in the reading behavior of the participants during the experiment.

[2] Official statistics on media publications available in Switzerland published by 'Schweizer Medien' for the year 2018: https://www.schweizermedien.ch/zahlen-fakten/branchendaten

# 9

# Future Work

The goal of this thesis was to develop a diverse recommender algorithm that could be embedded into the DDIS News App. It is for this reason that the subsequent section will focus exclusively on future work that is related to the app's recommender located there. The first useful improvement is technical in its nature and has to do with improving system performance. The back end of the app consists of a virtual instance running on a server with only using a CPU. Having dedicated hardware, e.g., GPUs to offload some of the work, would significantly benefit the recommender system since it makes use of Google's TensorFlow library. Recommendations list, for example, could be calculated more frequently and contain more news articles. In addition to that, full code parallelization for calculating recommendations and the strict use of non-sparse matrices could further speed up the process with which users get updated recommendation list. This has not yet been implemented yet due to the adoption of large parts of a pre-existing code base of the recommender system that had to be adopted as is.

Apart from these methods focusing on reducing the time it takes to run the recommender, there are three possible additions to the current solution that can lead to significant improvements of the recommender system as a whole: content analysis, personalized news recommendations and optimization of the article distribution. It is worth pointing out that content pre-processing, personalized news recommendations and an optimization of the diversity distribution are all elements that can be introduced and used separately. These elements are not tied to a specific recommender algorithm and can work hand in hand with both diversity and baseline algorithms.

Content Analysis: In order to get less redundant articles and more news with an emphasis on political topics it is can be beneficial to implement a content-based filtering of news articles. The news app aggregates articles from various outlets. As previously discussed in the Chapter 5.2, the news article scrapers ignore non-text articles (e.g., excluding video coverage due to technical limitations and legal requirements) and ignore a few selected news categories (e.g., sponsored content). Apart from this selection criteria, however, there is no filtering of news articles in terms of topic, content etc. While this significantly speeds up the scraping process, it leads to the unfortunate situation that there are a number of similar articles on one and the same news story but from different outlets. This redundancy impairs the user experience, especially for users in the control group that get all news articles recommended.

Furthermore, news articles featured in the reading list of users are only to a varying degree related to a topic of political interest. With pre-processing in place, however, the scraper could focus on articles that convey content that is relevant and lends itself well to political labeling. Without any content pre-processing in place, as is currently the case, each and every news article gets assigned a political score indicative of its readers indiscriminately of the particular article topic. For example, an opinion piece on the current monetary policy of Switzerland gets assigned a political score in the same way a brief weather report of the coming weekend gets assigned a political score.

As a result of this, users might spend less time reading articles that would be effective for discriminating users' political positions. But while there are benefits that come along with pre-processing, there is also a negative impact such a system has. The main argument against content pre-processing it its negative performance impact. The scrapper currently needs only to visit a URL and then to copy the content available there. With automated content pre-processing in place, a series of complex operation is needed in addition to scraping the article. An alternative to this automated pre-processing of articles is to have a handful of experts evaluating and rating news articles. However, the downturn here is that only a reduced number of rated articles can be processed as well as an increase in the time it takes between an article being published and presented to the user inside the app. Furthermore, in the case of there being an additional dimensions that are included in the survey, it might be very challenging to find experts that can reliably assign scores to articles on all levels.

Personalized Recommendations: Recommendation algorithm such as the diversity algorithm used here create non-personalized recommendations. It is non-personalized because the political score is the only thing that is taken into account when calculating the recommendation list. This score can be shared by two or more users. If the score is the same, so is the recommendation list. As outlined in the Chapter 4.2, this has mainly to do with optimizing performance and decoupling the runtime of the recommender system from the numbers of users that access the system. However, as mentioned in the evaluation of the preliminary user study, one of the requested feature was being able to customize the recommendations presented by the algorithm.

While not implemented in the current version of the recommender system, personalized news recommendation can be included into the system and do not pose a technical challenge per se. Unfortunately, there is a problem non-technical in its nature and not that easy to solve. Ideally, users read all the diverse articles presented to them. They accept the level of diversity that the algorithm dictates and continue using the app regardless of whether or not there are a number or news articles that they disliked. For some people this number of articles may be larger than for other, but developing and testing an algorithm to recommend news articles from across the entire political spectrum necessarily leads to a recommendation of a news article that a participant deems unfit. It is a delicate trade-off between what researchers want to look at and what is acceptable for a participant to on a daily basis. It is currently assumed that the same level of diversity is suitable for each and every participant. Personal preferences are not taken into consideration in the current setup of the recommender.

With this setup, however, it could happen that a participant already exhibits a reading behavior prior to the experiment that is more diverse than what the diversity algorithm recommends. For this reason, it might be beneficial to allow for individual levels of diversity for personalized recommendations, requiring some knowledge of the past reading behavior of participants. Being able to input how broad or narrow one's past reading behavior is could be used as an initial value of what the distribution of news article should look like. Given an experiment runs over a large enough period of time, it would then possible to start with a narrow distribution that gets broader over time.

By introducing a diversity distribution that changes over time, the experimenters could mitigate the risk of a participant not actively making use of the app because of articles they might dislike. This way would provide a means of personalizing the recommendation list to a certain extent, while at the same time circumventing the need of implementing content filters to customize the recommendations list. However, the fear associated with these filters is that given the option to adjust or tweak the recommendations, over time it would result in a situation where the customized diverse recommendations look too similar to the result of the baseline algorithm focusing on accurate result. In other words, it would no longer be possible to compare diverse recommendations with the baseline recommendations for their distributions are too identical to one another.

This is not to say that any short of personalized filters should categorically be excluded from the recommender system. The case against personalized filters in the current scenario is that they are relatively broad. If anything, then categories of news outlets could be used to act as the basis of a filter. Sports news, for example, could be one category that a participant wants gone from their recommendation list. However, since there are no labels or categories displayed in the app, one might accidentally filter out, e.g., the *International* category because there was a sports-related news story that they disliked. Ideally, personalized diverse news recommendations are implemented hand in hand with content pre-processing. The reason for this is that content pre-processing can allow to establish reliable fine-grain filter criteria needed here for customizing reading lists.

Distribution and Diversity Optimization: In the end, the goal of the app that feature the algorithms implemented here is to answer the question how the reading habits of users change over time if they are presented with different types of news recommendations. A diverse algorithm is compared to an accuracy-only baseline as well as a temporal baseline. This setup allows to look at the trade-off in terms of accuracy and diversity and the influence on reading behavior that follows from putting an emphasis different article distributions. However, this does not yet allow to answer the question how a good diversity algorithm looks. For that purpose, it is necessary to compare different diversity algorithm with one another, each of which implements a different distribution of article scores over the political spectrum. The distribution of news articles chosen here for the diversity algorithm closely resembles a normal distribution with the political score of users at the center. It is but a first example of what a diverse distribution could look like. As stated earlier, the reason behind doing so was to create an algorithm that recommends articles that have a distribution of political scores that fall in between the and features overlapping article recommendations for all users.

# 10

# Conclusions

The goal of this thesis was to implement an algorithm for diverse news recommendation that can be used in future experiments in combination with the DDIS News App. A preliminary user studies successfully tested the stability and reliability of the app with the diversity and baseline algorithms embedded into the existing back end. A recruitment, intake and exit survey were prepared. These surveys ensure that the participants of the experiment are representative of the population and have diverse political viewpoints. Furthermore, two additional scrapers were added to the back end of the app in order to have a greater variety of news articles available. With the algorithms and scrapers in place, the DDIS News App is now setup for experiments in the context of the political landscape of the German-speaking part of Switzerland.

Unfortunately, due to legal restrictions and ethical considerations no personalized data could be processed during the preliminary user study. Despite this limitation, however, it was possible to evaluate the output recommendations in terms of their distribution across the political spectrum for both the baseline algorithms as well as the diversity algorithm. The data generated during the study was used to fine-tune the diversity algorithm's parameters in order to improve the quality of the recommendations and to make the app more engaging to use. In addition to ensuring that the system is working on a number of different mobile platforms and the server infrastructure, the study also provided valuable insight into user's reading behavior.

One particular insight that is relevant for the future development of the recommender algorithm and the app is that users generally prefer to read short articles in the limited time they have available to use the app. The current solution did not consider this aspect and solely focused on the ordering of the news articles instead. Short reading times do also entail that the updating of reading lists needs to happen more frequently. Them regularly checking the phone and not reading any articles is a strong indication of there being too few update, especially during the afternoon. However, since updates are only possible if new articles become available, increasing the number of participating news outlets might become a necessity. This would also allow for a broader selection of news articles, further benefiting the cause of the app.

# References

[Abbar et al., 2013] Abbar, S., Amer-Yahina, S., Indyk, P., and Mahabadi, S. (2013). Real-time recommendation of diverse related articles. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1–12, Rio de Janeiro, Brazil.

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Towards the next generation of recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

[Aggarwal, 2016] Aggarwal, C. C. (2016). *Recommender Systemsl.* Springer International Publishing AG, Cham.

[Aljukhadar et al., 2013] Aljukhadar, M., Senecal, S., and Daoust, C.-E. (2013). Using recommendation agents to cope with information overload. *International Journal of Electronic Commerce*, 17(2):41–70.

[Castells et al., 2011] Castells, P., Vargas, S., and Wang, J. (2011). Novelty and diversity metrics for recommender systems: Choice, discovery and relevance. In *Proceedings of International Workshop on Diversity in Document Retrieval*, pages 23–37, Chicago, Illinois, USA.

[Chakraborty et al., 2019] Chakraborty, A., Ghosh, S., Ganguly, N., and P. Gummadi, K. (2019). Optimizing the recency-relevance-diversity trade-offs in non-personalized news recommendations. *Information Retrieval Journal*, 2019(2).

[Chen et al., 2016] Chen, L., wu, W., and He, L. (2016). Personality and recommendation diversity. In *Emotions and Personality in Personalized Services: Models, Evaluation and Applications*, pages 201–225. Springer International Publishing AG, Cham.

[Colleoni et al., 2014] Colleoni, E., Rozza, A., and A., A. (2014). Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 54(2):317–332.

[Epure et al., 2017] Epure, E., Kille, B., Ingvaldsen, J., Deneckere, R., Salinesi, C., and Albayrak, S. (2017). Recommending personalized news in short user sessions. In *RecSys 17 - Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 121–129, Como, Italy.

[Flaxman et al., 2016] Flaxman, S., Goel, S., and M. Rao, J. (2016). Filter bubbles, echo chambers and online news consumption. *Public Opinion Quarterly*, 80(1):298–310.

[Garcin et al., 2013] Garcin, F., Dimitrakakis, C., and Faltings, B. (2013). Personalized news recommendation with context trees. In *RecSys 2013 - Proceedings of the 7th ACM conference on Recommender Systems*, pages 105–112, Hong Kong.

[Garcin et al., 2012] Garcin, F., Zhou, K., and Faltings, B. (2012). Personalized news recommendation based on collaborative filtering. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 437–441, Macau.

[Habermas, 1962] Habermas, J. (1962). *Strukturwandel der Öffentlichkeit*. Suhrkamp Taschenbuch Verlag, Frankfurt am Main.

[Han and Yamana, 2017] Han, J. and Yamana, H. (2017). A survey on recommendation methods beyond accuracy. *IEICE Transactions on Information and Systems*, 100(12):2931–2944.

[Hijikata, 2014] Hijikata, Y. (2014). Offline evaluation for recommender systems. Presentation of the Osaka University Graduate School of Engineering Science. GroupLens Research at the University of Minnesota.

[Irsan and Khodra, 2019] Irsan, I. C. and Khodra, M. L. (2019). Hierarchical multi-label news article classification with distributed semantic model based features. *International Journal of Advances in Intelligent Informatics*, 5(1):40–47.

[Isinkaye et al., 2015] Isinkaye, F. O., Folajimi, Y. O., and Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation evaluation. *Egyptian Informatics Journal*, 2015(16):261–273.

[Javari and Jalili, 2014] Javari, A. and Jalili, M. (2014). A probabilistic model to resolve diversity-accuracy challenge of recommendation systems. *Knowledge and Information Systems*, 44(3):609–627.

[Karimi, 2018] Karimi, M. (2018). News recommender systems - survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227.

[Karimova et al., 2016] Karimova, F., Jannach, D., and Jugovac, M. (2016). A survey of e-commerce recommender systems. *European Scientific Journal*, 12(34):75–89.

[Kunaver and Pozrl, 2017] Kunaver, M. and Pozrl, T. (2017). Diversity in recommender systems - a survey. *Knowledge-Based Systems*, 2017(123):154–162.

[Lathia et al., 2013] Lathia, N., Hailes, S., Capra, L., and Amatriain, X. (2013). Temporal diversity in recommender systems. In *33rd Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, pages 210–217, Geneva, Switzerland.

[Li et al., 2011a] Li, L., Wang, D., Li, T., Knox, D., and Padmanabhan, B. (2011a). Scene: A scalable two-stage personalized news recommendation system. In *34th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, pages 210–217, Beijing, China.

[Li et al., 2011b] Li, L., Wang, D., Zhu, S.-Z., and Li, T. (2011b). Personalized news recommendation: A review and an experimental investigation. *Journal of Computer Science and Technology*, 26(5):754–766.

[Liu and Zhou, 2012] Liu, C. and Zhou, W.-X. (2012). An improved heats+probs hybrid recommendation algorithm based on heterogeneous initial resource configurations. *Physica A Statistical Mechanics and its Applications*, 391(22):5704–5711.

[Liu et al., 2010] Liu, J., Dolan, P., and Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In *Proceedings of the 2010 International Conference on Intelligent User Interfaces*, pages 31–40, Hong Kong.

[Metla et al., 2014] Metla, S. J., Zuva, T., and M., N. S. (2014). Aggregate diversity techniques in recommender systems. *Lecture Notes on Information Theory - LNIT*, 2(3):238–242.

[Mittelstadt, 2016] Mittelstadt, B. (2016). Auditing for transparency in content personalization systems. *International Journal of Communication*, 2016(10):4991–5002.

[Möllera et al., 2018] Möllera, J., Trilling, D., Helberger, N., and van Es, B. (2018). Do not blame it on the algorithm: An empirical assessment of multiple recommender systems and their impact on content diversity. *Information, Communication & Society*, 21(18):959–977.

[Mutz, 2002] Mutz, D. C. (2002). Cross-cutting social networks: Testing democratic theory in practice. *American Political Science Review*, 96(1):111–126.

[Nikolakopoulos and Karypis, 2019] Nikolakopoulos, A. N. and Karypis, G. (2019). Recwalk: Nearly uncoupled randomwalks for top-n recommendation. In *12th ACM Insternational Conference on Web Search and Data Mining*, pages 150–158, Melbourne, Victoria, Australia.

[Peska, 2016] Peska, L. (2016). Using the context of user feedback in recommender systems. In *11th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Telc, Czech Republic.

[Rodriguez et al., 2012] Rodriguez, M., Posse, C., and Zhang, E. (2012). Multiple objective optimization in recommender systems. In *RecSys 12 - Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 11–18, Dublin, Ireland.

[Saranya and Sadhasivam, 2012] Saranya, K. G. and Sadhasivam, G. S. (2012). A personalized online news recommendation system. *International Journal of Computer Applications*, 57(18):6–14.

[Szlavik et al., 2011] Szlavik, Z., W., K., and C., S. M. (2011). Diversity measurement of recommender systems under different user choice models. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 269–376, Barcelona, Spain.

[Tintarev, 2017] Tintarev, N. (2017). Presenting diversity aware recommendations: Making challenging news acceptable. In *The FATREC Workshop on Responsible Recommendation*, pages 9–12, Como, Italy.

[Vargas and Castells, 2011] Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys 11 - Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 109–116, Chicago, Illinois, USA.

[Wang et al., 2018] Wang, H., Zhang, F., Xie, X., and Guo, M. (2018). Deep knowledge-aware network for news recommendation. In *WWW 2018 - Proceedings of the 2018 World Wide Web Conference*, Lyon, France.

[Zhang and Hurley, 2008] Zhang, M. and Hurley, N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *RecSys 08 - Proceedings of 2008 ACM Conference on Recommender Systems*, pages 123–130, Lausanne, Switzerland.

[Zhao et al., 2018] Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., and Yin, D. (2018). Recommendations with negative feedback via pairwise deep reinforcement learning. In *24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, London, United Kingdom.

[Ziegler et al., 2005] Ziegler, C.-N., McNee, S. M., Konstant, J. A., and G., L. (2005). Improving recommendations lists through topic diversification. In *WWW 2005 - Proceedings of the 2005 World Wide Web Conference*, pages 22–32, Chiba, Japan.

# A

# Appendix

## A.1 Intake Survey

Table A.1 lists the complete set of questions used in the intake-survey for determining the political score for each user. The values next to the question define the weights in the left/right as well as liberal/conservative dimension. The survey was created by the organization Politools (https://politools.net/) and is used online on their Parteienkompass (https://parteienkompass.ch/). Please note that this survey was designed specifically with Switzerland in mind. A one-to-one adoption of this survey to cover the political landscape of a different country is not possible and would require additional changes.

Table A.1: Questions used in the intake-survey to determine political orientation.

| Survey Questsion | Left / Right | Lib. / Cons. |
|---|---|---|
| Menschen in der Schweiz werden immer älter. Soll das Rentenalter darum weiter erhöht werden? | 1 | 0 |
| Soll der Staat Menschen in Armut stärker unterstützen (Ausbau der Sozialhilfe)? | -1 | 0 |
| Sollen die Kosten für die Krankenversicherung an das Einkommen angepasst werden (Gutverdienende müssten mehr zahlen)? | -1 | 0 |
| Sollen junge Arbeitslose durch den Staat stärker unterstützt werden? | -1 | 0 |
| Soll ein Mindestlohn von 4'000 Franken für alle Arbeitnehmer/-innen eingeführt werden? | -1 | 0 |
| Soll der Staat dafür sorgen, dass an abgelegenen Orten vergleichbare Leistungen wie in den Städten angeboten werden (z.B. Poststellen, Mobile-Empfang, Verkehrsverbindungen, medizinische Versorgung)? | 0 | -1 |

Table A.1 continues on the next side.

Continuation of Table A.1

| Survey Questsion | Left / Right | Lib. / Cons. |
|---|---|---|
| Sollen Bäuerinnen und Bauern vom Staat mehr Geld erhalten (höhere Subvention für die Landwirtschaft)? | 0 | -1 |
| Sollen Geschäfte ihre Öffnungszeiten selber festlegen dürfen (Liberalisierung der Ladenöffnungszeiten)? | 1 | 1 |
| Sollen die Steuern für wohlhabende Personen erhöht werden? | -1 | 0 |
| Sollen Firmen weniger Steuern bezahlen? | 1 | 0 |
| Sollen in der Schweiz neue Atomkraftwerke gebaut werden dürfen? | 1 | 0 |
| Soll der Staat mehr Geld für den öffentlichen Verkehrs (Bahn, Bus, Tram) und weniger Geld für den Privatverkehr (Strassenbau) aufwenden? | -1 | 0 |
| Sollen homosexuelle Paare (Schwule und Lesben) heiraten dürfen (vollständige Gleichstellung mit der Ehe zwischen Mann und Frau)? | 0 | 1 |
| Soll es weiterhin erlaubt sein, eine Schwangerschaft in den ersten zwölf Wochen abzubrechen? | 0 | 1 |
| Soll Cannabis legalisiert werden? | 0 | 1 |
| Ist die Schweiz gegenüber Asylbewerbern/-innen zu grosszügig? | 1 | 0 |
| Sollen in der Schweiz geborene Ausländer/-innen automatisch den Schweizer Pass erhalten? | 0 | 1 |
| Sollen Ausländer/-innen in der Schweiz wählen und abstimmen dürfen? | 0 | 1 |
| Sollen öffentliche Orte vermehrt mit Video überwacht werden? | 1 | 0 |
| Soll die Polizei Sprayer/-innen und Randalierer/-innen strikter verfolgen und härter bestrafen? | 1 | 0 |
| Soll die Schweizer Armee abgeschafft werden? | -1 | 0 |
| Sollen vermehrt Personenkontrollen an der Schweizer Grenze durchgeführt werden? | 1 | -1 |
| Sollen Schweizer/-innen in der EU und EU-Bürger/-innen in der Schweiz frei arbeiten und wohnen dürfen (freier Personenverkehr zwischen der EU und der Schweiz)? | 0 | 1 |

End of Table A.1

## A.2 Recruitment Survey

| Rekrutierungsfrage |
|---|
| Frage 1: Ich bin... (1) weiblich (2) männlich |
| Frage 2: Wie alt sind Sie? |
| Frage 3: Bitte geben Sie die Postleitzahl Ihres Wohnortes an. |
| Frage 4: Besitzen Sie ein Smartphone? |
| Frage 5: Was für ein Smartphone besitzen Sie?<br>(1) Android 5+ (2) iOS 8+ (3) Windows Phone (4) anderes<br>(5) weiss nicht |
| Frage 6a: Wie hüfig nutzen Sie Ihr Smartphone um Nachrichten auf Onlineausgaben von Zeitungsn zu lesen?<br>(1) fast täglich (2) mehrmals pro Woche<br>(3) mehrmals pro Monat (4) seltener (5) nie |
| Frage 6b: Wie hüfig nutzen Sie Ihr Smartphone für das Angebot von News Portalen?<br>(1) fast täglich (2) mehrmals pro Woche<br>(3) mehrmals pro Monat (4) seltener (5) nie |
| Frage 7: Welche der folgenden Parteien vertritt am ehesten Ihr Gedankengut?<br>(1) SP (2) GPS (3) BDP (4) CPV (5) EVP (6) GLP<br>(7) FDP (8) SVP (9) eine andere Partei (10) keine Angabe |
| Frage 8: Wie würden Sie sich auf einer politischen Skale von 1 = ganz links bis 7 = gant rechts einordnen?<br>1-2: ganz oder mehrheitlich links<br>3-5: mehrheitlich in der Mitte<br>6-7: ganz oder mehrheitlich rechts |

Table A.2: Questions used for sampling possible experiment participants.

## A.3 Exit-Survey

The complete exit-survey and form for debriefing users after the experiment is on the enclosed CD in the subfolder for Appendix 3. The survey was created by Dr. Juliane Lischka and Alena Birrer.

## A.4 Reading Metrics

The complete list of reading metrics recorded during the experiment and the detailed evaluation is on the enclosed CD in the subfolder for Appendix 4.

# List of Figures

3.1 Sample distributions of news article for the three user groups. . . . . . . . 20

4.1 Sample screen of the in-app survey tool. . . . . . . . . . . . . . . . . . . . . 26
4.2 Translation of a sample distribution into a recommendation list. . . . . . 32

5.1 The app's home screen (left), article view (center) and reading list (right). 38

6.1 Total number of read articles (y-axis) for each hour of the day (x-axis). . 42
6.2 Total number of read articles (y-axis) for each day of the week (x-axis). . 42
6.3 Total number of logins (y-axis) for each hour of the day (x-axis). . . . . . 44
6.4 Total number of logins (y-axis) for each day of the week (x-axis). . . . . . 44
6.5 Number of read articles (y-axis) over reading duration in seconds (x-axis). 45

# List of Tables

4.1 List of available reading metrics for each user-article pair. . . . . . . . . . 27

A.1 Questions used in the intake-survey to determine political orientation. . . 65
A.2 Questions used for sampling possible experiment participants. . . . . . . . 67