A System for Online News Recommendations in Real-Time with Apache Mahout

Paul Beck[1], Manuel Blaser[1], Adrian Michalke[1], and Andreas Lommatzsch[2]

[1] Technische Universität Berlin, Straße des 17. Juni 135, D-10623 Berlin, Germany
{p.beck,manuel.blaser,michalke}@campus.tu-berlin.de
[2] DAI-Labor, Technische Universität Berlin, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
andreas@dai-lab.de

Abstract. With the ubiquitous access to the internet, news portals have become heavily consumed online services. The huge amount of published news makes it difficult for users to find relevant articles. Recommender systems have been developed for supporting users in finding the most interesting items in vast collections of available items. In contrast to traditional recommender systems, news recommender systems must address additional challenges. These challenges include the continuous changes in the set of items and the highly contextually dependent relevance of items as well as tight time constraints for providing recommendations and scalability requirements.
In this work, we present our recommender system built based on APACHE MAHOUT tailored to the needs of news recommender systems. Two algorithms are combined to ensure highly precise recommendations and a high reliability. The system is evaluated in the CLEF NEWSREEL challenge. We discuss the performance of different tested algorithms and configurations. The evaluation shows that the developed system provides high quality results and fulfills the requirements of stream-based recommender scenarios.

Keywords: recommender system, news recommender, stream processing, algorithms, similarity metrics and scalability

1 Introduction

Recommender Systems are valuable tools to support users in finding interesting objects or information in huge collections of data. Very popular recommender systems have been developed for online shops (e.g. AMAZON) or entertainment services (such as NETFLIX or SPOTIFY [4]). These recommender systems analyze the user-item interaction and build models for suggesting items matching the user's individual preferences.

Due to the success in the domains of online shopping and entertainment, the application of recommender systems in other domains is an interesting option. With the ubiquitous internet availability, news and social media portals have entered the focus of interest. Due to the huge and still growing amount of news published every minute, recommender systems are a promising solution to help users finding the most relevant news articles.

In contrast to the items of traditional recommender systems, news items change frequently and the relevance of news strongly depends on the context. Thus, news recommender systems must be capable of handling news streams fast and compute the relevance of articles efficiently ensuring that recommended news are "new".

Additional challenges for news recommender systems are (i.) fuzziness in identifying users, (ii.) short life cycles of news articles, (iii.) context-dependent user preferences, and (iv.) the diversity of published news. Moreover, news recommenders must fulfill technical requirements such as scalability and the ability of handling concurrent requests ensuring tight response time constraints.

Traditional recommender systems are based on static datasets describing the interactions between users and items. The user preferences can be expressed either explicitly based on ratings or implicitly by showing interest in an item (e.g. by retrieving a detailed item description). Algorithms that have been successfully trained on these datasets are user-based and item-based Collaborative Filtering.

This paper goes beyond the traditional approach to offer a solution for the specific requirements of recommending news. The presented system is build based on the APACHE MAHOUT framework, which already has been successful in commercial use [1]. We discuss how to process the continuous stream of news articles and requests and how to ensure that the recommender model stays "fresh". The different recommendation algorithms are evaluated within the framework of the CLEF NEWSREEL challenge.

The remaining paper is structured as follows. Section 2 explains the analyzed setting and the specific requirements in detail. Related research is discussed in Section 3 and Section 4 presents the approach. The evaluation results are discussed in Section 5. Finally, a conclusion and an outlook on future work are given in Section 6.

2 Problem Description

In this work, we analyze the news recommendation task defined in the CLEF NEWSREEL challenge [9]. The NEWSREEL challenge gives researchers the unique chance to evaluate recommender algorithms both online and offline on real-world data. This section explains the news recommendation scenario and discusses the specific challenges.

Web news portals are popular sources for finding the most up-to-date information about interesting incidents ("news"). Typically, there is a continuous stream of freshly published high varying articles making it difficult for users to find interesting articles in the huge mass of news. News Recommender Systems address this issue by suggesting articles that are likely to match the user's preferences. Such recommendations are usually presented as an overlay or in a box at the bottom of the news portal.

In contrast to most traditional recommender systems, news recommendation systems provide recommendations for web portals that do not force the user to register explicitly. Thus, user tracking must be done based on the web session resulting in a fuzzy identification of the user. Moreover, user preferences with respect to news strongly depend on the context. This means that web session and context data are the basis for computing recommendations.

The NEWSREEL challenge provides an *online task* ("living lab") and an *offline task*. The Evaluation of the *online task* is based on live user interaction and feedback. If a user requests a page from a news portal (participating in the NEWSREEL challenge), all teams are required to provide recommendation lists. From these lists of recommendations one list is randomly chosen and displayed to the user.

For the *online task*, the performance is measured in terms of the Click Through Rate (CTR). The CTR is defined as the proportion of clicked recommendations in terms of the total number of recommendations presented to the user [10].

The *offline task* is based on a stream of user interactions recorded over 4 weeks. Since no live user feedback is available in the offline task, the recommender algorithms should predict which items a user will request within the next minutes (evaluated based on the recorded stream). The performance is measured by the Prediction Accuracy ("offline CTR").

The major challenges in the NEWSREEL scenario are:

- Fast changing sets of news items requiring frequent model updates
- Fuzzy identification of users due to the fact that users do not have to login or register
- Multi-dimensional benchmarking considering both recommendation precision and technical complexity.

Before we present our approach, we analyze related work in the next section.

3 Related Work

In recent years, the amount of data published in the web and the number of items available in entertainment services have grown rapidly. Recommender systems address this problem by analyzing huge data collections and extracting items potentially matching the user preferences. In contrast to classic Information Retrieval Systems, users do not need to provide explicit queries. Recommendations are computed based on implicit or explicit user profiles. This section reviews existing recommender frameworks and recommender systems with respect to the considered news recommendation scenario. Furthermore, recommender algorithms and earlier NEWSREEL-contributions are discussed.

3.1 Recommender Frameworks

With the growing importance of recommender systems, different recommender frameworks have been developed, such as LENSKIT, MYMEDIALITE, and MAHOUT.

LENSKIT [7] is an open source framework originally developed by the GROUPLENS research group at the University of Minnesota. This framework is tailored to the demands of the research community. It is focusing on modularity allowing researchers to replace the provided components. The framework is written in JAVA enabling the platform independent deployment.

MYMEDIALITE [8] is an open source project developed at the University of Hildesheim. It supports Collaborative Filtering in the scenario of positive rating prediction and item prediction from positive only feedback. Further recommender algorithms and evaluation approaches are also implemented.

The MAHOUT framework is part of the Apache Software Foundation and is available as open source project online. It started as a part of the Apache Lucene project and went on becoming a top-level project in 2010. The first goal was to implement all 10 algorithms of Andrew Ng's paper "Map-Reduce for Machine Learning on Multicore" [2]. By now many additional algorithms are provided. In the field of recommendation, the implemented algorithms focus on Collaborative Filtering. Algorithms for clustering and classification are implemented as well. The framework provides a set of batch-based algorithms. Several algorithms are implemented based on the map-reduce paradigm enabling the execution in distributed environments.

MAHOUT recommenders are in commercial use by several institutions. The fields of application range from online shopping to the recommendation of research articles [1]. Due to the commercial use it appears to be a promising option to use MAHOUT recommenders in the setting of the NEWSREEL challenge as this challenge provides the opportunity to work with real-world data (see Section 2).

3.2 Existing News Recommender Systems

In the internet economy, there are several examples of good news recommendation systems in action. We analyze GOOGLE NEWS and BUZZER, a recommender for RSS feeds, which can be used for TWITTER.

GOOGLE NEWS [11] is an online news platform and recommendation system which aggregates news from other platforms. Initially a Collaborative Filtering algorithm was used. The system has been improved by combining Collaborative Filtering and content-based filtering in a hybrid recommender system.

BUZZER [16] is a recommender system developed by the University College of Dublin. The project studied methods for recommending niche news stories, typically receiving only a small number of clicks. Different recommender algorithms have been tested, such as PUBLIC-RANK, FRIENDS-RANK and CONTENT-RANK. The evaluation showed that the highest Click-Through-Rate was reached by Friends-Rank. Recommendations provided by the Content-Rank algorithm were least liked by the users.

The evaluation results for BUZZER indicate that popularity based approaches outperform the content-based approach in the news domain. Users do not seem to be restricted to fixed news topics. News stories liked by the majority of users are typically a good recommendation. In general, algorithms based on collaborative knowledge are more successful than algorithms relying on content-based knowledge.

Both systems show that in the field of news recommendation the use of recommenders may be a successful endeavor. The good performance of Collaborative Filtering methods is a further encouragement to work with the recommenders implemented in MAHOUT.

3.3 Recommender Algorithms

APACHE MAHOUT contains different implementations of Collaborative Filtering recommenders. Collaborative Filtering algorithms may apply either a user-based or an item-based approach. Dependent on the selected approach, similarities between users or items need to be calculated. For this purpose adequate similarity metrics have to be selected (e.g. *Cosine similarity* or *Tanimoto coefficient* [18]).

User-based Collaborative Filtering In order to compute recommendations for a user u_0, the user-based recommender identifies the users most similar to user u_0. The similarity can be computed based on similar preferences and rating behavior. Having identified similar users, the recommender algorithm suggest the items that the similar users like most (excluding the items u_0 already knows).

User-based approaches work successfully, if the underlying data contain for every user a sufficient number of similar users. The algorithms tend to suggest popular items since those items are liked by most users.

With regard to news portals a disadvantage of user-based approaches is that these algorithms require user-profiles. Thus, these algorithms are less suitable for web portals that can be used anonymously, as it is usually the case for news portals.

Item-based Collaborative Filtering Item-based recommenders compute the similarity between items. The algorithms determine in a first step the items user u_0 likes and suggest the items most similar to these items. The similarity is computed based on collaborative data ("users who liked item i also liked item j").

Considering the domain of news recommendation, the advantage of the item-based approach is that no explicit user profiles are required. Recommendations can be computed based on anonymous session data.

In the setting of the NEWSREEL challenge, item-based Collaborative Filtering appears to be a promising approach. Since recommendations may be gained from anonymous session data this approach is appropriate for the present scenario.

3.4 Approaches evaluated in NEWSREEL

In recent years several recommender systems have been implemented in the NEWSREEL challenge. A focus has been put on distributed processing of the message stream.

Domann et al. [5] implemented a recommender system applying most popular algorithms based on APACHE SPARK. This system reached a very good response time and a high availability in the NEWSREEL challenge 2016. The CTR has been above the baseline.

A similar approach has been developed by Ciobanu and Lommatzsch [3] based on the APACHE FLINK framework. Compared with APACHE SPARK, FLINK provides extended functions for handling stream data, but showed to be less stable in the evaluation.

Verbitskiy et al. [17] developed a recommender system using different variants of most popular items algorithms based on the AKKA framework. Overall, the system reached a high CTR significantly above the baseline and showed to be highly scalable.

The considered approaches show the significance of stream processing in the implementation of recommender systems. In order to handle data streams in our recommender scenario, we adapted a batch-processing approach to be capable of computing recommendations based on streams.

3.5 Discussion

Due to its success in commercial use the APACHE MAHOUT recommender framework provides a promising starting point for developing a recommender system tailored to the specific requirements of the NEWSREEL scenario. A further argument for using MAHOUTs Collaborative Filtering recommenders is that Collaborative Filtering algorithms have already achieved good results within existing news recommendation systems.

For the setting of the NEWSREEL challenge batch-based stream processing and item-based Collaborative Filtering seem to be appropriate approaches. In contrast to user-based approaches, item-based Collaborative Filtering algorithms do not require explicit user-profiles.

4 Approach

The objective of this work is to examine the potentials of using the APACHE MAHOUT framework for computing news recommendations. For revealing MAHOUTs potentials, the settings of both NEWSREEL tasks are highly suitable as these tasks comprise the evaluation of recommender algorithms using real-world data (see Section 2). Therefore, MAHOUT recommenders have been implemented within an evaluation environment for the NEWSREEL challenge [12].

This section provides an overview of the system architecture and the evaluation method. It explains how the freshness of the recommender models is ensured and how the cold-start problem is addressed. Furthermore, the choice of MAHOUT's configuration options is discussed. The developed approaches are evaluated in the offline task with respect to Prediction Accuracy and technical aspects as well as in the online task with respect to the Click Through Rate.

4.1 System Architecture

The developed recommender system implements an optimized handling of the different message types. Data extracted from *Impressions* messages, *Item Updates* messages, and *Recommendation Requests* messages are used for building the recommender models. *Impressions* represent an interaction between a user and a news item. No response is expected for *impression* messages. *Item Updates* inform the system about new items or changes in news articles. If a *Recommendation Request* is received, the recommender is supposed to return a list of recommendations.

Internally, the system maintains two models. The MAHOUT-based recommender aggregates the messages in batches and rebuilds recommender models periodically. In addition to the MAHOUT-based recommender, a ring-buffer-based recommender is trained that relies mainly on the NEWSREEL baseline recommender. This ring-buffer-based model is updated continuously.

A second recommender is required as there are situations in that the MAHOUT-based recommender cannot provide sufficient recommendations. A typical issue for Collaborative Filtering algorithms is the cold-start-problem. If an item is new or rarely requested there is not enough information to compute recommendations. A similar problem may arise from the batch building. For the recommender, an item is not known if the batch on which the recommender is based does not contain any information on this item. Hence, it is not possible to determine recommendations for this item.

Despite being used in model building, *Recommendation Requests* are also forwarded to the MAHOUT recommender. If the MAHOUT recommender provides the requested number of results, these results are returned. In case that the MAHOUT recommender does not provide sufficient results, the Default Recommender provides a fallback solution. The Default Recommender is supposed to complete the system's response. In addition to the recommendations already provided by the MAHOUT recommender, the Default Recommender delivers the missing number of recommendations in order to fulfill the recommendation request. Fig. 1 visualizes the architecture of the developed system.

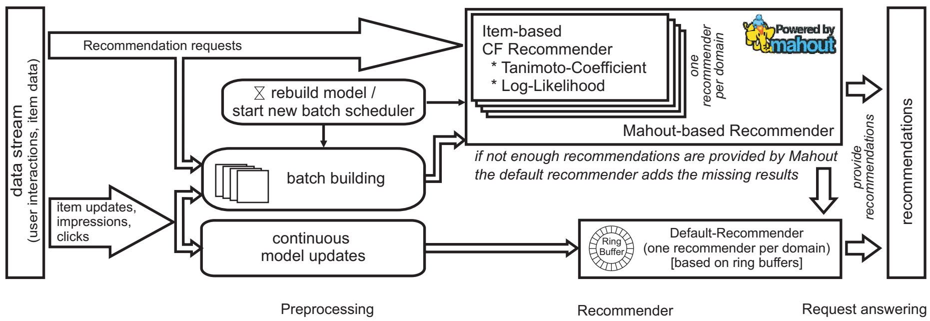

Fig. 1: Architecture of the recommender system. For providing recommendations two recommender models are available. Messages from the data stream are used for model building and for initializing the recommendation procedure. Source of MAHOUT-logo: [1]

Message Processing The recommender system implements the NEWSREEL recommender protocol. Messages are sent as HTTP requests. Received messages are processed using a JSON parser. The news portal ID ("domainID") and the message type are extracted in order to forward the message to the specific system component. Each message is processed in a separate thread enabling efficient concurrent message handling.

The MAHOUT data model The core component of our recommender system is the MAHOUT recommender. Since MAHOUT is not able to process data streams directly, a data model tailored to the setting of the NEWSREEL challenge has been developed.

A component has been implemented that builds batches based on the received stream data. We extract userID and itemID from received messages and store these data in a buffer. After having received n messages, a new model is built; this new model replaces the old one. Hence, each model is based on a non-overlapping batch of the message stream.

This approach has the advantage that the model building is done concurrently in the background without slowing down the answering of recommendation requests. The continuous re-building of the recommender model ensures the freshness of the model making sure that changes in the user behavior and changes in the item set are taken into account.

Configuration of MAHOUT The MAHOUT framework offers a variety of configuration options for Collaborative Filtering. Several recommender types and similarities are provided. The recommender type together with the similarity defines the method and internal implementation by which the similarity will be computed. In the following, the selection of MAHOUTs configuration options are explained.

The choice of configuration options is mainly determined by the lack of preference values in the stream data. Recommender types as well as similarities have to be appropriate for this situation. Two MAHOUT configurations have been selected for evaluation. Both configurations include a *GenericBooleanPrefItemBasedRecommender* and a *GenericItemSimilarity*. They differ with respect to the used similarity metrics. MAHOUT provides two applicable item similarity metrics for input data without preference values: *TanimotoCoefficientSimilarity* and *LogLikelihoodSimilarity*. The chosen configuration options are summarized in Table 1.

By using the *GenericItemSimilarity* all item similarity values are precomputed when the model is created. Hence, the effort for providing recommendations is reduced. Similarity values need to be compared but the values do not need to be computed when processing a recommendation request. The *GenericItemSimilarity* uses an item similarity metric which must be specified in the implementation.

The *TanimotoCoefficientSimilarity* is defined based on the number of users who share an item set. For two items, the Tanimoto Coefficient is given by the ratio of the number of shared users and the number of users who requested at least one of these items (see [18]).

According to a general description, a broader view is taken by the *LogLikelihood-Similarity*. The focus of this metric is described as the probability that objects are similar. In this context, it is concluded that high probability values indicate high similarities [15] (see [6] for calculation details).

Table 1: MAHOUT configurations for further examination. These configurations are considered to be suitable for the setting of the NEWSREEL challenge.

Configuration	
GenericBooleanPrefItemBasedRecommender with *GenericItemSimilarity* using	*LogLikelihoodSimilarity*
	TanimotoCoefficientSimilarity

Fallback Strategy If the MAHOUT-based recommender fails to provide the requested number of recommendations, a default ("fallback") recommender is used. This Default Recommender relies mainly on the NEWSREEL baseline recommender (see [12]). It is implemented based on a ring buffer containing the most recently requested news items.

The Default Recommender provides recommendations reliably and independently from the request properties since the Default Recommender is based on a most popular item approach. The combination of the MAHOUT recommender and the Default Recommender ensures a higher reliability; but a high fraction of requests completed by the Default Recommender may result in a reduced recommendation precision.

4.2 Evaluation Method

The evaluation is divided in an offline and online evaluation. In the online evaluation live user interactions have been collected by PLISTA. The official evaluation metric for this setting is the Click Through Rate. For the offline evaluation recorded user interactions are used.

The following criteria are considered in the offline evaluation: (i) Prediction Accuracy, (ii) Query Latency (iii) and the number of complete recommendations by the MAHOUT recommender. The Prediction Accuracy is the official evaluation metric for the offline evaluation, which can be thought of as “offline CTR”. Query Latency is given by the response time of the recommender to send a result. The number of complete recommendations by the MAHOUT recommender refers to the number of cases where the MAHOUT recommender alone can fulfill recommendation requests. This is relevant since a recommendation should consist of a requested number of items. A recommendation is correctly completed if the number of recommended items is equal to the number of requested items.

In both evaluation settings, we analyzed how the recommendation system performs based on the configurations, which have been assessed as being promising (see Table 1). The results are compared with the performance of a ring buffer-based model.

4.3 Discussion

In this section, we have shown how we implemented the recommendation system using MAHOUT recommenders. To ensure the freshness of MAHOUT recommenders, the data stream is processed block-wise and the recommenders are recalculated periodically on these blocks. For handling the cold-start-problem the Default Recommender is used as fallback solution.

From the variety of configuration options that the MAHOUT framework offers, those options have been chosen, that appear to be most promising for the setting of the NEWSREEL challenge. The performance that is achieved with these configurations is compared with respect to the results of the Default Recommender.

5 Evaluation

The developed recommender system has been evaluated within the settings of the offline and online task provided the NEWSREEL challenge [14]. MAHOUT-based recommenders have been configured to use a periodic interval of 50,000 messages for batch building and model rebuilding as this configuration has proven to be useful in a previous study [13].

5.1 Offline Task

The evaluation considers the complete dataset for the offline task and covers 4 weeks lasting from 31-Jan-2016, 22:00 to 28-Feb-2016, 22:00. Within this time 170,274,314 messages were recorded. In addition to Prediction Accuracy, which serves as the official evaluation metric for the offline task, evaluation metrics considering reliability and query latency are reported. The Default Recommender's results provide the baseline for the performance evaluation.

As the Default Recommender has been used in both MAHOUT-based models, the rate of complete responses to recommendation requests has been quite similar in all evaluation runs. There are 168,029,589 recommendation requests in the evaluation log file for the offline task. All considered models fulfilled more than 98.8% of recommendation requests.

Prediction Accuracy In each evaluation run of the offline evaluation the system requested 1,008,177,534 recommendations. A recommendation was considered to be correct if a user clicked on the recommended item within 6 minutes after the recommendation was made.

With regard to the whole evaluation period both MAHOUT-based models have achieved higher Prediction Accuracies than the baseline model. The best Prediction Accuracy has been measured for the model using the Tanimoto Coefficient Similarity. In comparison to the Baseline Model the fraction of correct recommendations is about 0.5 % higher. Figure 2 visualizes the Prediction Accuracies of the considered models.

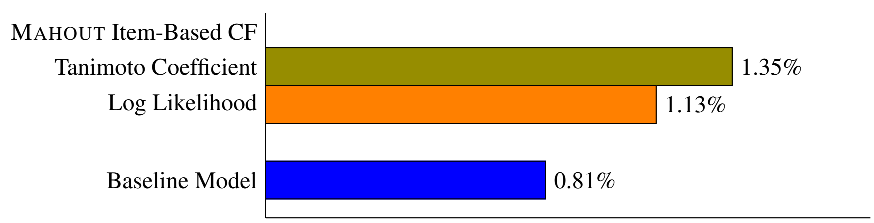

Fig. 2: Prediction Accuracy. Models using an item-based Collaborative Filtering recommender outperform the Baseline Model. The best results are achieved with the Tanimoto Coefficient Similarity.

The difference between the MAHOUT-based recommenders and the baseline model varies over the evaluation period. Most of the time the MAHOUT-based recommenders show a CTR above the baseline model. The model with Tanimoto Coefficient Similarity performs almost always better than the model based on the Log Likelihood Similarity. The changes of the Prediction Accuracy values (covering 6 hour intervals) for the evaluation period are visualized in Figure 3. Even though we did not change the recommender, the figure shows lower performance starting from calendar week 7. This is likely due to system modifications during the data collection.

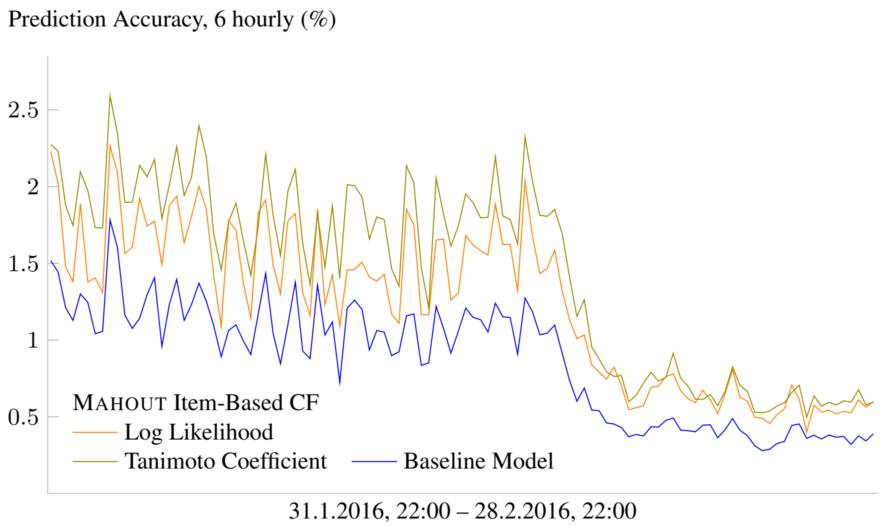

Fig. 3: Prediction Accuracy, six-hourly values for the whole evaluation period. Nearly throughout the entire evaluation period both MAHOUT-based recommenders perform better than the baseline recommender.

Query Latency The query latency of a recommender is measured by the time the recommender needed to respond to requests. The cumulative frequencies of response times are visualized in Figure 4.

MAHOUT-based recommenders responded to more than 50 % of the requests within 100 ms. Despite the complexity of finding the most similar items, the difference between the Baseline Model and the MAHOUT-based recommenders is moderate.

Complete Recommendations by MAHOUT recommenders MAHOUT-based models contain two recommenders. Recommendation requests are forwarded at first to a MAHOUT recommender and in case of incomplete responses the Default Recommender is used (see Section 4). For the two MAHOUT-based models under consideration, the experiment examined how the respective MAHOUT recommender performed in terms of providing complete recommendations.

Both MAHOUT recommenders were able to completely fulfill recommendation requests in a clear majority of cases. For about 84% of recommendation requests the two MAHOUT recommenders provided complete recommendations. The remaining requests were forwarded to the Default Recommender. Due to missing items in the user-item-table, about 6% of recommendations were delivered empty by both MAHOUT recommenders. The cold-start-problem is one reason for the remaining 10% of incomplete responses.

5.2 Online Task

The online task includes a 14-day evaluation period lasting from 24-Apr-2017 to 07-May-2017. As evaluation metric the NEWSREEL challenge official standard provided

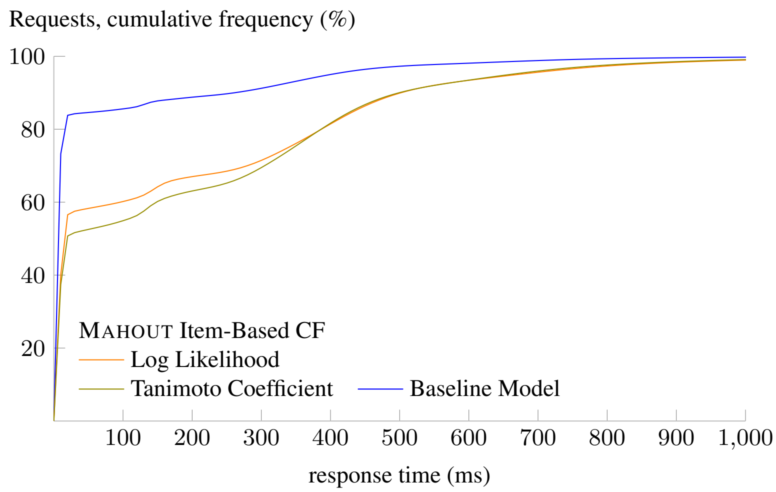

Fig. 4: Query Latency. MAHOUT-based recommenders have processed many recommendation requests within a brief period.

by the Click Through Rate has been used. The performance of MAHOUT-based recommenders is compared against the results for the baseline implementation of the challenge organizers.

Click Through Rate In the setting of the online task the recommendation lists of the analyzed models have been transformed to widgets. Throughout the evaluation period 68,582 widgets resulted from the model with Tanimoto Coefficient Similarity and 79,120 widgets from the model with Log Likelihood Similarity. Using the recommendations of the Baseline Model 62,052 widgets were produced.

Figure 5 visualizes the CTR's for the analyzed models. One MAHOUT-based recommender outperforms the Baseline Model. The difference between the model which uses the Tanimoto Coefficient and the Baseline Model is about 0.18 %.

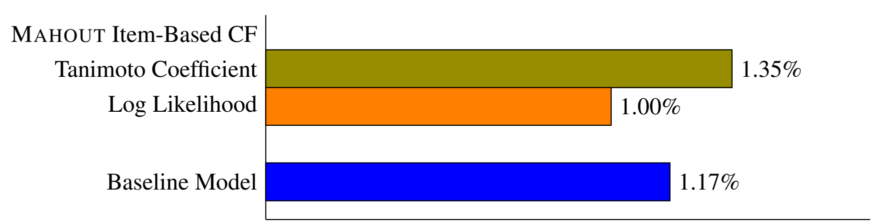

Fig. 5: Click Through Rate in the Online Evaluation. The best results are achieved by the item-based Collaborative Filtering recommender with Tanimoto Coefficient Similarity. The Log Likelihood model is not as good as the competitors.

The CTR's of all considered models vary over the evaluation period. Every model outperforms the others at some point of time. Figure 6 shows that the model using the Tanimoto coefficient performs well during the evaluation period. For most of the time, this model is above or not too far behind the baseline model. The model that uses the Log-Likelihood similarity is often behind the baseline model.

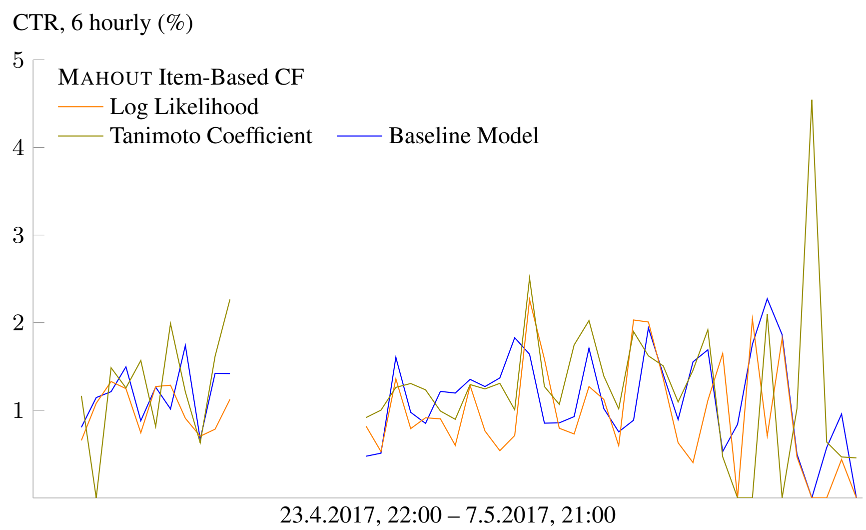

Fig. 6: Click Through Rate. Six-hourly values for all available measurements throughout the evaluation period. The lack of data points is caused by issues of the evaluation system.

5.3 Discussion

The evaluation has shown that MAHOUT-based recommenders perform well in terms of recommendation precision. In the offline and online task, the item-based Collaborative Filtering recommender using Tanimoto Coefficient Similarity delivered more correct recommendations than the Default Recommender. The limited performance of the recommender using the Log-Likelihood Similarity may be attributed to the stochastic component of this metric (see Section 4.1). Possibly, there are items with high Tanimoto Coefficient values but low probabilities of being similar. These items could make a difference in terms of recommendations.

The rather limited use of the Fallback Solution by the MAHOUT-based models is quite encouraging. It seems that the batch building every 50,000 messages leads to an acceptable number of missing items.

MAHOUT-based recommenders have limitations with regard to query latency. The number of requests for that MAHOUT-based recommenders respond quickly is substantially lower than for the Default Recommender. One approach for this problem consists of an optimized execution scheduling. The MAHOUT-based recommender might be restricted to a certain time-frame after which the Default Recommender takes over. This could prevent late system responses due to the runtime of the MAHOUT recommender.

6 Conclusion and Future Work

In this work, we presented our recommender system tailored to providing relevant news based on streamed data. An APACHE MAHOUT-based recommender has been combined with a most popular recommender (implemented based on a ring buffer). We have evaluated the developed recommender system in the framework of the CLEF NEWSREEL challenge. The results show that the implemented solution reliably provides precise recommendation results.

Results The use of MAHOUT leads to a higher CTR and Prediction Accuracy compared to using the default ring buffer-based recommenders only. The best results have been achieved by using the *GenericBooleanPrefItemBasedRecommender* with the *GenericItemSimilarity* wrapped around the *TanimotoCoefficientSimilarity*. Together with the ring buffer-based recommender a high system reliability is achieved. The batch building every 50,000 messages ensures the freshness of the recommender models and results in a reasonable data density while building the model.

Future Work Optimizing the batch building process considering the specific context is a promising approach for future work. Furthermore, sampling could be used to reduce the complexity of model building for very large data streams.

The developed solution combines two algorithms. In our analysis, we have studied several different recommender algorithms. A promising approach is to build an ensemble combining more than two algorithms. Based on our experiences, an ensemble may improve the recommendation precision; but the combination of algorithms also leads to a higher complexity and a longer response time for handling requests. The optimization of ensembles is a promising research direction.

References

1. Apache Mahout. Powered by mahout. web page. `http://mahout.apache.org/general/powered-by-mahout.html` (last accessed: 27/06/17).
2. C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 281–288, Cambridge, MA, USA, 2006. MIT Press.
3. A. Ciobanu and A. Lommatzsch. Development of a news recommender system based on apache flink. *CLEF2016 Working Notes*, 1609:606–617, 2016.
4. S. Dieleman. Recommending music on spotify with deep learning. web page, 08 2014. `http://benanne.github.io/2014/08/05/spotify-cnns.html`, (last accessed: 27/06/17).
5. J. Domann, J. Meiners, L. Helmers, and A. Lommatzsch. Real-time news recommendations using apache spark. *CLEF2016 Working Notes*, 1609:628–641, 2016.

6. T. Dunning. Accurate methods for the statistics of surprise and coincidence. *COMPUTATIONAL LINGUISTICS*, 19(1):61–74, 1993.
7. M. D. Ekstrand, M. Ludwig, J. Kolb, and J. T. Riedl. Lenskit: A modular recommender framework. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 349–350, New York, NY, USA, 2011. ACM.
8. Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 305–308, New York, NY, USA, 2011. ACM.
9. F. Hopfgartner, T. Brodt, J. Seiler, B. Kille, A. Lommatzsch, M. Larson, R. Turrin, and A. Serény. Benchmarking News Recommendations: The CLEF NewsREEL Use Case. *SIGIR Forum*, 49(2):129–136, Jan. 2016.
10. B. Kille, T. Brodt, T. Heintz, F. Hopfgartner, A. Lommatzsch, and J. Seiler. NEWSREEL 2014: Summary of the news recommendation evaluation lab. In *Working Notes for CLEF 2014 Conference*, pages 790–801, 2014. urn:nbn:de:0074-1180-0.
11. J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10, pages 31–40, New York, NY, USA, 2010. ACM.
12. A. Lommatzsch. Newsreel-template. github repository `https://github.com/andreas-dai/NewsREEL-Template` (last accessed: 25/05/17).
13. A. Lommatzsch. Real-time news recommendation using context-aware ensembles. In M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, editors, *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 51–62. Springer International Publishing, 2014.
14. A. Lommatzsch, B. Kille, F. Hopfgartner, M. Larson, T. Brodt, J. Seiler, and Ö. Özgöbek. CLEF 2017 NewsREEL Overview: A Stream-Based Recommender Task for Evaluation and Education. In *CLEF'17: Proceedings of the 8th International Conference of the CLEF Initiative*. Springer International Publishing, 2017.
15. S. Owen, R. Anil, T. Dunning, and E. Friedman. *Mahout in Action*. Manning Publications Co., Shelter Island, USA, 2012.
16. O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.
17. P. Probst and A. Lommatzsch. Optimizing a scalable news recommender system. *CLEF2016 Working Notes*, 1609:669–678, 2016.
18. D. J. Rogers and T. T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.