Optimizing a Scalable News Recommender System

Patrick Probst[1] and Andreas Lommatzsch[2]

[1] Technische Universität Berlin, Straße des 17. Juni 135, D-10623 Berlin, Germany
patrick.c.probst@campus.tu-berlin.de
[2] Agent Technologies in Business Applications and Telecommunication Group, AOT
Technische Universität Berlin, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
andreas.lommatzsch@dai-labor.de

Abstract. The huge amount of news articles published every hour makes it hard for users to find the relevant news matching the user's expectations. The main challenges when developing a recommender for the news domain are the continuous changes in the set of items, the context-dependent relevance of items, as well as the requirements with respect to scalability and response time. In this work, we present a scalable and distributable implementation of a real-time news recommender system based on the AKKA framework. Our approach focuses on optimizing the recommendation precision. It is able to adapt to the continuous changes of the set of relevant news articles as well as it considers the different user preferences dependent from the hour the day. Our implementation ensures that tight response time constraints are fulfilled and the system can be easily extended to streams of much larger volume. We implement three different recommendation algorithms namely, *Most Popular Items*, *Most Recent Items*, and *Most Recent Items of the Most Popular Categories.* A time-dependent delegation strategy is used for assigning requests to a recommender algorithm. We evaluate the developed recommender system in the CLEF-NEWSREEL challenge 2016. The evaluation shows that the recommender performs very successfully; the developed recommender has won the online evaluation in several timeframes.

Keywords: recommender system, scalability, Akka framework, most popular recommender, stream-based recommender

1 Introduction

The amount of available information in the World Wide Web is more and more increasing. This richness of information overwhelms users if not handled sophisticatedly. A filtered view on the huge amount of available content helps users to find interesting items. Recommender systems are developed to support users in filtering out the most relevant items matching the individual user's preferences. Recommender algorithms are used in many modern e-commerce applications. In this paper we focus on recommending news articles. With the spreading of handheld devices, such as smartphones and tablets, and the ubiquitous availability of internet connectivity, online news portal are becoming an important channel for real-time information. The main challenges for recommender systems in the news domain are the continuous changes in the set of potentially relevant items as well as the limited accuracy of user tracking (since users do not have to log in). User preferences often highly depend on the context and the specific domain. Furthermore, recommender systems in online settings must ensure tight response time constraints and be able to handle heavy load peaks [6].

The CLEF NewsREEL challenge [3] is a yearly competition giving researchers the opportunity to analyze and evaluate innovative news recommendation algorithms based on real-life data. We participate in the NewsREEL challenge in the second year. We further extended and optimized the system developed in 2015 [8]. The NewsREEL challenge consists of a Living lab task ("task 1") and an offline task ("Task 2") [5].

Task 1: The Living Lab Scenario In the Living Lab task participating teams must provide recommendations for different news portals in real time. The teams receive data describing freshly published articles as well as information about the interactions between users and items. Four types of messages are used for modeling the different types of information. The messages are transferred via HTTP connections and encoded in the JSON format. *Recommendation Requests* expect recommendable items as an answer and have to be replied within 100 ms. *Impressions* and *Item Updates* inform about the activity on the publisher's web sites. *Error Messages* indicate technical problems. Teams can register their algorithms and then compare the performance via the *Click-Through-Rate* (CTR) on a leader-board. The web portal allows participant registering new algorithms. In addition, the portal visualizes the evaluation results. The portal is called the Open Recommendation Platform (ORP) [1].

Task 2: The Simulated Stream Scenario For the offline evaluation scenario NewsREEL provides a large dataset consisting of a recorded data stream. The dataset contains all interaction data for two months [4]. In addition, a component for re-playing the dataset as a stream is provided. The dataset allows researchers to analyze the user behavior in detail. In addition, the offline evaluation ("Task 2") enables the reproducible evaluation of implemented algorithms with respect to scalability and throughput.

The remaining paper is structured as follows. In the next Section 2, we describe the scenario and the challenges in detail. Subsequently, we explain our approach and the details of the implemented algorithms in Section 3. The evaluation results are discussed in Section 4. Finally, Section 5 provides a conclusion and an outlook to future work.

2 Problem Description

Recommending news in real time is a challenging task due to the continuous changes in the item set, the fuzzy user identification, the context-depended user preferences as well as the requirements with respect to scalability and response time. In 2015 several different recommender algorithms have been tested in the NewsREEL challenge [6,7]. The algorithm we used in NewsREEL reached an average CTR, slightly above the baseline recommender. Based on the experiences we improve and optimize our algorithms, seeking to improve the CTR performance without sacrificing the scalability and the flexibility of our approach.

3 Approach

In order to consider context-dependent user preferences, we implement three different recommender algorithms. In addition, we learn a time-dependent delegation strategy selecting the most promising algorithm based on context parameters. In the next paragraphs we explain the different recommender algorithms in detail.

3.1 Most Popular Recommender

The *Most Popular Recommender* ranks news articles by the number of impressions. In order to take into account the continuous changes in the stream of items and user-item interactions, our algorithm uses a window-based approach. For each publisher, a sliding window is defined consisting of a fixed number of impressions. If a user clicks on an article, an impression is received and stored in the window. When the maximum number of impressions for the sliding window is reached, the oldest entry is deleted from the window. To provide recommendations a ranking of the news articles of the current window is computed. For each article, the number of impressions within the window is counted. The articles having received the largest number of impressions in the sliding window are recommended to users.

3.2 Most Recent Recommender

The *Most Recent Recommender* ranks news articles based on the most recent user-item interaction criterion. Similar to the Most Popular Recommender, a sliding window is used, separately optimized for each publisher. The implemented algorithm recommends the news items most recently added to the sliding window (that stores the news articles for the specific publisher).

3.3 Most Popular Category Recommender

In order to provide recommendation focused on specific user interests, we consider the categorization of the news items provided in the NewsREEL challenge. We refine the Most Popular Recommender by computing the popularity separately for each category and each publisher. Only articles from the most popular category are recommended. In contrast to the most popular recommender, this approach provides recommendations focused on the category for that the user already has showed an interest. The disadvantage is, that a smaller number of data is aggregated when computing the item ranking since only the data assigned for the requested category is considered. This might reduce the stability of the provided recommendations due to the smaller number of items considered for each category.

3.4 Delegation Strategy

We use a delegation strategy of assigning incoming requests to recommender algorithms. The delegation strategy is trained on the click events received in the most recent 15 minutes. The intersections of the clicked recommendations and the rankings from the three algorithms are compared. The largest intersection wins the competition and the winner algorithm is chosen to answer the next recommendation requests.

3.5 A Distributed Scalable Recommender System

Based on our positive experiences in NewsREEL 2015 we decided to implement the recommender algorithms in the Akka framework. The Akka framework provides a distributed real-time engine implementing the actor model [2]. It provides a flexible programming model and is designed for handling huge data streams efficiently. These features make the Akka framework a good choice for implementing a context-aware news recommender system [8]. Systems implemented based on Akka can be deployed on a cluster of computers being the basis for ensuring scalability. The nodes can either have the role of the master node or be one of n worker nodes (cf. Figure 1). Requests are distributed along the worker nodes using a load balancer.

4 Evaluation

We evaluated our developed recommender component in the CLEF NewsREEL challenge.

The ORP website (`http://orp.plista.com`) allows researchers to register implemented algorithms. In addition, the websites lists and visualizes several key figures describing the performance of the recommender algorithms. The portal does not only shows the CTR and the number of recommendation requests for our algorithms; it also lists the CTR of the other teams actively participating in the online evaluation. The teams are ranked based on the *Click Through Rate (CTR)* describing the proportion of clicks to the number of answered requests. Two baseline algorithms have been used. The first algorithm, named *Berlin*, uses a most popular strategy for recommending items. It considers the most recently requested 50 distinct items. The second baseline algorithm (maintained by the team named *baseline*) uses a most recent strategy implemented based on a ring buffer. We compare the performance of our recommender and the active recommender teams in the online challenge for two different timeframes.

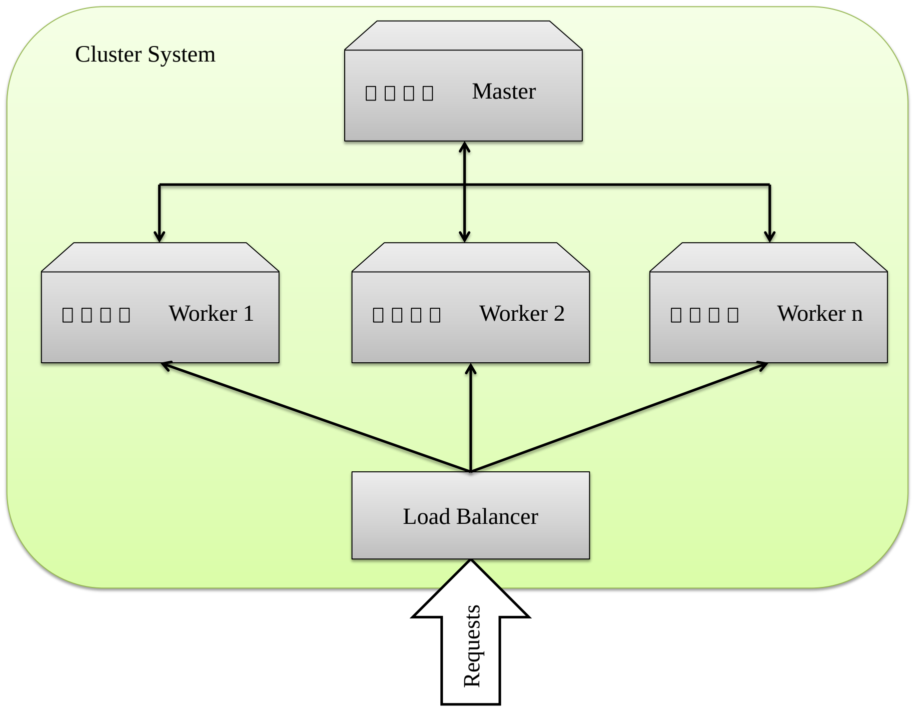

Fig. 1. Master-Worker-Model. This scheme visualizes a *Master-Worker Model*. The figure depicts a cluster consisting of one master node and n worker nodes. The workers are used for the communication with external components. The master node provides the commonly used resources. Master and worker nodes can either run on the same or on different machines. The number of workers can be chosen freely to increase the system throughput. A load balancer is used for assigning the requests optimally.

Analyzed Recommenders Algorithms Two recommender implementations have been analyzed. The first recommender, called XYZ uses the *Most Popular Items* algorithm only. The second recommender, XYZ-2.0 uses the delegation strategy. Recommendation requests are answered by the *Most Popular Items*, the *Most Recent User-item interactions*, or *Most Popular Category* algorithm.

Analyzed Timeframes In the following paragraphs we analyze two timeframes in detail. The first period includes 4 days in March (March 5th until March 8th). The second timeframe includes one week in April 2016 (April 10th until April 16th).

CTR analysis of the first timeframe

The CTR of the algorithm XYZ and the baseline recommenders are visualized in Table 1). The results show that our recommender outperforms the baseline recommenders. The average variance of the XYZ-recommender is slightly lower compared to the baseline recommenders.

Table 1. The table lists the evaluation results for the first timeframe (from 2016-03-05 until 2016-03-08). In each row, the highest CTR is colored green and the lowest CTR is highlighted red. The CTR is summarized for all instances of a recommender and for all publishers.

Date	XYZ		Baseline - DRB		Baseline - MP	
	CTR [%]	Var $[\cdot 10^{-4}]$	CTR [%]	Var $[\cdot 10^{-4}]$	CTR [%]	Var $[\cdot 10^{-4}]$
2016-03-05	0.609	0.00099	0.625	0.00197	0.566	0.00101
2016-03-06	0.891	0.00220	0.861	0.00374	0.903	0.00688
2016-03-07	0.943	0.00238	0.703	0.00124	0.869	0.00212
2016-03-08	1.306	0.00738	1.314	0.00875	1.129	0.00463
Average	0.937	0.00324	0.876	0.00393	0.867	0.00366

A-A Testing: Within the first time frame, the XYZ-recommender is registered four times in the ORP-interface. All instances map to the same recommender instance. Therefore, it is possible to compare the reached CTR. As the same instances are mapped, no differences in the CTR are expected. We are comparing the CTR for one publisher, namely *sport1.de*, who has the largest number of requests in this period (99.43 %) in Table 2. The variance is shown for the four days and is generally on a low level. Nevertheless, it varies between the days. On the first and last day it is notably higher compared to the other days.

Table 2. The table shows the result of the A-A testing analysis for the first timeframe. We list the measures CTR and the CTR variance for our four instances of the XYZ algorithm. Since the CTR is measured in %, we provide the variance measured in 10^{-4}.

Date	CTR of algorithm instances [%]				CTR variance
	XYZ-I	XYZ-II	XYZ-III	XYZ-IV	over the 4 instances
2016-03-05	0.397	0.648	0.579	0.825	$0.00031 \cdot 10^{-4}$
2016-03-06	0.773	1.008	0.982	0.863	$0.00012 \cdot 10^{-4}$
2016-03-07	0.949	0.912	1.019	0.979	$0.00002 \cdot 10^{-4}$
2016-03-08	1.508	1.410	1.587	1.875	$0.00037 \cdot 10^{-4}$

Second Timeframe: Week-based CTR-comparison.

We compare the CTR of participating teams for the second period (cf. Table 3 and Figure 2). Within this timeframe, our recommenders reach the top CTR. Recommender XYZ-2.0 reaches a slightly better CTR then recommender XYZ.

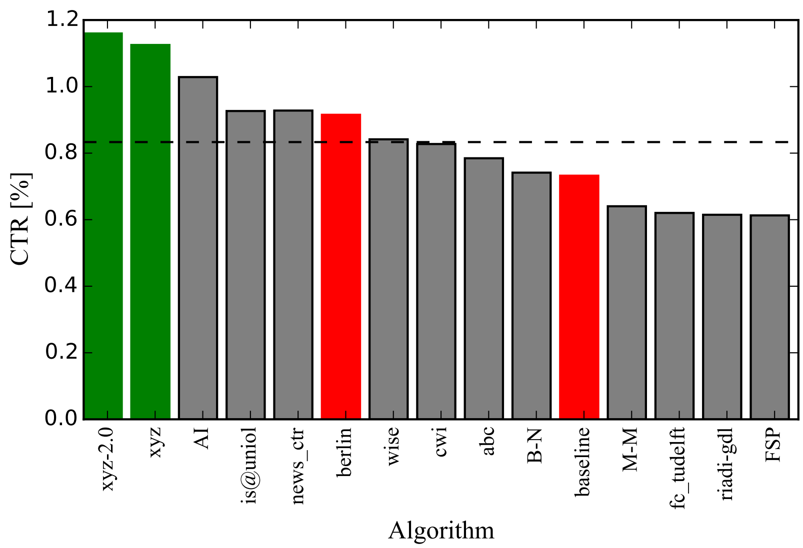

Fig. 2. The figure visualizes the CTR of the teams actively participating in the analyzed timeframe. Our recommender algorithms are highlighted in green. The baseline recommenders are colored red. The average CTR (= 0.83) is shown by a dotted line. The results show that our recommender algorithms outperform the baseline and reach the best CTR within the analyzed timeframe.

Delegation Strategy Evaluation We analyze the delegation strategy used by the XYZ-2.0 recommender. Table 4 shows the ratio of recommendations answered by the three implemented recommenders. We analyze at the same timeframe as depicted in Figure 3. The *Most Popular* recommender answered the majority of requests ($> 95\%$). Only a small percentage of requests has been delegated to the *Most Recent* and the *Most Popular Category* recommender ($< 5\%$).

5 Conclusion & Outlook

In this paper we presented our recommender components developed based on the AKKA framework. We evaluated two different versions of the recommender

Table 3. The table shows the evaluation results for the second analyzed timeframe computed on the dataset describing the timeframe from 2016-04-10 until 2016-04-16. The data is retrieved from the ORP-website. The total number of requests and clicks as well as the CTR are listed. Our recommenders are colored green. The baseline recommenders are highlighted in red. The Table is sorted by CTR.

Algorithm	Requests	Clicks	CTR [%]
xyz-2.0	117016	1357	1.160
xyz	72174	812	1.125
artificial intelligence (AI)	56973	586	1.029
news_ctr	25649	238	0.928
is@uniol	471092	4365	0.927
berlin	31686	290	0.915
wise	15092	127	0.842
cwi	90890	752	0.827
abc	42442	333	0.785
breaking-news (B-N)	10926	81	0.741
baseline	6146	45	0.732
moldawien-madness (M-M)	18741	120	0.640
fc_tudelft	37083	230	0.620
riadi-gdl	26516	163	0.615
flumingsparkteam (FSP)	85673	525	0.613

Table 4. The table lists the key figures for the implemented delegation strategy. The results show that most requests are delegated to the *Most Popular* recommender; only a small fraction of requests is handled by the *Most Recent* and the *Most Popular Category* recommender.

Date	Most Popular	Most Recent	Most Popular Category
2016-04-10	96.48%	3.50%	0.02%
2016-04-11	98.05%	1.81%	0.14%
2016-04-13	96.44%	3.50%	0.06%
2016-04-14	100.00%	0.00%	0.00%
2016-04-15	97.71%	1.02%	1.27%
2016-04-16	98.38%	0.92%	0.70%
∅	97.84%	1.79%	0.37%

approach in the online scenario. In the analyzed timeframes our recommender outperformed the competing teams. This shows that the recommender approach has a big potential. Unfortunately, we could not participate in the complete evaluation period; so the analyzed timeframes are relatively short. We will keep on participating in the online evaluation to verify the significance of the observed CTR performance.

Recommender System Performance In the first analyzed evaluation timeframe, our recommender XYZ outperformed the CTR of the baseline recommenders; but there were other teams in the challenge reaching a better CTR than our recommender XYZ.

In the second analyzed evaluation timeframe, we compared the performance of the algorithms XYZ and XYZ-2.0 with the other participating teams. Our algorithms reached the best CTR in this timeframe.

A-A Testing: For estimating the variance of the results, we performed an A-A testing. Our A-A tests showed only minor differences between the different instances of our algorithms. This indicates the variance of the CTR is low. Hence, there is a small random component in the data. This verifies the significance of the reached CTR in the analyzed timeframes.

The Delegation Strategy The algorithm XYZ-2.0 uses a delegation strategy to answer recommendation requests either by a *Most Popular*, a *Most Recent user-item Interaction* or a *Most Recent Category* recommender. Our evaluation shows that the majority of requests have been delegated to the most popular algorithm. This is an explanation why the CTR of the algorithms XYZ and XYZ-2.0 are very similar. However, the use of the delegation strategy leads to a CTR improvement.

Future Work In this paper we showed that a combination of different *Most Popular* algorithms reaches a high CTR. As future work, we plan to put a stronger focus on the delegation strategy and on optimizing the window size considered in the delegation component. In addition, we are working on considering additional aspects that can be used for measuring the popularity of news articles, such as number of clicks and total time spend on the news items.

Acknowledgments

The research leading to these results was performed in the CrowdRec project, which has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement No. 610594.

References

1. Torben Brodt and Frank Hopfgartner. Shedding light on a living lab: The clef newsreel open recommendation platform. In *IIiX'14: Proceedings of the Information Interaction in Context Conference*, pages 223–226. ACM, 08 2014.

2. Carl Hewitt, Peter Bishop, and Richard Steiger. A universal modular actor formalism for artificial intelligence. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI'73, pages 235–245, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.
3. Frank Hopfgartner, Torben Brodt, Jonas Seiler, Benjamin Kille, Andreas Lommatzsch, Martha Larson, Roberto Turrin, and András Serény. Benchmarking news recommendations: The clef newsreel use case. *SIGIR Forum*, 49(2):129–136, January 2016.
4. Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. The plista dataset. In *NRS'13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, ICPS, pages 14–22. ACM, 10 2013.
5. Benjamin Kille, Andreas Lommatzsch, Gebrekirstos Gebremeskel, Frank Hopfgartner, Martha Larson, Jonas Seiler, Davide Malagoli, Andras Sereny, Torben Brodt, and Arjen de Vries. Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-Time Stream- Recommendation Algorithms. In Norbert Fuhr, Paulo Quaresma, Birger Larsen, Teresa Goncalves, Krisztian Balog, Craig Macdonald, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction 7th Intl. Conf. of the CLEF Association, CLEF 2016, Evora, Portugal, September 5-8, 2016.*, LNCS 9822. Springer, 2016.
6. Andreas Lommatzsch and Sahin Albayrak. Real-time recommendations for user-item streams. In *Proc. of the 30th Symposium On Applied Computing, SAC 2015*, SAC '15, pages 1039–1046, New York, NY, USA, 2015. ACM.
7. Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*, chapter Introduction to Recommender Systems Handbook, pages 1–35. Springer US, Boston, MA, 2011.
8. Ilya Verbitskiy, Patrick Probst, and Andreas Lommatzsch. Development and evaluation of a highly scalable news recommender system. In *Working Notes of the 6th International Conference of the CLEF Initiative*. CEUR Workshop Proceedings, 2015. Vol-1391, urn:nbn:de:0074-1391-8.