Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-Time Stream-Recommendation Algorithms

– PREPRINT –

Benjamin Kille[1], Andreas Lommatzsch[1], Gebrekirstos G. Gebremeskel[7],
Frank Hopfgartner[6], Martha Larson[4,8], Jonas Seiler[5], Davide Malagoli[2],
András Serény[3], Torben Brodt[5], and Arjen P. de Vries[8]

[1] TU Berlin, Berlin, Germany
{benjamin.kille,andreas.lommatzsch}@dai-labor.de
[2] CWI, Amsterdam, The Netherlands
g.g.gebremeskel@cwi.nl
[3] University of Glasgow, Glasgow, UK
frank.hopfgartner@glasgow.ac.uk
[4] TU Delft, Delft, The Netherlands
m.a.larson@tudelft.nl
[5] Plista GmbH, Berlin, Germany
{torben.brodt,jonas.seiler}@plista.com
[6] ContentWise R&D — Moviri, Milan, Italy
davide.malagoli@moviri.com
[7] CWI, Amsterdam, The Netherlands
g.g.gebremeskel@cwi.nl
[8] Radboud University Nijmegen, The Netherlands
arjen@acm.org

Abstract. Successful news recommendation requires facing the challenges of dynamic item sets, contextual item relevance, and of fulfilling non-functional requirements, such as response time. The CLEF NewsREEL challenge is a campaign-style evaluation lab allowing participants to tackle news recommendation and to optimize and evaluate their recommender algorithms both online and offline. In this paper, we summarize the objectives and challenges of NewsREEL 2016. We cover two contrasting perspectives on the challenge: that of the operator (the business providing recommendations) and that of the challenge participant (the researchers developing recommender algorithms). In the intersection of these perspectives, new insights can be gained on how to effectively evaluate real-time stream recommendation algorithms.

Keywords: recommender systems · news · multi-dimensional evaluation · living lab · stream-based recommender

1 Introduction

Comparing the performance of algorithms requires evaluation under controlled conditions. Conventionally, in the recommender system research community, controlled conditions are created by adopting a static data set, and a single evaluation metric. In this paper, we discuss how evaluation of real-time stream recommendation algorithms presents challenges that cannot be so easily controlled for. Our topic is the News Recommendation Evaluation Lab (NewsREEL) [12] at CLEF 2016. NewsREEL makes it possible for participants to test news recommendation algorithms online. We focus here on two particular issues that online recommenders face: data variation and non-functional requirements. Our novel focus is a contrast between two perspectives in the online challenge: the perspective of *recommender system operators*, who wish to make a pragmatic choice of the best recommender algorithm for their purposes and the perspective of the *participants* of the challenge, researchers who are trying to understand the extent to which their experiments represent controlled conditions. First, we present the two issues in more depth. The data variation in the ecosystem of a real-time stream-recommendation algorithm is extreme, bringing to mind the adage "the only thing that stays the same is change". User interaction patterns with news items may shift radically, during a high-profile event, or unexpected breaking news. Interaction patterns may differ depending on region, device, or news source. New items are generated constantly, and the shelf life of old items expires. Different user subpopulations interact with content in different ways. Evaluating real-life recommender systems is challenging, since it is no longer possible to carefully control conditions in the face of such variation. Real-life recommender systems must be responsive to these variations, and, at the same time, must also fulfill non-functional requirements. Users request information continuously in stream of interactions. Huge numbers of simultaneously interacting users create peak loads. Recommender systems must remain available, and provide sub-second responses. Both recommender system operators and challenge participants agree that A/B testing is the approach to take in order to assess algorithms for stream recommendation. A/B testing splits users into disjoint groups each of which interacts with a specific system. A decision can then be made on which system is better. Operators and challenge participants contrast in their perspectives on how the comparison is made. We cover the position of each briefly in turn. The goal of the operator is to run a successful service and/or business. The operator is interested in making a practical choice between algorithms. As differences emerge between systems running online, the operator disables inferior systems. The algorithm that survives this "survival of the fittest" process suits the operators' needs. However, the particularities of the performance of the recommender algorithms during the test window are tied to the specific time of the window and the specific user assignments. Repeating the evaluation is infeasible. Businesses deploy sophisticated system architectures which enable them to cope with the requirements of scale and response time. The value of an algorithm is related to its ability to perform within a certain architecture. The goal of the challenge participant is to test algorithms in a real-world system, as well as to understand the differences between algorithms in detail. A participant in CLEF NewsREEL (Task 1) must deploy a recommendation engine that serves different publishers of online news in real-time. Participants are interested in repeatable experiments. In past years, we have noted that participation in NewsREEL requires the investment of a great deal of engineering effort on the part of participants. This year, we go beyond that observation to look at the contrast between the operators' and the participants' point of view. We hope that explicitly examining the differences will lead us to deeper insight on how they can productively complement each other. The operator/participant perspective contrast makes NewsREEL arguably more difficult and less straightforward than other recommender system benchmarks. Researchers who are accustomed to working with static data sets face a steep learning curve when it comes to stream-recommendation. Anyone who starts with the assumption that NewsREEL is just another Netflix-type competition will soon be frustrated. Offline evaluation procedures abstract from functional restrictions. Researchers who are used to offline evaluation tend not to consider such requirements. These skills are not taught in conventional machine learning or data science courses. Further, within NewsREEL, the 'view' of the participant on the data is limited because the associations between items and interactions is not explicit, but rather established via temporal proximity. For this reason, researchers might find that the depth to which they can analyze their results is more limited than they would otherwise expect. Such limitations arise because online systems exist to serve users, and their function as a living lab to evaluate algorithms, although important, remains secondary. The contrast, however, gives rise to a number of advantages. We believe that the interplay between functional and non-functional aspects is not taught in conventional courses, since it is simply very hard to teach without concrete example systems. NewsREEL allows researchers to experience in real-life what it means to have a highly promising algorithm which turns out to struggle when faced with real-world variation in data patterns and volume flow. Further, the contrast inspires us to dig more deeply into what can be done in order to add a certain amount of control to real-time recommender system evaluation. Specifically, NewsREEL releases a dataset (Task 2) that allows researchers to replay a certain period of the recommender system. The remainder of the paper discusses the objectives and challenges of NewsREEL 2016, and presents the contrasting perspectives of operator and participant in more depth. Section 2 sheds light on existing efforts to benchmark recommender systems. Section 3 introduces both tasks defined in the scope of NewsREEL. Section 4 elaborates on benchmarking tools used in NewsREEL. We introduce ORP (Task 1) and Idomaar (Task 2) supporting evaluation. Section 6 presents preliminary findings. Finally, Section 7 summarizes objectives of NewsREEL and outlines steps to further enhance benchmarking of news recommender systems.

2 Related Work

Evaluating information access systems challenges academia and industry alike, but conventionally they take different approaches. Academic researchers tend to focus on data-driven evaluation. Industry favors exploring algorithms in form of A/B tests. This section provides an overview of related work on these two approaches.

2.1 Benchmarking in Static Environments

Recommender systems carry out evaluation on standard test collections, similar to those performed in the field of information retrieval. A test collection usually consists of time-aligned ratings on items provided by a larger number of users, and of user attributes. The most popular test collection consists of movie ratings [11]. In order to benchmark recommendation performance, the dataset is usually split based on the time when a rating was provided, resulting in a training and a test dataset. The recommendation task is then to predict the rating that a user provided for an item in the test set. Over the years, various benchmarking campaigns have been organized to promote recommender systems evaluation, e.g., as part of scientific conferences ([2, 21, 19]) or as Kaggle[9] competitions (e.g., [18]). Apart from providing static datasets and organizing challenges to benchmark recommendation algorithms using these datasets, the research community has been very active in developing software and open source toolkits for the evaluation of static datasets. For example, Ekstrand et al. [7] introduce the `LensKit`[10] framework that contains several recommendation algorithms and benchmarking parameters. Similar frameworks have been been developed by Gantner et al. [8] and Said and Bellogín [20]. Such frameworks approach recommender systems evaluation from a static point of view, i.e., given a static dataset, the recommendation task is to predict users' ratings. Although this approach has some merits, it fails to address dynamic aspects that might influence recommendation tasks. Little work has focused on the relation between findings in static environments and online performances. Maksai et al. [17] evaluate how accuracy, diversity, coverage, and serendipity measured offline transfer to online settings. Their results indicate that offline accuracy does not suffice to predict users reactions. An overview of limitations of offline evaluation is provided in the next section.

2.2 Benchmarking in Dynamic Environments

In recent years, an increase has been observed in research efforts focusing on the evaluation of recommender system performance outside of the standard evaluation setting outlined above. For example, Chen et al. [4] performed experiments on recommending microblog posts. Similar work is presented by Diaz-Avilez et al. [6]. Chen et al. [5] studied various algorithms for real-time bidding of online ads. Garcin et al. [9] and Lommatzsch [16] focus on news recommendation. These approaches have in common that they are all evaluated in a live scenario, i.e., recommender algorithms have been benchmarked by performing A/B testing. A/B testing addresses various limitations that arise when using static datasets. In particular, research on static databases does not take external factors into account that might influence users' rating behavior. In the context of news, such external factors could be emerging trends and news stories. In the same context, the freshness of items (i.e., news articles) plays an important role that needs to be considered. At the same time, computational complexity is out of focus in most academic research scenarios. Quick computation is of uttermost importance for commercial recommender systems. Differing from search results provided by an information retrieval system, recommendations are provided proactively without any explicit request. Another challenge is the large number of requests and updates that online systems have to deal with. Offline evaluation using a static dataset conducts an exact comparison between different algorithms and participating teams. However, offline

[9] `http://www.kaggle.com`

[10] `http://lenskit.org/`

evaluation requires assumptions, such as that past rating or consumption behavior is able to reflect future preferences. The benchmarking community is just starting to make progress in overcoming these limitations. Notable efforts from the Information Retrieval community include the CLEF Living Labs task [1], which uses real-world queries and user clicks for evaluation. Also, the TREC Live Question Answering task[11] involves online evaluation, and requires participants to focus on both response time and answer quality. .

3 Problem Description

Publishers let users access news stories on digital news portals. The number of articles available can overwhelm users inducing an information overload problem. To address this problem, publishers deploy recommender systems suggesting interesting articles to their users. CLEF NewsREEL evaluates such systems on the basis of how well users respond to the suggestions provided. NewsREEL divides into two tasks. Task 1 interfaces with an operating news recommender system making it possible to conduct A/B testing. For a detailed description of the evaluation scenario, we refer to [13]. Task 2 uses a dataset [14] to compare recommendation algorithms. For a detailed overview of this task, we refer to [15]. Both settings are subject to a variety of challenges. First, we cannot reliably track users over longer periods of time. Publishers use session cookies to recognize visitors. Those entail multiple issues. Users may share devices creating ambiguous profiles. Users may use multiple devices spreading their activity across multiple identifiers. Finally, users may prohibit cookies. Consequently, systems only receive limited knowledge about their users. Second, we deal with fluctuating collections of items. New stories emerge every day. Simultaneously, older stories become less interesting to the public.

3.1 Task 1: Benchmark News Recommendations in a Living Lab

Task 1 has participants access an operating recommender system — the Open Recommendation Platform (ORP) [3]. Publishers run webportals offering news articles. As users visit these portals, they trigger recommendation requests. ORP receives these requests and distributes them randomly across recommendation engines deployed by participants. Subsequently, the chosen recommendation engine returns a ranked list of news articles which ORP forwards to the publisher. The length of the list depends on the publishers' user interface. ORP keeps track of how recipients respond to recommendations embedded in the publishers' website. Users signal interest when they click on recommendations. Missing clicks represent a somewhat unclear form of feedback. We cannot determine whether the lack of a click means that the user was not interested in the recommendation, or simply did not notice it. An underlying assumption is that disparities between groups of users will even out as participants serve a sufficiently large number of requests. In other words, the chance that an individual participant has a noticeable disadvantage becomes small as the number of requests gets larger. We determine the best contribution in terms of *click-through-rate* (CTR). The CTR represents the proportion of

[11] `https://sites.google.com/site/trecliveqa2015/`

suggestions which recipients click. Later we will see that a key question is at which rate the differences between two streams of recommendation requests even out.

3.2 Task 2: Benchmark News Recommendations in a Simulated Environment

In addition to the online task evaluated based on live feedback, NewsREEL also offers Task 2, which involves offline evaluation based on a large dataset. The dataset has been created by recording the messages in the online evaluation over two months. The dataset consists of $\approx$ 100 million messages (Table 1). Each message contains a timestamp allowing the simulation of the online stream by replaying the dataset in the original order. A detailed description of the nature of the dataset is provided in [14].

Table 1. The key figures of the offline dataset provided for Task 2

item create/update	**user-item**	**interactions**	**sum of messages**
July 2014	618 487	53 323 934	53 942 421
August 2014	354 699	48 126 400	48 481 099
sum of messages	973 186	101 450 334	102 423 520

The offline task focuses on reproducible evaluation of recommender algorithms. Simultaneously, the goal is to stay as close to the online system as possible. The participants should show that their recommender algorithms achieve a high CTR in different contexts (compared to the baseline recommender). In addition, the participants should show that the recommender scales well with the number of messages per minute. Since the offline tasks enables the simulation of different load levels, participants can show how new algorithms handle load peaks and how much time is required for processing the requests (expected response time distribution). NewsREEL Task 2 enables the reproducible evaluation of recommender algorithms. The realistic simulation of the NewsREEL message streams enables the detailed debugging as well as the simulation of different load levels. Since the evaluation is offline, teams can abstract away from network problems and optimize the algorithms on a well-defined dataset. Problems can be debugged and the performance of algorithms can be analyzed with respect to different metrics.

3.3 Summary

In this section, we have presented the two tasks that NewsREEL offers to participants. We have introduced ORP, which lets participants connect to a stream of recommendation requests issued by actual users. We have detailed the dataset released by NewsREEL to allow participants to evaluate recommendation algorithms offline and optimize their algorithms prior to deploy them online. It provides more than 100 million interactions, representing a comprehensive data set. Participants can implement collaborative filtering as well as content-based recommenders as the data set contains both interaction logs and item descriptions.

4 Multi-dimensional Evaluation Online and Offline

CLEF NewsREEL uses two tools supporting participants evaluating their news recommendation algorithms. First, we introduce a platform to access a stream of recommendation requests thus enabling A/B testing. Second, we present a framework that lets participants repeat recorded interaction thus allowing offline evaluation.

4.1 Online Evaluation Methods

NewsREEL lets participants connect with a continuous stream of requests in order to evaluate their recommendation algorithms online. The setting resembles the situation which industrial recommender systems face as they serve suggestions. The Open Recommendation Platform (ORP) lets participants access a request distribution interface. ORP receives recommendation requests by a variety of news publishers. Subsequently, ORP delegates requests randomly to linked recommendation servers. Such requests entail a variety of information. This includes references to the session, the news article currently displayed, browser settings, and keywords. Participants' systems ought to select a subset of permissible articles to return to the user. ORP takes the list and forwards it to the user. Subsequently, ORP monitors users' reactions and keeps track of click events. In this way, we gain insights on how well recommendation algorithms perform over time.

Multi-dimensional Objectives Businesses determine their success in part by their market share. Market share reduces to the number of visits in the context of online media. Visits signal attention which represents a valuable asset for marketing. Whenever users click on a recommended item, they prolong their session thus adding another visit. Consequently, businesses seek to determine the recommendation strategy yielding best expected chance of clicks. In other words, businesses maximize the *click-through rate* (CTR). Additionally, however, there are other aspects which we have to consider. In particular, we need to assure availability and scalability. Availability concerns the proportion of time during which the system can receive requests. This proportion may be limited by maintenance, model updating, and failures. Scalability concerns how well systems handle large numbers or sudden increases of requests. ORP reports an error rate for each system. This error rate reveals how many requests resulted in error states. Errors arise whenever systems delay their recommendations or return invalid items.

Expected Setting The contest allowed participants to operate multiple recommendation services simultaneously. ORP delegates requests randomly to responsive recommendation services. Consequently, we expect recommendation services with similar availability and error rate to receive similar numbers of requests. ORP has a fixed set of publishers assigned. This limits the total number of requests. The more algorithms participants deploy, the fewer requests each recommendation service receives. Experiences from previous editions of NewsREEL indicate that we can expect 5000 to $10\,000$ requests per day for recommendation services with high availability and low error rate. This corresponds to a mean request frequency of $0.06\,\mathrm{Hz}$ to $0.12\,\mathrm{Hz}$. Requests distribute unevenly across the day. As a result, we expect participants to experience considerably higher frequencies of more than $10\,\mathrm{Hz}$ at peak times.

4.2 Offline Evaluation Methods

The offline task allows participants to evaluate recommender algorithms in a replicable way. It enables the detailed debugging as well as the analysis of algorithms in predefined load scenarios. Due to the possibility to replicate the experiments exactly, the offline evaluation ensures the comparability of different recommender algorithms and the optimization of parameters.

Replaying Recorded Streams The sequence of messages in a stream often contains important information. In order to ensure a realistic evaluation, we preserve the message order (recorded in an online setting) also in the offline evaluation. We provide a component that, roughly spoken, replays the stream of messages. We preserve the order of the messages as well as the timestamps keeping the stream similar to the originally recorded stream as possible. The simulation of the stream ensures realistic simulation of the online stream. At every timeslot the recommender algorithms "knows" only the items the recommender would also "know" in the online evaluation.

Evaluation Method In the evaluation, we use a window-based approach. We do not use cross-validation, since cross-validation does not preserve order of the messages. Instead of the n-fold splitting used in cross-validation, we use a continuously growing training window. The window begins with the start of the simulated stream and grows continuously over time. The part of the stream consisting of the 5 minutes right after the training window is used as ground truth window. A recommendation for a user is handled as correct if the user reads the recommended article in the 5 minutes after the request.

CTR-Related Metrics In contrast to the online evaluation, there is no direct feedback from users. Thus, we have to define the Click-Through-Rate based on the log data collected in the online challenge. In order to decouple the offline evaluation from the recommender algorithms used while recording the offline dataset, we define the metric based on the impressions. Impressions characterize all events when users access news articles. They arise from search, browsing, and recommendations. Empirically, clicks occur in approximately 1of100 impressions. Thus, we expect at most a marginal bias by shifting our focus to impressions. Figure 1 illustrates the procedure.

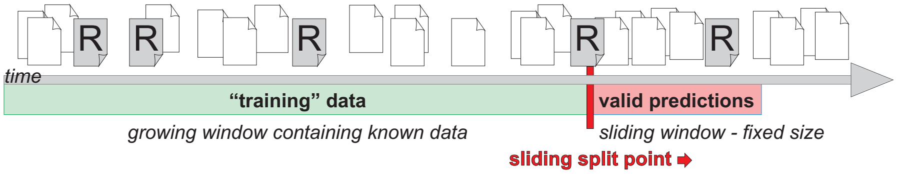
R
R
R
R
R
time
"training" data
valid predictions
growing window containing known data
sliding window - fixed size
sliding split point ➡

Fig. 1. The figure visualizes the calculation of the offline Click-through-Rate based on a simulated stream.

Metrics focusing on Technical Aspects Ensuring short response time as well as the scalability of the recommender algorithms are important requirements in the NewsREEL challenge. Based on the requirements we define metrics allowing us measuring the performance of the analyzed algorithms with respect to technical aspects. We use response time to determine how well algorithms scale to the load of requests.

Response time In order to ensure that recommendations can be seamlessly embedded into websites, they must be delivered within a predefined time limit. That is the motivation for analyzing the response time of the recommender algorithms in detail. Typically, the response time varies. We address this observation by calculating the distribution of response time values. The distribution expresses how frequently specific response times are measured. The distribution allows us to determine average and variance of response times. In addition, we compute the average response time and the fraction of requests that are not answered within the predefined time limit.

Offline Evaluation Framework The exact reproducibility of offline evaluation requires that all steps and all environmental parameters are exactly defined. In order to compare the technical complexity of different algorithms, the computational environment must be defined in a reproducible way. We address this issue by using the evaluation framework *Idomaar*[12]. The framework is a recommender system reference framework developed in the settings of the European Project CrowdRec[13]. It builds reproducible computing environments based on virtual machines having an exactly defined software environment based on PUPPET. The resources and all software components (and versions) available during the evaluation are clearly defined, ensuring that neither old software components nor remainders from earlier evaluation runs may distort the results. All steps of the evaluation are executed based on scripts ensuring that the complete evaluation is reproducible.

- **Architecture independence**. Participants can use their preferred environments. Idomaar provides an evaluation solution that is independent of the programming language and platform. The evaluation framework can be controlled by connecting to two given communication interfaces by which data and control messages are sent by the framework.
- **Effortless integration**. The interfaces required to integrate the custom recommendation algorithms make use of open-source, widely-adopted technologies: Apache Spark and Apache Flume. Consequently, the integration can take advantage of popular, ready-to-use clients existing in almost all languages.
- **Consistency and reproducibility**. The evaluation is fair and consistent among all participants as the full process is controlled by the reference framework, which operates independently from the algorithm implementation.
- **Stream management**. Idomaar is designed to manage, in an effective and scalable way, a stream of data (e.g., users, news, events) and recommendation requests.

[12] `http://rf.crowdrec.eu/`

[13] `http://www.crowdrec.eu/`

Advantages of Idomaar Idomaar automates the evaluation process. It implements a three-stage workflow: (i) data preparation, (ii) data streaming, and (iii) result evaluation. The Orchestrator controls the environment. This includes setting up virtual machines, regulating communication between components, and measuring aspects such as response times. The configuration of virtual machines is fully specified including hardware resources and installed software packages. Therefore, evaluations will reproduce identical results. In addition, manual mistakes are limited due to automated evaluation protocols.

4.3 Discussion

In this section, we introduced two tools supporting participants evaluating news recommendation algorithms. First, we discussed how ORP enables participants to connect to a stream of recommendation requests. This yields a similar experience to A/B testing. Second, we presented Idomaar which is designed to support the efficient, reproducible evaluation of recommender algorithms. Idomaar is a powerful tool allowing users to abstract from concrete hardware or programming languages by setting up virtual machine having exactly defined resources. The evaluation platform allows a high degree of automation for setting up the runtime environment and for initializing the evaluation components. This ensures the easy reproducibility of evaluation runs and the comparability of results obtained with different recommender algorithms. Idomaar supports the set-based as well as the stream-based evaluation of recommender algorithms. In NewsREEL Task 2, the stream-based evaluation mode is used. In contrast to most existing evaluation frameworks Idomaar can be used out of the box and, for evaluation, considers not only the recommendation precision but also the resource demand of the algorithms.

5 The Participant Perspective

In this section, we present an appraisal of CLEF NewsREEL from the participants' perspective. In particular, we discuss opportunities, validity, and fairness. A more detailed discussion of the analysis presented in this section can be found in [10].

5.1 Opportunities

CLEF NewsREEL provides a unique opportunity for researchers working on recommender systems. It enables researchers to test their algorithms in a real-world setting with real users and items. In addition, participants compete with one another. Thus, they get feedback on how their algorithms compare with competitors' algorithms. Further, participants get access to a large number of log files comprising interactions between users and items. They can conduct offline experiments with these data thus optimizing their system prior to deploying them. Researchers hardly have access to such conditions otherwise, making CLEF NewsREEL a unique form of benchmarking.

5.2 Validity and fairness

Participants seek to compare their algorithms with competing algorithms. They need to know how valid comparisons are in order to estimate how well their systems will perform in the future. Determining validity represents a challenging task. Unlike the operators of recommender systems, participants only perceive parts of the environment. Various effects can potentially bias observed performance. We distinguish operational and random biases, the latter resulting from random effects such as the dynamics in user and item collections. Operational bias refers to the result of operational choices of the evaluation framework, including those that lead to favoring some participants' systems over others, or delegating a disproportional number of requests from specific publishers to a few systems only. The latter in particular would skew results, as items originating from specific publishers have been found to receive a stronger user response.

Fairness of the competition is closely related to the validity of findings, especially when considering operational biases. A (limited) level of random bias due to dynamic fluctuations in user and item collections is to be expected, but it would be very useful to be able to quantify its influence. In the absence of biases, we would expect to observe similar performance of identical systems over sufficiently long periods of time. Therefore, we have applied a method of evaluation that is best described as A/A testing; unlike in the usual A/B testing, A/A testing subjects the users to different instances of the exact same algorithm. The instances were run in the same computer and the same environment; only the port numbers they used to interact with Plista were different. With this setup, we do not expect the ORP to treat the two algorithms differently, since their behavior should be identical. Since the exact same algorithm was used to generate the recommendations, we attribute differences in the responses by users to those recommendations to bias, and we analyze those differences to quantify its effect.

Experiment As participants, we conducted an experiment to estimate operational and random biases in CLEF NewsREEL. We set up two instances of the same recommendation algorithm, implementing an A/A testing procedure. We implemented a recency-driven recommender, which keeps the 100 most recently viewed items and suggests the five or six most recent upon request. Random biases may cause performance variations on a daily level. In the absence of operational biases, we may expect these performance measures to converge in the long-term. Both instances of the recency recommender have run in NewsREEL's editions 2015 and 2016. In 2015, the two instances ran from Sunday 12th April, 2015 to Monday 6th July, 2015, a total of 86 days. In 2016, both instances ran from Monday 22nd February, 2016 to Saturday 21st May, 2016, a total of 70 days. We considered only the recommendation requests and clicks of days on which the two instances of our algorithms ran simultaneously. Table 2 presents requests, clicks, and the CTR for both periods. The observed difference in CTR is small, $0.04\,\%$ in 2015 and $0.07\,\%$ in 2016, based on which we conclude that the evaluation does not show evidence of an operational bias. On the other hand, we notice a marginal level of random bias. Figure 2 shows the average CTR as a function of the number of days, for the year 2015 and Figure 3 for the year 2016. Initially, we observe fairly high levels of variance between both instances in 2015. Over time, the variance levels off and both instances of the algorithm approach a common level of $\approx 0.85\,\%$. In 2016, we observe the opposite trend in that the algorithms perform more similarly and diverge towards the end.

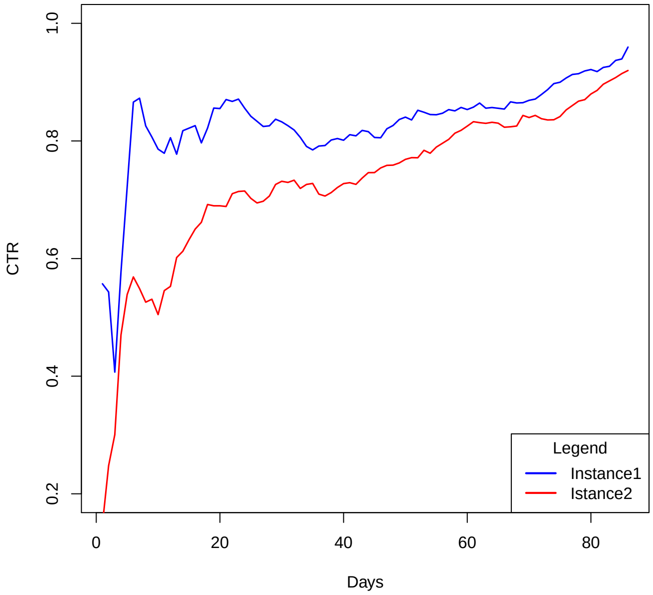

Fig. 2. The cumulative CTR performances of the two instances as they progress on a daily basis in 2015.

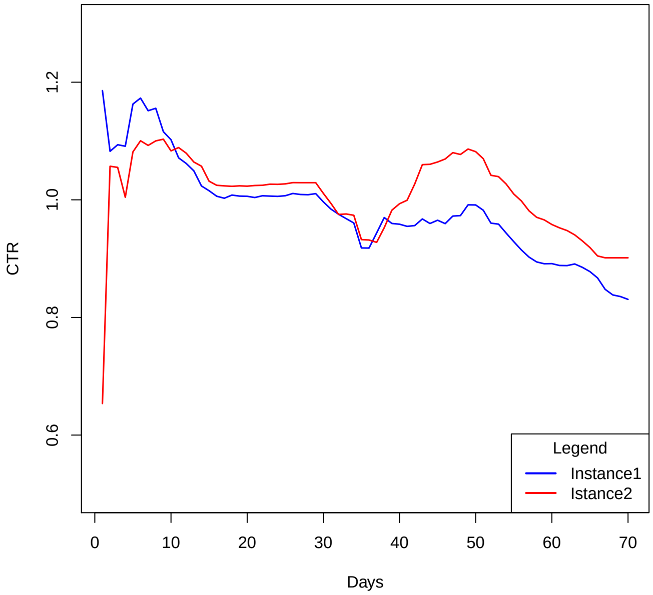

Fig. 3. The cumulative CTR performances of the two instances as they progress on a daily basis in 2015.

Table 2. Data collected by running two instances of the Recency recommender in the 2015 and 2016 editions of NewsREEL.

	2015			2016		
Algorithms	Requests	Clicks	CTR (%)	Requests	Clicks	CTR (%)
Instance1	90 663	870	0.96	450 332	3741	0.83
Instance2	88 063	810	0.92	398 162	3589	0.90

Log Analysis We noticed that A/A testing with two instances of the same algorithm results in performance variations, that, in 2015, smoothed out when observed over a sufficiently long period of time, but in 2016 showed divergence towards the end. We analyzed our log files from 2015 to identify two hypotheses to explain these variations. First, operational bias might induce an unfair setting, in which some algorithms naturally perform better than others. Alternatively, random bias due to the selection of users and items presented to each recommender may explain the performance variation observed.

Analyzing Recommendation Requests by Publisher: We look into the distribution of requests across publishers. In a fair competition, each participant will be subject to a similar distribution across publishers. We aggregated all requests on a publisher-level for both instances. Subsequently, we computed the Jensen-Shannon Divergence (JSD) metric to quantify the differences between both distributions. We obtained a divergence score of approximately 0.003, indicating that both instances received similar distributions of requests. At the level of a publisher, We conclude that we did not find a noticeable bias that would be attributed to operational design choices in the evaluation framework.

Analyzing Recommendation Requests and Responses at Item and User Levels: We investigate the overlap between the sets of users and items processed by both instances, by measuring their Jaccard similarity; high overlap would signal the absence of random biases. Comparison of the sets of items produced a Jaccard similarity of 0.318 whereas the sets of users resulted in a score of 0.220. Given the low overlap between users and items presented to both instances, we conjecture that the chance to observe the same user on both systems is relatively low (which can be explained by the limited number of events in each session). We note that the overlap is impacted by the fact that there are tens of other systems running simultaneously. The observed overlap is not inconsistent with the conclusion that user and item variation arises due to natural dynamics.

5.3 Discussion

In this section, we have discussed the NewsREEL challenge from the participants perspective. Our focus has been understanding the perspective that is accessible to the participants on whether or not the NewsREEL evaluation treats all participating algorithms fairly. We reported on the results of A/A testing conducted to estimate the level of variance in CTR for identical algorithms. We hypothesized that random effects or operational design choices could cause varying performances. We observed varying trends, in 2015 and 2016, in the cumulative performances of the two instances. In 2015, the variance diminished over time, but in 2016 the variance emerged later. We analyzed the logs of our participating systems to determine which kind of effect produced the variance. We found that requests were distributed equally across publishers for both instances. On the basis of this observation we were able to conclude, from the participant perspective, that operational design choices are unlikely to have caused the variance. Instead, we observed that collections of users and items differed between both instances.

From the participants perspective and the current setup, it is possible to conduct partial investigation into possible operational biases, have a reasonable estimate of the impact of those causes on the performance of a participating system. We conclude that participants do have the means to assure themselves of NewsREEL's fairness using only information available from the participant's perspective. We note, however, that an exhaustive investigation of all possible operational biases is either too complicated and/or impossible from the participant's perspective. For example, operational biases could be implemented at the level of pairing logged-in and logged-off users to different teams or participant systems, pairing some item categories to some participants or systems, and disfavoring one system on the basis of response and other network factors. The possibility to explore some of the biases is somewhat hampered by the fact that participants do not receive direct information on whether their recommendation are clicked. It is possible to extract a system's recommendation clicks from the logs, but it requires expensive implementation, and is also subject to error. The error is in turn dependent on the way in which the participant chooses to implement the mapping of recommendations to clicks.

6 Evaluation Results

At the time of writing, we have not yet received participants' working notes. This section highlights preliminary results observed for baseline method and some additional systems contributed by the organizers.

6.1 Task 1: Online Competition

Participants are required to provide suggestions maximizing the expected response rate. For this reason, we monitor how often users click recommended articles. Figure 4 shows the relation of clicks to requests for all participants over the stretch of three weeks. We note that all recommendation services fall into the range from $0.5\,\%$ to $1.0\,\%$. Further, we observe that some recommendation services obtained considerably larger numbers of requests. These systems have had a higher availability than their competitors. They produce less errors by providing valid suggestions in a timely manner. They produce less errors by timely providing valid suggestions. Figure 5 illustrates how the error rate relates to the number of requests received. Participants with high error rates received fewer requests than those who managed to keep their error rates low. We note that additional factors affect the number of requests. Some participants had low error rates but still received few requests. Their systems had not been active for as long as their counterparts with higher number of requests.

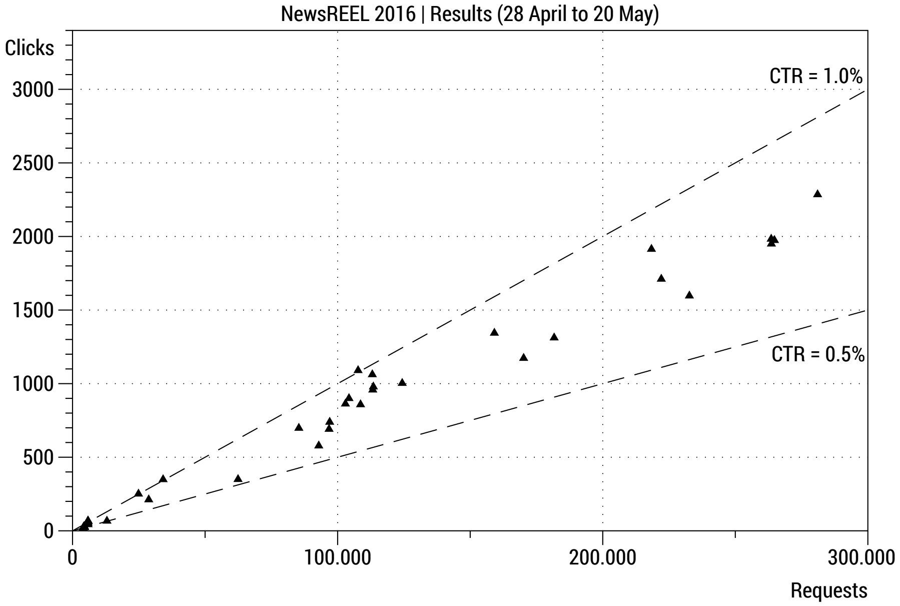

Fig. 4. Participating recommendation services delivered suggestions upon requests for period of three weeks. The figure shows how recipients responded in terms of clicks. Each triangle refers to a specific algorithm.

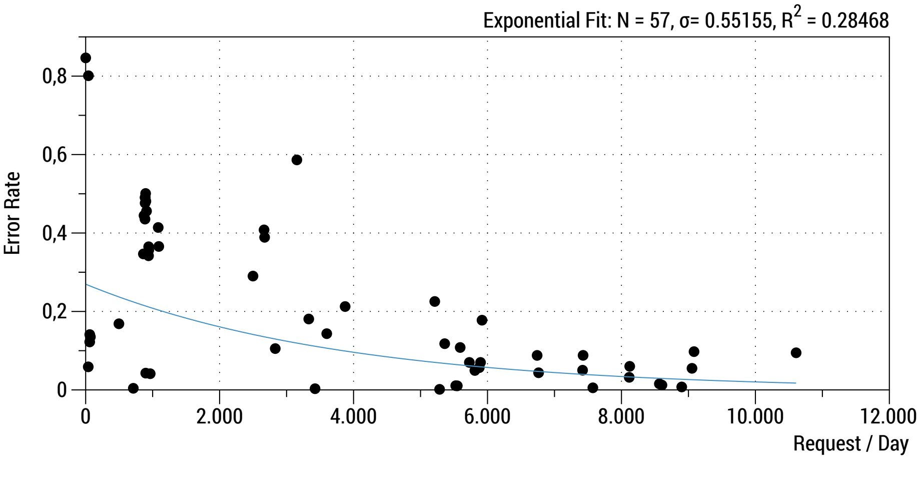

Fig. 5. Errors occur when recommendation services fail to timely return valid suggestions. ORP controls request delegation accordingly. The figure shows that the more errors systems produce, the fewer requests they receive.

6.2 Task 2: Offline Evaluation

Responding quickly to requests is essential for successful recommendations. We deployed two identical recommendation services to determine how network latency affects response times. Recommender service A replied from within the local subnet. Recommender service B replied from another net. Figure 6 illustrates the effect on response time. The orange line refers to recommender service A while the green line represents recommender service B. Both systems exhibit a bi-modular shape. System A has a higher peak at low response times. System B appears shifted toward higher response times. This illustrates the latency effect.

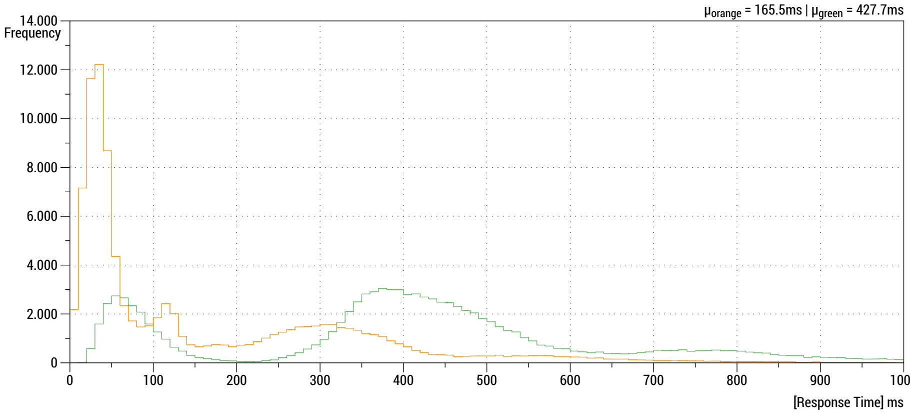

Fig. 6. Illustration of response times with identical implementation. The orange curve represents a system deployed in the local subnet whereas the green curve's underlying system operates from outside the local subnet. Network latency shifts the green distribution to the right.

6.3 Comparing Online and Offline

Online and offline evaluations are frequently considered separately. Academia targets reproducible results on offline data sets. Businesses monitor user feedback continuously online. NewsREEL gives researchers the opportunity to compare performances in both regimes. Participants observe their performance in Task 1 and Task 2. Both settings support multi-dimensional evaluation. Task 1 reports click-through rates to assess how well systems cater to user preferences. Task 2 considers how accurately systems predict impressions. Impressions occur on various ways including browsing and search. Conversely, clicks are directly linked to recommendations. Thus, Task 2 is less affected by presentation biases of user interfaces than Task 1. Users might not perceive recommendations displayed online. Still, they can access articles that have been recommended. In contrast to Task 1, Task 2 would consider such user reading events as successful recommendations. As a result, we expect varying results as we compare online with offline accuracy. The question remains whether offline and online accuracy track each other. Task 1 determine reliability and scalability in terms of error rates. Recommendation services failing to return valid results obtain high error rates. Technical issues beyond the recommendation algorithm contribute to error rates. For instance, hardware defects, system maintenance, and network malfunctions induce errors not related to the recommendation algorithm. Task 2 simulates critical scenarios as it delegates requests at maximum capacity to the recommender system. This neglects the presence of periods with relatively low load in the online setting. Recommender systems only reply to a subset of requests in Task 1. Contrarily, Task 2 requires recommender systems to provide suggestions for all requests. As a consequence, systems can succeed online even though they exhibit inferior response times offline. Additionally, the offline evaluation lets participant detect flaws in their implementations.

6.4 Participation

In this year's edition, 48 participants registered for NewsREEL. Thereof, 46 signed up for Task 1 whereas 44 enlisted in Task 2. Multiple participants registered from the Netherlands (6), India (5), Turkey (4), Germany (3), United Kingdom (3), China (2), France (2), Norway (2), and Tunisia (2). Nine participants received virtual machines to deploy their recommendation service onto. This was meant to limit disadvantages due to network latency or the lack of hardware. We observed 21 teams competing with 73 different algorithms during the evaluation period of Task 1. In contrast, seven teams conducted offline experiments and shared their insights in form of working notes.

6.5 Discussion

The NewsREEL lab gives participants the opportunity to evaluate news recommendation algorithms. Analyzing the implemented strategies and discussing with the researchers, we find a wide variety of approaches, ideas, and frameworks. The performance as well as the response time of the algorithms varies with the algorithms and contexts. Thus, the performance ranking may change during the course of a single day. In order to compute a ranking, the challenge uses a comprehensive evaluation period (4 weeks in Task 1) and a huge dataset (consisting of ≈ 100 million messages in Task 2) respectively. The baseline recommender performs quite successfully, being always among the best 8 recommender algorithms. We observe that high error rates and low availability lead to few requests. This hampers comparing participants' systems. We cannot be sure that we can reproduce the ranking in a different context. For instance, the same set of recommenders performs differently 6 months later when an important event shapes users' interests in a different way. The CTR ranges from $0.5\,\%$ to $1.0\,\%$.

7 Conclusion and Outlook

Suggesting news articles challenges recommender systems. Similarly to other domains, news recommender systems face streams of recommendation requests as visitors continue to interact with digital news websites. Streams make it challenging to update recommendation models and they also require scalable infrastructures. Additionally, systems have limited information about their users. Frequently, they lack any form of user profiles and rely on tracking them by session cookies. Furthermore, stories are continuously added to the collection of existing news items. For these reasons, establishing reproducible evaluation protocols is an ongoing struggle. Innovative strategies are needed to deal with this cumbersome problem.

CLEF NewsREEL provides participants with a unique opportunity to contribute ideas. Participants gain access to an operating news recommender system thus obtaining live feedback by actual users. In addition, they receive a large-scale data set covering news and interactions with news over a stretch of two months. Both tasks address not only preference modeling, but additionally they challenge participants to consider technical aspects such as scalability, reliability, and complexity. Other contests hardly address such factors even though businesses cannot ignore them. Task 1 measures the CTR as well as error rates. Task 2 measures how well algorithms predict future interactions as well as response times. By taking part in both tasks, participants can determine how well offline results transfer to online setting and what we can learn from them. This year's edition of NewsREEL allowed participants to evaluate their systems for several weeks online. Receiving several thousands request a day suffices to draw meaningful conclusions. However, we have to keep in mind that user preferences as well as news articles are continuously evolving. For this reason, algorithms providing the best suggestions today might fall behind in the future. Participants needed time to accustom themselves to ORP, which, in a yearly benchmarking cycle, means there is less time left for a long evaluation period.

Participants had the opportunity to provide feedback about the experiences with NewsREEL in an open conference call. We summarize what they suggested as improvements for future editions of NewsREEL. ORP ought to become more transparent and functional. As discussed above, currently, it is hard to track systems' success in terms of recommendations which are presented to users and then clicked. ORP does not explicitly provide references to recommendation requests when informing about click events. Instead, participants have to keep track of their recommendations and compare them with events from the continuous stream of messages. In addition, ORP currently disables recommenders producing errors without notifying participants. Thereby, participants' system availability decreases leading to fewer recommendation requests. Having been notified, participants could repair their system more quickly. In the future, we would like to allow for more time evaluating in order to have a more insightful comparison between offline and online performance. Additionally, we will clarify procedures and provide additional support for participants interested in offline evaluation. We plan to provide a ready-to-use installation of Idomaar on Amazon's S3 platform facilitating system setup.

Acknowledgments

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

References

1. K. Balog, L. Kelly, and A. Schuth. Head First: Living Labs for Ad-hoc Search Evaluation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1815–1818, New York, NY, USA, 2014. ACM.
2. J. Blomo, M. Ester, and M. Field. RecSys Challenge 2013. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 489–490, 2013.
3. T. Brodt and F. Hopfgartner. Shedding light on a living lab: the CLEF NEWSREEL open recommendation platform. In *IIiX '14*, pages 223–226, 2014.
4. J. Chen, R. Nairn, L. Nelson, M. S. Bernstein, and E. H. Chi. Short and Tweet: Experiments on Recommending Content from Information Streams. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 1185–1194, 2010.
5. Y. Chen, P. Berkhin, B. Anderson, and N. R. Devanur. Real-time Bidding Algorithms for Performance-based Display Ad Allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1307–1315, 2011.
6. E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, and W. Nejdl. Real-time Top-n Recommendation in Social Streams. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 59–66, 2012.
7. M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. In *RecSys'11*, pages 133–140. ACM, 2011.
8. Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A Free Recommender System Library. In *RecSys'11*, pages 305–308. ACM, 2011.
9. F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. Offline and Online Evaluation of News Recommender Systems at swissinfo.ch. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 169–176, 2014.
10. G. Gebremeskel and A. de Vries. Random Performance Differences Between Online Recommender System Algorithms. (Manuscript submitted for publication), 2016.
11. F. M. Harper and J. A. Konstan. The Movielens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, Dec. 2015.
12. 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, 2015.
13. F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, and T. Heintz. Benchmarking news recommendations in a living lab. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*, pages 250–267, 2014.
14. B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz. The plista Dataset. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, NRS '13, pages 16–23, New York, NY, USA, 2013. ACM.
15. B. Kille, A. Lommatzsch, R. Turrin, A. Serény, M. Larson, T. Brodt, J. Seiler, and F. Hopfgartner. Stream-based Recommendations: Online and Offline Evaluation as a Service. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, pages 497–517, 2015.
16. A. Lommatzsch and S. 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.

17. A. Maksai, F. Garcin, and B. Faltings. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 179–186, New York, NY, USA, 2015. ACM.
18. B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. The Million Song Dataset Challenge. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 909–916, 2012.
19. T. D. Noia, I. Cantador, and V. C. Ostuni. Linked Open Data-enabled Recommender Systems: ESWC 2014 Challenge on Book Recommendation. In *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 129–143, 2014.
20. A. Said and A. Bellogín. Rival: A Toolkit to Foster Reproducibility in Recommender System Evaluation. In *RecSys'14*, pages 371–372, New York, NY, USA, 2014. ACM.
21. M. Tavakolifard, J. A. Gulla, K. C. Almeroth, F. Hopfgartner, B. Kille, T. Plumbaum, A. Lommatzsch, T. Brodt, A. Bucko, and T. Heintz. Workshop and Challenge on News Recommender Systems. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 481–482, 2013.