Overview of CLEF NewsREEL 2015: News Recommendation Evaluation Lab

Benjamin Kille[1], Andreas Lommatzsch[1], Roberto Turrin[2], András Serény[3], Martha Larson[4], Torben Brodt[5], Jonas Seiler[5], and Frank Hopfgartner[6]

[1] TU Berlin, Berlin, Germany
{benjamin.kille,andreas.lommatzsch}@dai-labor.de
[2] ContentWise R&D - Moviri, Milan, Italy
roberto.turrin@moviri.com
[3] Gravity R&D, Budapest, Hungary
sereny.andras@gravityrd.com
[4] TU Delft, Delft, The Netherlands
m.a.larson@tudelft.nl
[5] Plista GmbH, Berlin, Germany
{torben.brodt,jonas.seiler}@plista.com
[6] University of Glasgow, Glasgow, UK
frank.hopfgartner@glasgow.ac.uk

Abstract. News reader struggle as they face ever increasing numbers of articles. Digital news portals are becoming more and more popular. They route news items to visitors as soon as they are published. The rapid rate at which new news is published gives rise to a selection problem, since the capacity of new portal videos to absorb news is limited. To address this problem, new portals deploy news recommender systems in order to support their visitors in selecting items to read. This paper summarizes the settings and results of CLEF NewsREEL 2015. The lab challenged participants to compete in either a "living lab" (Task 1) or an evaluation that replayed recorded streams (Task 2). The goal was to create an algorithm that was able to generate news items that users would click, respecting a strict time constraint.

Keywords: news recommendation, recommender systems, evaluation, living lab

1 Introduction

News recommendation continues to draw the attention of researchers. Last year's edition of CLEF NewsREEL [4] introduced the Open Recommendation Platform (ORP) operated by plista. ORP provides an interface to researchers interested in news recommendation algorithms. They can easily plug in their algorithms and receive requests from various news publishers. Subsequently, the systems records recipients' reaction. This feedback allows participants to improve their algorithms. In contrast to traditional offline evaluation, this "living lab" approach reflects the application setting of an actual news recommender system.

Participants must satisfy technical requirements, and also face technical challenges. These include response time restrictions, handling peaks in the rate of requests, and handling continuously changing collections of users and items. Conceptually, the evaluation represents a fair competition. All participants have the same chance to receive a request since ORP distributes them randomly. Random distribution helps avoid selection bias.

In addition to providing fair comparison, the NewsREEL challenge would like to level the playing field for all participants. Specifically, the environments in which participants operate their recommendation algorithms vary widely. First, participants' servers have to bypass varying distances to communicate with ORP. ORP is located in Berlin, Germany. Participants from America, East-Asia, or Australia face additional network latency compared to participants from Central Europe. Their performance might suffer from failing to serve some requests only due to latency. Second, participants use different hardware and software to run their algorithms. Suppose a participants has access to a high-performance cluster. Another participant runs their algorithm on a rather old stand-alone machine. Is it fair to compare the performance of these participants? The latter participant may have developed a sophisticated algorithm not perform well in the competition since they cannot meet the response time requirements.

This year's edition of CLEF NewsREEL seeks to add another level of comparison to news recommendation. Our aim is to be able to fairly measure systems with respect to non-functional requirements, and also allow all participants to take part in the challenge on equal footing. We continue to offer the "living lab" evaluation with ORP as Task 1. In addition, we introduce an offline evaluation targeted at measuring additional aspects. These aspects include complexity and scalability. In Task 2, we provide a large data set comprising interactions between users and various news portals in a two-month time span. Participants are able to re-run the timestamped events to determine how well their system scales. We introduce IDOMAAR, a framework designed to measure technical parameters along with recommendation quality. IDOMAAR instantiates virtual machines. Since these machines share their configuration, we obtain comparable results. These results do not depend on the actual system. We kept the interfaces similar to ORP's such that participants could re-use their algorithms with only a minor adaption effort.

The remainder of this lab overview paper is structured as follows. In Section 2, we introduce the two subtasks of NewsREEL'15. The results of the evaluation are presented in Section 3. Section 4 concludes the paper.

2 Lab Setup

CLEF NewsREEL'15 consisted of two subtasks. Task 1 was a repetition of the online evaluation task ("Task 2") of NewsREEL'14. In Section 2.1, we briefly introduce the recommendation use case of this task. For a more detailed overview, the reader is referred to [4]. Section 2.2 introduces the second subtask that focuses on simulating constant data streams, hence allowing evaluation of real-time

recommenders using an offline data set. For a more detailed overview of this use case, we refer to [6].

2.1 Task 1: Benchmark News Recommendations in a Living Lab

This task implements the idea of evaluation in a living lab. As such, participants were given the chance to directly interact with a real-time recommender system. After registering with The Open Recommendation Platform (ORP) [1] provided by plista GmbH, participants receive recommendation requests from various websites offering news articles. Requests were triggered by users visiting those websites.

The task followed the idea of providing evaluation as a service [3]. Participants had access to a virtual machine where they could install their algorithm. The recommender system forwarded the incoming requests to a random virtual machine which produced the recommendation to be delivered to the requester. The random choice was uniformly distributed over all participants. Alternatively, participants could set up their own server to respond to incoming requests.

As a fixed response time limitation was set, the participants experienced typical restrictions for real-world recommender systems. Such restrictions pose requirements regarding scalability and computational complexity for the recommendation algorithms.

ORP monitored the performance of all participants during the challenge duration by measuring the recommenders' click through rate (CTR). CTR represents the ratio of clicks by requests. Participants had the chance to continuously update their parameter settings in order to improve their performance levels. Results were published on a regular basis to allow participants to compare their performance with respect to baseline and competing approaches. An overview of the results is given by Kille et al. [6], and also in this paper in Section 3.

2.2 Task 2: Benchmarking News Recommendations in a Simulated Environment

For the second task, we employed the benchmarking framework Idomaar[7] that makes it possible to simulate data streams by "replaying" a recorded stream. The framework is being developed in the CrowdRec project[8] It makes it possible to execute and test the proposed news recommendation algorithms, independently of the execution framework and the language used for the development. Participants of this task had to predict users clicks on recommended news articles in simulated real-time. The proposed algorithms were evaluated against both functional (i.e., recommendation quality) and non-functional (i.e., response time) metrics. The data set used for this task consists of news updates from diverse news publishers, user interactions and clicks on recommendations. An overview of the features of the data set is provided by Kille et al. [5].

[7] `http://rf.crowdrec.eu/`
[8] `http://crowdrec.eu/`

3 Evaluation

In this section, we detail results of CLEF NewsREEL 2015. We start by giving some statistics about the participation in general. Then, we discuss the results for both tasks.

3.1 Participation

Forty-two teams registered for CLEF NewsREEL 2015. Of these teams, 38 teams expressed interest in both tasks. A single participant registered for Task 2 only. Three teams wanted to focus on Task 1. Participating teams distribute across the world including all continents except Australia. ORP's operators, plista, provided five virtual machines to participants who were located far from Berlin, Germany. Without these machines participants would have faced issues with network latency, already discussed above.

Nine teams actively competed in Task 1. The competition's schedule consisted of three evaluation time frames: 17–23 March, 7–13 April, and 5 May to 2 June 2015. Seven out of nine teams competed in all three periods. Team "irit-imt" stopped competing after the second period. Team "university of essex" entered the competition as the final period started. Each team could operate several recommendation services. Each recommendation service obtained a similar volume of requests if active for similar times. We received a submission describing the idea and results of team "cwi" [2].

3.2 Baselines

Within the evaluation, we sought to obtain comparable results. Baselines allow us to determine how well a participant performs relative to a very basic approach. In last year's edition of NewsREEL, we established the baseline discussed in [4]. This baseline allocates an array of fixed length for item references. As we observe visitors interacting with the news portal, we put item references into the array. As we receive a recommendation request, we reversely iterate the array returning the first item references that are unknown to the target user. In this way, the baseline considers both freshness and popularity. We operated the baseline on two machines, "riemannzeta" and "gaussiannoise", which represented two different levels of machine power. The team "riemannzeta" administered a virtual machine with a dual-core Intel Xeon X7560 @ 2.27 GHz, 2 GB of RAM, and 8 GB hard drive. The team "gaussiannoise" operated a more powerful virtual machine with a quad-core Intel Xeon X7550 @ 2.0 GHz, 8 GB of RAM, and 26 GB hard drive. We released the baseline approach in form of a tutorial. Participants could take advantage of the baseline. Additionally, we sought to establish comparability with respect to last year's winner. Last year's winning approach has been documented in [7]. The approach competed as "abc" and in a slightly adjusted version as "artificial intelligence", also described in [7].

3.3 Results

Task 1 We observed nine teams actively participating throughout CLEF NewsREEL 2015. We recorded the performance of participants during three periods: 17–23 March, 7–13 April, and 5 May - 2 June 2015. The former two periods span a week each; the latter amounts to four weeks of data. The schedule had intentional gaps between the periods allowing participants to improve their algorithms. Table 1 summarizes the performances on team level. Each team has the number of requests (R), number of clicks (C), and their proportion (C/R) assigned for each of the three periods. Fields with 'n/a' refer to lack of participation. The highest average CTR per time slot is typeset in bold face. We observe that these values increased as the competition progressed. This indicates that teams managed to improve their recommendation algorithms over time. In addition, this could signal that teams learned to adjust their systems better to the challenge's requirements. Team "irit-imt" received 44 clicks at 5597 requests leading to the highest CTR (0.79 %) in the time slot from 17–23 March. Team "abc" received 56 clicks at 6483 requests obtaining a CTR of 0.86 % surpassing all competitors in the time slot from 7–13 April. Team "artificial intelligence" collected 302 clicks at 23 756 requests resulting in a CTR of 1.27 % in the final four week time slot.

Table 1. We present the results of nine participating teams. Each participant could operate several algorithms simultaneously. Results are aggregated over all algorithms. The evaluation includes three periods. We report the number of clicks (C), requests (R), and their relation (C/R). We highlight the highest CTR for each interval by bold typeface.

	17–23 March			7–13 April			5 May – 2 June		
Team	**C**	**R**	**C/R**	**C**	**R**	**C/R**	**C**	**R**	**C/R**
abc	73	9740	0.75%	56	6483	**0.86%**	349	30649	1.14%
artificial intelligence	71	10234	0.69%	49	6479	0.76%	302	23756	**1.27%**
cwi	161	24644	0.65%	130	22767	0.57%	1082	149544	0.72%
gaussiannoise	55	10063	0.55%	44	6515	0.68%	249	31343	0.79%
insight-centre	26	6833	0.38%	27	6500	0.42%	48	28857	0.17%
irit-imt	44	5597	**0.79%**	63	9481	0.66%	n/a	n/a	n/a
riadi-gdl	0	26	0.00%	45	6303	0.71%	177	27412	0.65%
riemannzeta	50	7684	0.65%	23	3833	0.60%	185	22064	0.84%
university of essex	n/a	n/a	n/a	n/a	n/a	n/a	17	5562	0.31%

Each participant could simultaneously operate several recommendation engines. Some participants took advantage of this offer. Consequently, those teams accumulated considerably more requests than others. Figure 1 illustrates the performance of individual algorithms. We present performance on a plane defined by the number of clicks and requests. A point on this plane refers to a specific

CTR. Points' colors refer to the respective team. The teams "cwi" and "riadi-gdl" deployed several algorithms. Two lines depict two CTR levels. A drawn through line marks the 1.0 % level. A dashed line represents the 0.5 % level. The illustration confirms that teams "abc" and "artificial intelligence" outperformed their competitors.

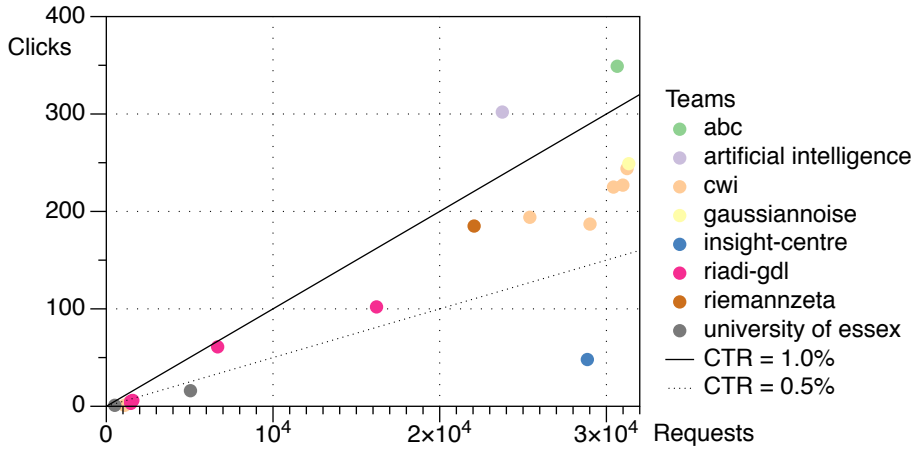

Fig. 1. Team were eligible to run several algorithms simultaneously. We observe some teams operating various recommenders. Teams "abc" and "artificial intelligence" managed to achieve a CTR of more than 1 %.

We investigate how individual algorithms perform over time. Figure 2 displays 16 algorithms' CTR relative to the average CTR over the final evaluation period's 28 days. Areas below 0 indicate a CTR lower than the average CTR of that day. Areas above 0 represent days with above average CTR. First, we observe that only a subset of algorithms ran throughout the period. Algorithms A, C, E, and K operated only scarcely. Algorithms F ("artificial intelligence") and J ("abc") managed to perform above the average CTR on almost all days. The majority of algorithms' CTR fluctuates around the system's average CTR. This confirms the difficulty inherent to news recommendation. The choice of an algorithm may depend on factors which are subject to change.

The competition featured a variety of news publishers. Some provide general as well as regional news. Other news portals specialize on topics such as sports or information technology. Figure 3 relates 16 competing algorithms with four major publishers. Publishers "418" (`www.ksta.de`) and "1677" (`www.tagesspiegel.de`) provide general and regional news. Publisher "35774" (`www.sport1.de`) targets sport-related news stories. Publisher "694" (`www.gulli.com`) presents information technology news. Combined, they account for ≈ 85 % of recommendation requests. The heatmap illustrates higher CTR with darker shades. CTR ranges up to 2.5 % for some combinations of publishers and algorithms. We

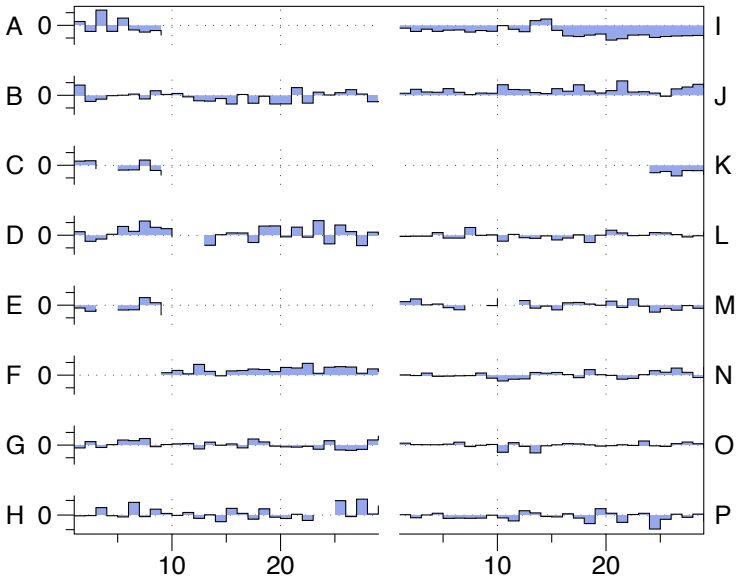

Fig. 2. We compare the performance of 16 algorithms over the final evaluation period's span of 29 days. We compute the average CTR for each day. Subsequently, we subtract the result from each algorithm's individual CTR for the same day. The labels map to teams as follows: A–E → "riadi-gld", F → "artificial intelligence", G → "gaussiannoise", H → "riemannzeta", I → "insight-centre", J → "abc", K → "university of essex", and L–P → "cwi".

observe that publishers "694" and "1677" have lesser CTR for almost all algorithms compared to "418" and "35774". This might be partially due to how the publishers present the recommendations. Some presentation might draw more attention toward the suggested articles than other. The top-performing algorithms "andreas" (team "abc") and "Recommender" (team "artificial intelligence") achieve the relatively highest CTR independent of the publisher.

We expect a recommendation service's reliability to affect the overall performance. Failing to serve plenty of requests will negatively affect CTR. Successfully suggesting news items will harness valuable feedback to further improve the recommendation algorithm. Figure 4 contrasts CTR and error rates observed during the final evaluation period. CTR refers to the ratio of clicked suggestions to received requests. Error rates reflect the proportion of requests that could not be served by the algorithm. Performances are colored with respect to the team operating the recommendation service. Most teams managed to keep error rates below 10 % with the exceptions of "riemannzeta", "riadi-gdl", and "university of essex". Remarkably, team "riadi-gdl" achieved a CTR of $\approx 0.9\,\%$ at an error rate of $\approx 53\,\%$. This indicates that their algorithm frequently failed to provide suggestions. Simultaneously, the suggestions given were particularly relevant to the

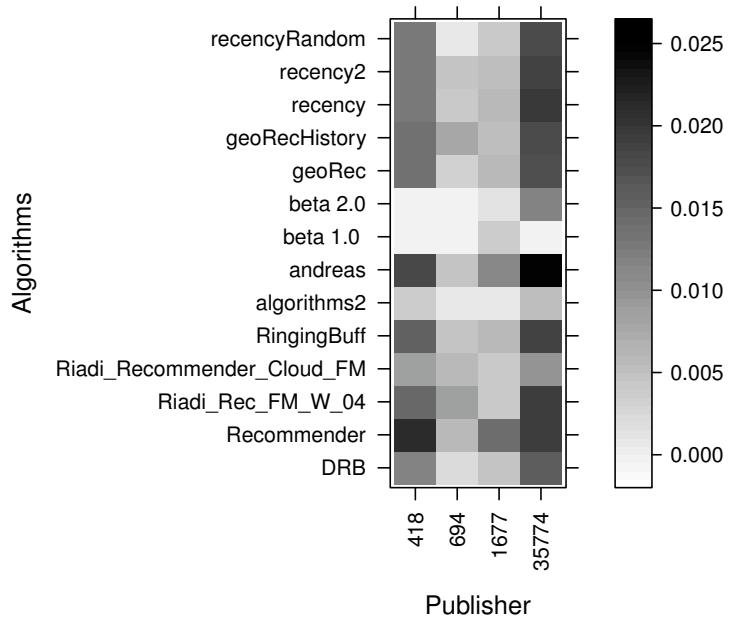

Fig. 3. The heatmap shows the Click-Through-Rates observed for combinations of algorithms and publishers. The four publishers account for $\approx 85\,\%$ of requests. Publishers "418" and "35774" obtain a higher CTR compared to "694" and "1677" on average.

recipients. Conversely, team "insight-centre" achieved a rather low error rate of $\approx 5.4\,\%$. Still, their CTR did not exceed $0.2\,\%$. Thereby, we conclude that while reliability can affect CTR, we have to consider additional factors. We note the difference in computing power between the baselines "riemannzeta" and "gaussiannoise" described in Section 3.2 The more powerful "gaussiannoise" achieved an error rate close to 0. In constrast, "riemannzeta" failed to respond to $\approx 16\,\%$ of its requests.

Task 2 The offline evaluation (based on a dataset recorded in July and August 2015) enables the reproducible evaluation of stream-based recommender algorithms. Having complete knowledge about the data set allows us to implement new baseline strategies. In addition to the baseline recommender used in Task 1, we implemented the "optimal" recommender. The recommender searches in the data set the items that will be rewarded for the current request by the evaluation component. This strategy used knowledge about the future. Thus, the strategy is not a recommender algorithm; it only implements a data set look-up. Consequently, this strategy cannot be used in the online "live" evaluation. Nevertheless

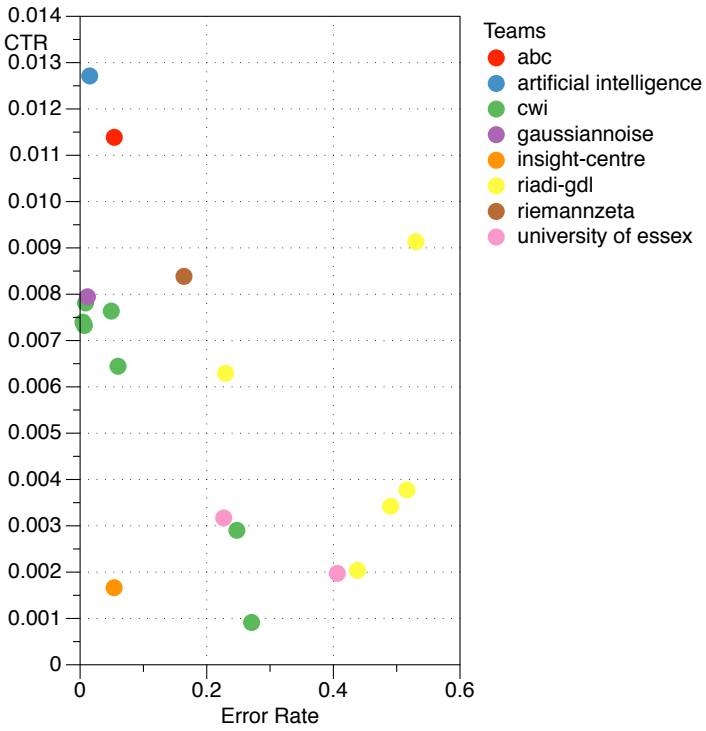

Fig. 4. The figure illustrates the relation between error rates and CTR as observed in the final evaluation period. Algorithms are colored according to their team membership. CTR refers to the ratio of clicks to requests. Error rates represent the proportion of requests which could not be served by the algorithm.

the measured CTR of the optimal recommender algorithm is interesting since the strategy allows us to measure the upper bound for the CTR in the analyzed setting.

Figure 5 shows the maximal achievable CTR for the three different domains in the offline dataset. The graphs show that the CTR varies highly from day to day. In addition, the graphs show that the average offline CTR for each of the analyzed news portals is specific for each of the portals. This can be explained by the different user groups and the differences in the number of messages per day. Due to the definition of the offline CTR, the expected CTR correlates with the number of messages forwarded as requests to a participant.

The evaluation with respect to scalability focused on maximizing the throughput. Since the teams in the competition used different hardware configurations, the measured results cannot be compared directly. A common optimization objective that has been addressed by the teams working on Task 2 is the effective synchronization of concurrently executed threads. This can be reached by using

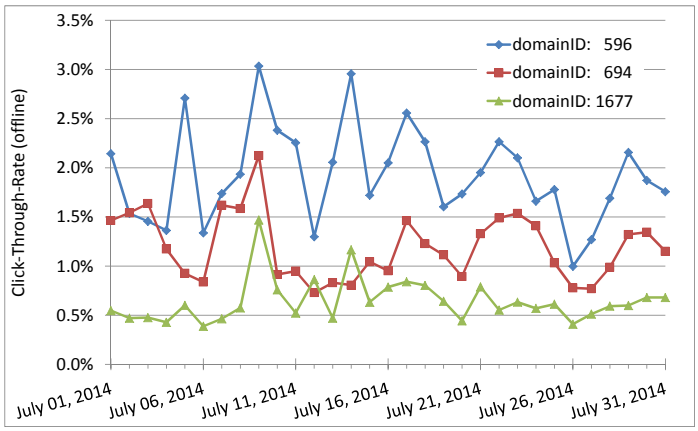

Fig. 5. The figure visualizes the offline CTR for the optimal "recommender algorithm". The optimal recommendation strategy is implemented by looking up the items that will be rewarded by the evaluator. The strategy defines the upper bound of the CTR reachable in Task 2.

highly optimized data structures (such as concurrent collections or Guava[9]) [8] or by using frameworks for building asynchronous, distributable systems [10]. Distributing a recommender algorithm over several machines adds extra overhead but gives a high degree for flexibility.

For the next year, we plan to use standardized virtual machines for the scalability evaluation, ensuring that all teams run the algorithms on exactly the same "virtual" hardware. In order to hide the complexity of building the evaluation environment, we plan to improve the Idomaar framework[10] and facilitate getting started with it.

3.4 Submissions

We received two submissions detailing the efforts of two teams. Gebremeskel and de Vries [2] explored the utility of geographic information. They hypothesize that visitors have special interest in news stories about their local community. The implement a recommender which leverages geographic data when matching visitors and news articles. We refer to their results as team "cwi".

Verbitskiy, Probst, and Lommatzsch [10] developed a most-popular recommender. Their investigation targets scalability. They use the Akka framework to benefit from concurrent message passing. They conducted their evaluation outside the final evaluation period. Still, they managed to obtain higher CTR than the continued baselines.

[9] `https://github.com/google/guava`
[10] `https://github.com/crowdrec/idomaar`

3.5 Discussion

NewsREEL aims to discover strategies that filter relevant news articles. Last year's edition introduced a "living lab" setting. This allows participants to evaluate their algorithms with actual users' feedback. This year's edition extended the previous setting. We developed the IDOMAAR framework. It not only keeps track of recommendation quality but records other performance metrics.

We continued competing with our baseline and last year's winning approach in order to demonstrate the ability of approaches to improve over both a basic system, and also the state of the art. Task 1 provided results which confirmed last year's findings. The baseline proved to be hard to beat. Last year's winner re-claimed the title. What produced this success story? Which factors determine the superior recommendation quality of the "artifcial intelligence" approach?

A team might have an advantage as it receive a larger or lower volume of requests than its competitors. We observed a comparable volume of requests for all algorithms active for the full evaluation period. These algorithms collected on average ≈ 1000 requests per day. The few exceptions with less requests were exactly those teams exhibiting higher error rates. Table1 shows requests on team level. Teams running several algorithms simultaneously have more request in total. Nevertheless, individual algorithms obtained similar shares of requests considering error rates and periods of inactivity. Has "artificial intelligence" received disproportionately many requests of visitors disproportionately likely to click? In that case, we would expect to observe varying performances at different days and on different publishers. In other words, we assume only marginal chances of receiving a specific subset of visitors consistently throughout time and publishers. Contrarily, Figure 2 shows consistent performance over average for almost all days. Similarly, Figure 3 lacks evidence for variations with respect to publishers. Is "artificial intelligence" running more reliably than its competitors? In fact, Figure 4 shows extremely low error rates. On the other hand, competitors including "gaussiannoise" and "cwi" achieve similar error rates but fall behind with respect to CTR. We conclude that combining popularity, freshness, and trend-awareness gives "artificial intelligence" a competitive advantage. Neither chance, bias, nor reliability explain the superior performance over four weeks.

We observed team "riadi-gdl" achieving the third best performance for an individual algorithm. This algorithms suffered from high error rates. We lack knowledge of the approach as we have not receive a working note for this performance. Still, it appears to involve promising algorithms which we would like to see more from in the future. Compensating the errors, the approach could potentially achieve even higher CTR than "artificial intelligence".

4 Conclusion

CLEF NewsREEL 2015 has been an interesting challenge motivating teams to develop and benchmark recommender algorithms online and offline. An addition to the online evaluation focused on the maximizing the CTR, the offline task

(Task 2) also considered technical issues (scalability, throughput). This year, the participating teams tested several different approaches for recommending news, ranging from a location-based approaches to most-popular algorithms optimized for streams to ensemble recommender for streams. Analyzing the results we found, that the provided baseline is hard to beat. Further, CTR varied with respect to the publisher indicating additional factors that affect performance. We observed higher CTR levels compared to last year's edition. This indicates that teams continue to optimize their algorithms.

The technical challenges have been addressed by means of applying optimized data structures supporting the simultaneous access by concurrently running threads. One team focused on machines with multiple cores; another team implemented an approach enabling the distribution over different machines (using the Akka framework).

Finally, we detected issues with the challenge and derived ways to further improve participants' experience. Users struggled to get started. We had provided tutorials for both tasks but participants appeared to require additional support. The Idomaar framework had been updated during the competition. On the one hand, this was necessary to fix technical issues. On the other hand, this required participants to adjust and monitor their systems to a larger degree. Besides improving participants' support, we seek to increase the interchange between both tasks. Participants who evaluate their news recommender with ORP should take advantage of the recorded data to better tune their algorithms. Conversely, participants working with the recorded data should check their algorithms' performance with ORP. Thereby, they assure that their algorithms not only scale well but provide relevant suggestions. Said et al. [9] strongly advocate such multi-objective evaluation.

Acknowledgement

The work leading to these results has received funding (or partial funding) from the Central Innovation Programme for SMEs of the German Federal Ministry for Economic Affairs and Energy, as well as from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement number 610594.

References

1. T. Brodt and F. Hopfgartner. Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform. In *Proceedings of the Information Interaction in Context conference*, IIiX'14, pages 223–226. Springer-Verlag, 2014.
2. G. Gebremeskel and A. P. de Vries. The degree of randomness in a live recommender systems evaluation. In *Working Notes for CLEF 2015 Conference, Toulouse, France*. CEUR, 2015.
3. F. Hopfgartner, A. Hanbury, H. Mueller, N. Kando, S. Mercer, J. Kalpathy-Cramer, M. Potthast, T. Gollup, A. Krithara, J. Lin, K. Balog, and I. Eggel. Report of the evaluation-as-a-service (EaaS) expert workshop. *SIGIR Forum*, 49(1):57–65, 2015.

4. F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, and T. Heintz. Benchmarking news recommendations in a living lab. In *5th International Conference of the CLEF Initiative*, pages 250–267, 2014.
5. B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz. The plista dataset. In *NRS'13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, pages 14–21. ACM, 10 2013.
6. B. Kille, A. Lommatzsch, R. Turrin, A. Sereny, M. Larson, T. Brodt, J. Seiler, and F. Hopfgartner. Stream-based recommendations: Online and offline evaluation as a service. In *Proceedings of the 6th International Conference of the CLEF Association*, CLEF'15, 2015.
7. 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.
8. A. Lommatzsch and S. Werner. Optimizing and evaluating stream-based news recommendation algorithms. In *Proceedings of the Sixth International Conference of the CLEF Association, CLEF'15*, LNCS, vol. 9283, Heidelberg, Germany, 2015. Springer.
9. A. Said, D. Tikk, K. Stumpf, Y. Shi, M. Larson, and P. Cremonesi. Recommender systems evaluation: A 3D benchmark. In *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012)*, RUE'12, pages 21–23. CEUR-WS Vol. 910, 2012.
10. I. Verbitskiy, P. Probst, and A. Lommatzsch. Developing and evaluation of a highly scalable news recommender system. In *Working Notes for CLEF 2015 Conference, Toulouse, France*. CEUR, 2015.