Overview of CLEF NEWSREEL 2014: News Recommendation Evaluation Labs

Benjamin Kille[1], Torben Brodt[2], Tobias Heintz[2], Frank Hopfgartner[1], Andreas Lommatzsch[1], and Jonas Seiler[2]

[1] DAI-Labor, Technische Universität Berlin, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
{hopfgartner,kille,lommatzsch}@dai-labor.de
[2] plista GmbH
Torstr. 33-35, D-10119 Berlin, Germany
{tb,thz,jse}@plista.com

Abstract. This paper summarises objectives, organisation, and results of the first news recommendation evaluation lab (NEWSREEL 2014). NEWSREEL targeted the evaluation of news recommendation algorithms in the form of a campaign-style evaluation lab. Participants had the chance to apply two types of evaluation schemes. On the one hand, participants could apply their algorithms onto a data set. We refer to this setting as *off-line evaluation*. On the other hand, participants could deploy their algorithms on a server to interactively receive recommendation requests. We refer to this setting as *on-line evaluation*. This setting ought to reveal the actual performance of recommendation methods. The competition strived to illustrate differences between evaluation with historical data and actual users. The on-line evaluation does reflect all requirements which active recommender systems face in practise. These requirements include real-time responses and large-scale data volumes. We present the competition's results and discuss commonalities regarding participants' approaches.

Keywords: recommender systems, news, on-line evaluation, living lab

1 Introduction

The spectrum of available news continuously grows as news publishers keep producing news items. At the same time, we observe publishers shifting from pre-dominantely print media towards on-line news outlets. These on-line news portals confront users with the choice between numerous news items inducing an information overload. Readers struggle to detect relevant news items in the continuous flow of information. Therefore, operators of news portals have established systems to support them [2]. The support includes personalisation, navigation, context-awareness, and news aggregation.

CLEF NEWSREEL focuses on support through (personalised) content selection in form of news recommendations. We assume that users benefit as news portals adapt to current trends, news' relevancy, and individual tastes. News recommendation partially includes enhanced navigation as well as context-awareness. Recommended news items serves as a mean to quickly navigate to relevant contents. Thus, users avoid returning to the home page to continue consuming news. In addition, news recommender systems

may take advantage of contextual factors. These factors include time, locality, along with trends.

Within *NEWSREEL* participants ought to find recommendation algorithms suggesting news items for a variety of news portals. These news portals cover several domains including general news, sports, and information technology. All news portals provide pre-dominantely German news articles. Consequently, approximately 4 out of 5 visitors' browsers carry location identifiers pointing to Germany, Austria, or Switzerland, respectively. The goal of the lab was to let participants determine which of these factors play an important role when recommending news items. The remainder of this paper is organised as follows. Section 2 describes the two tasks and their evaluation methodology. Section 3 summarises the results of the lab and discusses difficulties reported by participants. Section 4 concludes the paper and gives an outlook on how we attempt to continue evaluating news recommendation algorithms.

2 Lab Setup

CLEF NEWSREEL consisted of two tasks. For Task 1 we provided a data set containing recorded interactions with news portals. We refer to Task 1 as *off-line evaluation*. In addition, participants could deploy their recommendation algorithms in a living lab for Task 2. We refer to Task 2 as *on-line evaluation* and to the living lab platform as the *Open Recommendation Platform* (ORP)[3]. ORP is operated by plista[4], a company that provides content distribution as well as targeted advertising services for a variety of websites. We dedicate a section to each task describing the goal and evaluation methodology. The reader is refered to [10] for a detailed overview of the setup.

2.1 Task 1: Off-line Evaluation

Task 1 mirrors the paradigm of formerly held recommendation challenges such as the *Netflix Prize* challenge (cf. [1]). As part of the challenge, Netflix released a collection of movie ratings. Participants had to predict ratings for unknown (user, item)-pairs in a hold-out evaluation set. Analoguously, we split a collections of interaction with news items in training and test partitions. The initial data set has been described in [11]. Netflix could split their data randomly. This is due to the underlying assumption that movie preferences remain constant over time. In other words, users will continue to (dis-)like movies they once (dis-)liked. In contrast, we refrain to assume that users will enjoy reading news articles they once read. Conversely, we suppose that news' relevancy decreases relatively quickly. Thus, we relinquished to randomly select interactions for evaluation. Instead, we randomly selected time frames which we completely removed from the data set. We avoided a moving time-window approach, as this would have meant to release the entire data collection. We considered 3 parameters for the randomised sampling:

- Portal specificity

[3] `http://orp.plista.com`
[4] `http://plista.com`

– Interval width
– Interval frequency

Portal specificity refers to the choice between using identical time intervals for all news portals and having portal-specific intervals. The former alternative lets us treat all portals in the same way. On the other hand, the latter alternative provides a setting where participants may utilise information from other sources – i.e., other news portals – which better reflects the situation actual news recommenders reflect. For instance, articles targeting a certain event may have been published on some news portals already. The complementary portals could use interactions with these articles to boost their own articles. We decided to sample portal-specific time slots for evaluation. Selecting a suited interval width represents a non-trivial task. Choosing the width too small will result in an insufficient amount of evaluation data. Conversely, setting the width too large will entail a rather high number of articles as well as users missing in the training data. Additionally, the amount of interactions varies over the day and the week. For instance, we observe considerably fewer interaction in the night than in day times. We decided not to keep the interval width fixed, since we expected that this would remedy coincidental bad choices. Thus, we varied the interval width in the set $\{30, 60, 120, 180, 240\}$ minutes. We observed that recommendation algorithms will struggle to provide adequate suggestion based on training data that lacks the most recent 4 hours [15]. This is mainly due to the rapidly evolving character of news. Moreover, we observe that news portals continue to provide new items which attract a majority of readers. The intial raw data covers a time span of about 1 month. We faced the decision on how many time slots to remove for evaluation. We had to avoid removing data as extensively as leaving insufficient data for training. Conversely, we strived to obtain expressive results. We decided to sample approximately 15 time slots per news portal. Thus, we expected to extract evaluation data about every second day. We noticed some time slots overlapped by chance. We decided to merge both time slots and refrained from resampling. Algorithm 1 outlines the sampling procedure.

Algorithm 1 Sampling Procedure

Input: set of p news portals P, set of w interval widths W, number of samples s

```
T ← ∅                                              ▷ set to contain the sample result
function SAMPLE(P, W, s)
    for i ← 1 to p do
        for j ← 1 to s do
            T ← T ∪ random(t, w)        ▷ randomly choose a time point t and interval width w
        end for
    end for
    return T
end function
```

Having created data for training and testing, we yet have to determine an evaluation metric. Literature on recommender systems' evaluation provides a rich set of metrics.

Metrics relating to rating prediction accuracy and item ranking are among the most popular choices. Hereby, root mean squared error (RMSE) and mean absolute error (MAE) are frequently used for the former evaluation setting (cf. [8, 9]). Supporters of ranking-oriented evaluation favour metrics such as precision/recall [5], normalised discounted cumulative gain [16], or mean reciprocal rank [14]. Rating prediction as well as ranking-based evaluation require preferences with graded relevancy as input. However, users do not tend to rate news articles. Thus, we cannot apply rating prediction metrics. Also, we cannot apply ranking metrics as we lack data about the pair-wise preference towards news items. Our data carry the signal of users interacting with news items. Thus, we ended up to define the evaluation metric based upon the ability to correctly predict whether an interaction will occur.

Let the pair (u, i) denote user u reading news item i included in the evaluation data. We challenged participants to select the 10 items each previously observed user would interact with in each evaluation time slot. The choice of exactly 10 items to suggest may appear arbitrary. We observe a majority of users interacting with only few items. Thus, most of the suggestions are likely not correct. On the other hand, limiting the number of suggestions to very few items entails drawbacks as well. Imagine a user who actually reads five articles in a time slot contained in the evaluation data. Having participants suggesting 3 items, a recommender predicting all 5 interactions correctly will appear to perform on level with a recommender only predicting 3 interactions correctly. Thus, requesting many suggestions will provide more sensitivity. This sensitivity allows us to better differentiate the individual recommendation algorithms' performances. On the other hand, the included news portals do not provide more than 6 recommendations at a time. Hence, requesting substantially more recommendation will induce a setting which insufficiently reflects the actual use case. Thus, we opted for 10 suggestions which represents a reasonable trade-off between sensitivity and reflecting the actual scenario. Note that [6] found that 10 preferences typically suffice to provide adequate recommendation. Finally, we define the evaluation metric according to Equation 1:

$$h = \frac{\sum_{u \in U} \sum_{j=1}^{10} \mathbb{I}(u, i_j)}{10|U|} \tag{1}$$

where h refers to the *hitrate*. $\mathbb{I}$ represents the indicator function returning 1 if the predicted interaction occurred and 0 otherwise. The denominator normalises the number of hits by the maximal number possible. Thus, the hitrate falls into the interval $[0, 1]$. Since most users will not exhibit 10 interactions in the evaluation time slot, we expect the hitrate to be closer to 0.

2.2 Task 2: On-line Evaluation

Task 2 follows an alternative paradigm compared to Task 1. Task 1 assures comparability of results. This is mainly due to the fact that all participants apply their algorithm onto identical data. Contrarily, Task 2 provides a setting where participants have to handle similar yet not identical data. The plista GmbH has established a living lab where researchers and practitioners can deploy their recommendation algorithms to interact with actual users. This approach allows us to observe the actual performance of

recommendation methods. This means that our findings will reflect actual benefits for real users. Further, we are able to observe variations throughout time and conduct studies on large scale as we record more and more data. Conversely, evaluation on recorded data expresses how a method *would* have performed. The approach does entail some disadvantages as well. Participants had to deal with technical requirements including response times, scalability, and availability. Deployed systems faced numerous requests which they had to reply to in at most 100ms. This response time restriction represents a particular challenge for participants located far from Germany where the ORP servers are located. Network latencies might further reduce the available response time. We offered virtual machines to participants who either had no servers at their disposal or suffered from high network latency. As a result, these requirements allowed us to verify how well certain recommendation algorithms adapt to real-world settings.

We asked participants to deploy their recommendation algorithm to a server. Subsequently, they connected the server to ORP which forwarded recommendation requests. Widgets on the individual news portals' website displayed the suggested news items to users. ORP tracks success in terms of clicks. This opens up several ways to evaluate participants' performances. One option is to consider the number of clicks. Considering the relative number of clicks by requests represents another option. Industry refers to this metric as click-through-rate. Given a comparable number of requests, both quantities coincide. In situations with varying number of requests, evaluation becomes tricky. Considering the total number of clicks may bias the evaluation in favour of the participant with more clicks. Conversely, considering the relative number of clicks per requests may favour teams with few requests. We want to evaluate the performance of a recommendation algorithm. ORP provides all participants with the chance to obtain similar number of requests. We decided to consider the absolute number of clicks as decisive criteria. Nevertheless, we additionally present the relative number of clicks per requests in our evaluation.

Baselines support comparing the relative performance of algorithms. We deployed a baseline which is detailed in [13]. The baseline combines two important factors for news recommendation: popularity and recency. We consider a fixed number of interactions that most recently occurred. Our baseline recommends news items included in this list that users had not previously seen. Consequently, we obtain a computationally efficient method that inherently considers popularity and recency.

We realised the participants' need to tune their algorithms. For this reason, we explicitly defined 3 evaluation periods during which performances would be logged. Participants could improve their algorithms before as well as in between the periods. We set the 3 periods to 7-23 February, 1-14 April, and 27-31 May.

3 Evaluation

In this section, we detail results of *CLEF NEWSREEL* 2014. We start by giving some statistics about the participation in general. Then, we discuss the results for both tasks. Note that we unfortunately did not receive any submissions for Task 1. We provide some considerations about reasons for this.

3.1 Participation

51 participants registered for Task 1. 52 participants registered for Task 2. Thereof, no participant submitted a solution for Task 1. We observed 13 active participants for Task 2. Note that participants had the chance to contribute several solutions for Task 2. 4 participants submitted a working notes paper to the CLEF proceedings [3].

3.2 Evaluation of Task 1

We have not received any submissions for Task 1. Thus, we cannot report any results on how well the future interactions could be predicted. We can think of several reasons which may have prevented participants from submitting results. First, the data set exhibits a large volume of more than 60GB. Thus, we required participants to process such volumes. Participants' available computational resources may not have allow to iteratively optimise their recommendation algorithms for this amount of data. Second, we imagine that participants might have preferred Task 2 over Task 1. This preferences may be due to the interactive character as well as the rather unique chance to evaluate algorithms with actual users' feedback. We admit that there are rather plenty of data set driven competition. For instance, the online platform `www.kaggle.com` offers a variety of data sets. Finally, the restriction to German news articles might have prevented participants who attempted to evaluate content-based approach but do not speak German.

3.3 Evaluation of Task 2

Throughout the pre-defined evaluation periods, we observed 13 active participants on the ORP. Unfortunately, the component recording the performance failed twice. Thus, we did not receive data for the times between 7-12 February and 27-31 May. None of the teams were active in all periods. This illustrates the technical requirements which participants faced. ORP automatically disables the communication with participants in case their servers do not respond in time. ORP tries to re-establish the communication. We noticed that the re-establishing has not succeeded in all occasions. We allowed participants to simultaneously deploy several algorithms. Some participants used this more extensively than other did. Table 1 shows the results for the evaluation periods 7–23 February, 1–14 April, and 27–31 May. We list the number of clicks, requests and their ratio for each algorithm which was active during the period. We note that the number of requests does vary between algorithms. Algorithm *AL* gathered the most clicks in periods 2 and 3 as well as the second most clicks in the first period. Note that the *baseline* constantly appears under the five best performing algorithms. This indicates that popularity and

recency represent two important factors when recommending news. Additionally, the baseline provides low computational complexity such that it is able to reply to a large fraction of requests. Table 2 aggregates the results per participant. The aggregated results confirm our impressions from the algorithm-level.

In addition to the overall figure, we investigate whether particular algorithms perform exceptionally well in specific contexts. Context refers either to specific news portals or daytimes. News portals offer varying contents. For instance, `www.sport1.de` is dedicated to sports-related news while `www.gulli.com` provides news on information technology. Thus, we look at the performance of individual algorithm with respect to specific publishers. Likewise, we investigate algorithms' performances throughout the day. We suppose that different types of users consume news at varying hours of the day. For instance, users reading news early in the morning may have other interests than users reading late in the evening. This matches with the findings of [18]. Figure 1 shows a heatmap relating algorithms with the publisher and hour of day. We note that few algorithms perform on comparable levels for all publishers and throughout the day. This indicates that combining several recommendation algorithms in an ensemble yields potential to obtain better performance.

We do not know details to all recommendation algorithms. The participants who submitted their ideas in form of working notes used different ideas. Most systems carried a fall-back solution in terms of most-popular and/or most-recent strategies. Additionally, participants contributed more sophisticated algorithms. These algorithms included association rules [12], content-based recommenders [4], and ensembles of different recommendation strategies [7]. Reportedly, more sophisticated methods had trouble dealing with the high volume of requests. In particular the peaking hours during lunch break were reportedly hard to handle. Our baseline method combines the notions of most-popular and most-recent recommendation. The evaluation shows that the baseline is hard to beat. This may be due to the technical restrictions rather than the recommendation quality. More sophisticated method which just miss the response time limit may provide better recommendations.

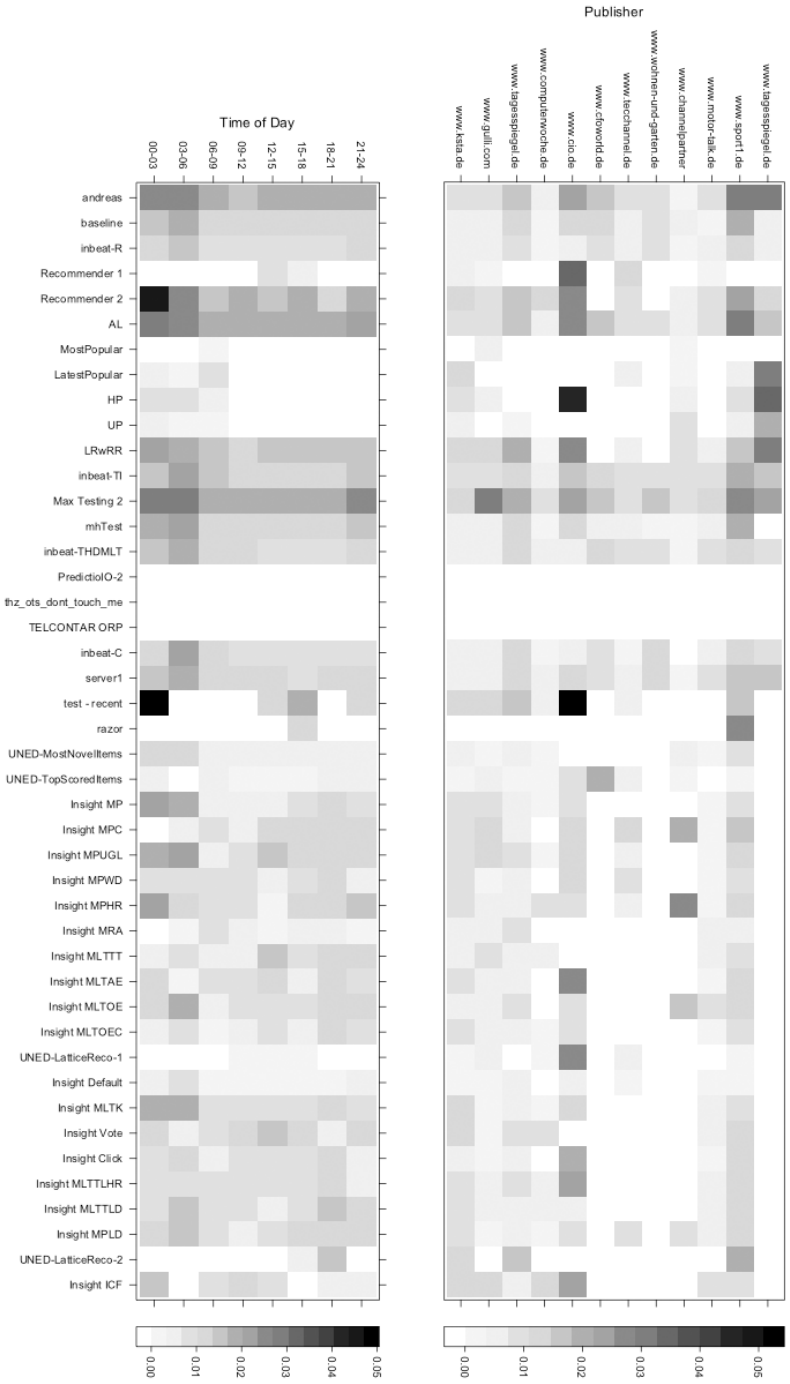

Fig. 1: Algorithms' click-through-rate grouped by time of day as well as by publisher.

Table 1: Results of Task 2 grouped by the evaluation periods. We list the number of clicks, number of requests, and their ratio for all participating algorithms. Notice that the highest numbers of clicks per period are highlighted in bold font.

	7–23 Feb			1–14 Apr			27–31 May		
Algorithm	**Clicks**	**Requests**	**CTR**	**Clicks**	**Requests**	**CTR**	**Clicks**	**Requests**	**CTR**
AL	6,426	436,094	0.01	**17,220**	801,078	0.02	**2,519**	127,928	0.02
andreas	**8,649**	581,243	0.01	16,004	767,039	0.02	422	30,519	0.01
baseline	2,642	256,406	0.01	5,616	418,556	0.01	663	54,912	0.01
HP	-	-	-	56	8,501	0.01	-	-	-
inbeat-C	1,855	193,114	0.01	120	13,987	0.01	-	-	-
inbeat-R	2,118	228,930	0.01	141	16,747	0.01	1,192	106,303	0.01
inbeat-THDMLT	1,883	187,466	0.01	97	14,373	0.01	3	618	0.00
inbeat-TI	3,139	251,529	0.01	1,222	88,540	0.01	-	-	-
Insight Click	-	-	-	-	-	-	153	17,751	0.01
Insight Default	-	-	-	-	-	-	122	42,903	0.00
Insight ICF	-	-	-	-	-	-	29	2,949	0.01
Insight MLTAE	-	-	-	-	-	-	128	14,833	0.01
Insight MLTK	-	-	-	-	-	-	177	21,620	0.01
Insight MLTOE	-	-	-	-	-	-	107	11,711	0.01
Insight MLTOEC	-	-	-	-	-	-	86	11,526	0.01
Insight MLTTLD	-	-	-	-	-	-	170	19,380	0.01
Insight MLTTLHR	-	-	-	-	-	-	181	20,185	0.01
Insight MLTTT	-	-	-	-	-	-	96	12,455	0.01
Insight MP	-	-	-	-	-	-	126	16,481	0.01
Insight MPC	-	-	-	-	-	-	119	11,933	0.01
Insight MPHR	-	-	-	-	-	-	135	13,754	0.01
Insight MPLD	-	-	-	-	-	-	171	19,394	0.01
Insight MPUGL	-	-	-	-	-	-	115	11,425	0.01
Insight MPWD	-	-	-	-	-	-	130	16,553	0.01
Insight MRA	-	-	-	-	-	-	78	14,297	0.01
Insight Vote	-	-	-	-	-	-	179	17,632	0.01
LatestPopular	-	-	-	52	8,736	0.01	-	-	-
LRwRR	-	-	-	1,333	89,428	0.01	-	-	-
Max Testing 2	1,932	106,151	0.02	6,972	337,551	0.02	151	10,192	0.01
mhTest	521	119,067	0.00	3,914	220,303	0.02	1,102	100,604	0.01
MostPopular	-	-	-	2	8,652	0.00	-	-	-
razor	0	57	0.00	1	81	0.01	-	-	-
Recommender 1	-	-	-	18	2,816	0.01	-	-	-
Recommender 2	-	-	-	606	35,758	0.02	-	-	-
server1	3,592	335,861	0.01	-	-	-	-	-	-
TELCONTAR ORP	0	370,616	0.00	0	874,759	0.00	-	-	-
test - recent	-	-	-	122	8,231	0.01	-	-	-
UNED-Lattice-Reco-1	-	-	-	-	-	-	61	29,543	0.00
UNED-Lattice-Reco-2	-	-	-	-	-	-	34	2,384	0.01
UNED-MostNovelItems	-	-	-	-	-	-	370	69,607	0.01
UNED-TopScoredItems	-	-	-	-	-	-	142	51,489	0.00
UP	-	-	-	41	8,687	0.00	-	-	-

Table 2: Aggregated results for the participants by clicks, requests, and their ratio. Column 2 refers to publications detailing the algorithms if available.

Team	Reference	Clicks	Requests	CTR
labor	[13]	26,165	1,365,100	0.02
abc	[13]	25,075	1,378,801	0.02
inbeat	[12]	11,770	1,101,607	0.01
plista GmbH		9,055	1,765,379	0.01
baseline	[13]	8,921	729,874	0.01
ba2014	[17]	5,537	439,974	0.01
student		3,592	335,861	0.01
insight	[7]	2,425	305,094	0.01
recommenders.net		1,333	89,428	0.01
artificial intelligence		624	38,574	0.02
uned	[4]	607	153,023	0.00
i2r		151	34,576	0.00
TELCONTAR		0	1,245,375	0.00

4 Conclusion

CLEF NEWSREEL attempted to let participants evaluate their recommendation algorithms. Participants could evaluate their algorithm in two varying fashions. Task 1 offered a rich data set recorded through a one month period on 12 news portals. We removed time slots for evaluation purposes. Participants ought to predict which articles users would read during these held-out times. We received no contribution for this task. Task 2 enabled participants to evaluate their recommendation algorithms by interacting with actual users. Participants could deploy their algorithms on a server which subsequently received recommendation requests. This setting closely mirrors circumstances under which actual recommender systems operate. Participants struggled with the high volume of requests and the narrow response time limits. We observed that most-popular and most-recent approaches are hard to beat due to their low complexity.

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 Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 610594.

References

1. J. Bennett and S. Lanning. The netflix prize. In *KDDCup*, pages 3–6, 2007.
2. D. Billsus and M. J. Pazzani. Adaptive News Access. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, chapter 18, pages 550–570. Springer, 2007.
3. L. Cappellato, N. Ferro, M. Halvey, and W. Kraajl. Clef 2014 labs and workshops, notebook papers. In *CLEF 2014 Labs and Workshops, Notebook Papers*. CEUR Workshop Proceedings, 2014.
4. A. Castellanos, A. Garcia-Serrano, and J. Cigarran. Uned @ clef-newsreel 2014. In *CLEF 2014 Labs and Workshops, Notebook Papers*, 2014.
5. P. Cremonesi. Performance of recommender algorithms on top-n recommendation tasks categories and subject descriptors. In *Proceedings of the 2010 ACM Conference on Recommender Systems*, pages 39–46, 2010.
6. P. Cremonesi, P. Milano, and R. Turrin. User effort vs . accuracy in rating-based elicitation. In *6th ACM Conferene on Recommender Systems*, pages 27–34, 2012.
7. D. Doychev, A. Lawor, and R. Rafter. An analysis of recommender algorithms for online news. In *CLEF 2014 Labs and Workshops, Notebook Papers*, 2014.
8. A. Gunawardana. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.
9. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. (TOIS)*, 22(1):5–53, 2004.
10. F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, and T. Heintz. Benchmarking news recommendations in a living lab. In *CLEF'14: Proceedings of the Fifth International Conference of the CLEF Initiative*. Springer Verlag, 2014. to appear.

11. B. Kille, F. Hopfgartner, T. Brodt, and T. Heinzt. The plista dataset. In *Proceedings of the International News Recommender Systems Workshop and Challenge*, 2013.
12. J. Kuchar and T. Kliegr. Inbeat: Recommender system as a service. In *CLEF 2014 Labs and Workshops, Notebook Papers*, 2014.
13. A. Lommatzsch. Real-time news recommendation using context-aware ensembles. In *Advances in Information Retrieval*, pages 51–62. Springer, 2014.
14. Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf : Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys*, pages 139–146, 2012.
15. 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 *Proceedings of the 7th ACM conference on Recommender systems*, pages 481–482, 2013.
16. S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11*, page 109, 2011.
17. S. Werner and A. Lommatzsch. Optimizing and evaluating stream-based news recommendation algorithms. In *CLEF 2014 Labs and Workshops, Notebook Papers*, 2014.
18. J. Yuan, S. Marx, F. Sivrikaya, and F. Hopfgartner. When to recommend what? a study on the role of contextual factors in ip-based tv services. In *MindTheGap'14: Proceedings of the MindTheGap'14 Workshop*, pages 12–18. CEUR, 2014.