Defining a Meaningful Baseline for News Recommender Systems

Benjamin Kille
Institute of Technology Berlin
Berlin, Germany
benjamin.kille@tu-berlin.de

Andreas Lommatzsch
Institute of Technology Berlin
Berlin, Germany
andreas.lommatzsch@dai-labor.de

ABSTRACT

Evaluation protocols for news recommender systems typically involve comparing the performance of methods to a baseline. The difference in performance ought to tell us what benefit we can expect from using a more sophisticated method. Ultimately, there is a trade-off between performance and effort in implementing and maintaining a system. This work explores what baselines have been used, what criteria baselines must fulfil, and evaluates a variety of baselines in a news recommender evaluation setting with multiple publishers. We find that circular buffers and trend-based predictions score highly, need little effort to implement, and require no additional data. Besides, we observe variations among publishers, suggesting that not all baselines are equally competitive in different circumstances.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

news recommender systems, evaluation, baselines

1 INTRODUCTION

Readers struggle to keep up with the plethora of stories which publishers continue to release on their digital platforms. News Recommender Systems (NRS) support readers navigating the dynamic news landscape. They deliver a subset of articles deemed interesting. Publishers' success depends—at least partially—on how much of readers' attention they obtain. Revenue strongly correlates with the number of advertisements shown to readers in an "attention economy" [2]. Consequently, publishers want to know whether an NRS recommends relevant articles to their readers. The dynamic character of the news landscape impedes on the comparability of evaluation results. For instance, breaking news may shift readers' attention in a way completely unrelated to the recommendation algorithms. Publishers account for this effect by comparing recommendation algorithms to baselines [30].

The choice of baseline introduces a variable into the evaluation protocol. Ideally, evaluators would use the same baseline to arrive at comparable results. Not every baseline applies to each recommendation task. For instance, collaborative filtering requires user profiles [15], whereas content-based filtering needs meta-data about items [24]. Section 2 reviews previous research on NRS focusing on which baselines they have used. Section 3 derives requirements which baselines must fulfil. Section 4 describes experiments conducted on a large-scale data set from three publishers. The experiments compare the performance of a variety of baselines. Section 5 summarises our findings and hints at directions for future research.

2 RELATED WORK

A consensus concerning the evaluation protocol of NRS has yet to establish. The availability of data affects what baselines evaluators can use in their experiments. Frequently, researchers use recorded interactions between users and news articles. Whenever researchers have access to the NRS directly, they can even employ counterfactual reasoning [31]. The earliest works on automated NRS relied on items' popularity as a baseline. The rationale behind the popularity baseline suggests that items relevant to many users are suited candidates. Das et al. [7] and Garcin et al. [9] employ the popularity baseline. Lommatzsch [22] uses a circular buffer implementation as baseline. This implementation combines the popularity of items with a recency focus. Researchers interested in content-based news recommendation devise baselines using content features. Gao et al. [8] and Zheng et al. [37] use a term-frequency model as a baseline. Cantador et al. [3] use a keyword-based baseline for their semantic news recommendation model. Okura et al. [27] define a word-based baseline for their embedding experiments. Li et al. [17] use an ϵ-greedy strategy as baseline in their contextual bandit evaluation. Li et al. [20] and Lu et al. [25] use collaborative filtering and content-based filtering baselines. Some researchers invest considerable resources to replicate existing results by re-implementing proposed news recommendation algorithms. Li and Li [18] compare their results to [5, 7, 17, 19, 21]. Zheng et al. [36] contrast their findings to [4, 17, 28, 32]. Wang et al. [33] consider [4, 6, 10, 13, 28, 34, 35]. Khattar et al. [14] compare their approach with [11, 12, 16, 26, 29]. Studying the baselines, we notice variances across works. Some baselines appear frequently. Some papers describe baselines tailored to their particular use case. There is no consensus on what baseline should be used. Moreover, it remains unclear how baselines correlate or interdepend on one another.

3 BASELINE REQUIREMENTS

Baselines allow us to measure the relative improvement of an algorithm. Evaluators can not only report a value but express how much better the value is compared to a more straightforward method. Thereby, we learn whether it is worth investing additional effort into developing more sophisticated algorithms. Still, defining a meaningful baselines poses a technical challenge. Defining a baseline which fails miserably does not reveal much about relative improvements. Defining a baseline which requires too much effort or has too many design choices might render evaluation results hard to compare. Requiring additional data can make it impossible for some baselines to be used in experiments where these data are lacking. Based on these propositions, we introduce three requirements for news recommendation baselines:

(1) low implementation effort
(2) competitive performance
(3) no additional data required

The next section introduces a variety of candidate algorithms and analyses how well they perform in a large-scale experiment.

4 EXPERIMENT

In this experiment, we use the NewsREEL evaluation platform described in Lommatzsch et al. [23]. NewsREEL offers researchers the opportunity to evaluate their news recommendation algorithms on real users with several connected publishers. We use data recorded in the time 1 March 2017 to 30 June 2018 (14 months) including approximately 94 million sessions of three publishers. The system tracks readers using session cookies. The session information allows us to link reading events to a particular reader. Whenever more than 1 h passes in between events, we create a new session. Note that we disregard all sessions with a single reading event as we cannot compare predictions to future events in these cases. All three publishers operate an NRS. Empirically, the NRS produce clicks on the order of a few per thousand recommendations. Hence, their effect on the collected reading events appears negligible.

Table 1: Statistics describing the interactions of readers with news articles for three publishers. The symbols refer to the number of sessions (S), the number of interactions (N), the average number of events per sessions (EpS), and the number (S^1) and proportion ($S^1_\%$) of sessions with only one event.

Statistic	Publisher A	Publisher B	Publisher C
S	17 019 523	22 683 047	54 272 242
N	36 859 823	175 930 128	105 998 109
EpS	2.17	7.76	1.95
S^1	10 529 390	416 506	34 998 380
$S^1_\%$	61.19 %	1.84 %	64.49 %

Table 1 outlines the characteristics of the data set. Publishers A and C have to deal with expectedly short sessions. Table 1 also lists the number and proportion of sessions with only a single interaction. This matters as we have to disregard those sessions in the evaluation. Without a second interaction, we cannot determine whether a recommendation was successful.

4.1 Candidate Baselines

In the experiment, we consider eight baseline candidates.

Random. The random method considers all known articles and picks a suggestion at random. In addition, we consider a slightly advanced version which draws only from the set of items published in a certain time window. This accounts for the readers' desire to read more current news.

Popularity. The popularity method suggests exactly the article which has been read most often. Similarly to the random method, we consider a version of the popularity version, which considers reading frequencies in a specific time window.

Recency. The recency method recommends news articles most recently published. The method disregards any form of popularity or personalisation.

Reading Sequences. The reading sequences method monitors which article users read in sequence. It recommends exactly the article which users read most frequently given the readers current article.

Collaborative filtering. Applying Collaborative Filtering to news recommendation is particularly challenging. Not only keep publishers adding new items, but also do systems know very little about users. We follow the suggestion of Das et al. [7] and implement a `MinHash` version of Collaborative Filtering.

Content-based Filtering. Content-based filtering requires a way to define similarity among news articles. Generally, we could employ a string matching approach on the title or text. Still, this would require considerable computational effort. Besides, the possibility of different languages adds another level of difficulty. We have implemented a more straightforward method. The Content-based filtering takes the category of news articles as a proxy for similarity and suggests articles from the same category at random.

Circular buffer. We use the circular buffer proposed by Lommatzsch [22] which has also been used as the baseline of CLEF NewsREEL [23]. This method has a fixed size list which the systems updates as interactions occur. The system adds the article of each interaction. When the system arrives at the end of the list, it moves the index to the first position and goes on. The methods select recommendations by reverse lookup in the list. Thereby, it combines popularity and recency. More popular articles occur more frequently on the list. More recently read articles occur with a higher probability as well.

Trending. The trending method computes the trend for each article in a given time window. More specifically, the method carries out a regression on the reading frequency binned on an hourly level. The method recommends those articles with the steepest trend.

Implementing baseline candidates causes little effort. Content-based filtering requires knowing the category of news articles. All remaining candidates only require interaction data with timestamps. Hence, they fulfil already two of our three requirements.

4.2 Evaluation Protocol

We present each event to all of the baseline candidates and request exactly one item as a recommendation. The evaluator stores the recommendations with reference to the session. When the session re-appears, the evaluator checks whether this article had been recommended previously. If that is the case, the evaluator adds one to the score (x) of that baseline.

Based on the recorded score, we compute two evaluation measures on recommendation success on event and session level:

$$R_x = |S|^{-1} \sum_{x \in S} x \tag{1}$$

$$R_s = |S|^{-1} \sum_{x \in S} 1_{\{x(s)>0\}} \tag{2}$$

The first score, R_x approximates the expected number of successful recommendations per session. The second score, R_s, estimates the chance that at least one recommendation will succeed.

4.3 Evaluation Results

Table 2 shows our observations for all combinations of publishers and baselines. We notice that the baselines' performances vary considerably. We have scaled the results by a factor of 10^{-5} to obtain more legible figures. Thus, a score of 1000 refers to one per cent. The random baseline performs poorly among all publishers. The chance to randomly suggest something interesting is below 1 in 1000. Constraining the time window to shorter periods slightly improves the success rate. The performance of popularity, recency, and sequences differs between publishers. Publisher A and C show better results for popularity and sequences. Sequences even perform best for Publisher A overall. Publisher B, on the other hand, shows the best results for recency, and, conversely, the worst performance for sequences. This is surprising as Publisher B sees the longest sessions on average (cf. Table 1). The presence of very specific sequences could spoil the predictions, particularly for recently published items. Content-based filtering performs poorly, especially for Publisher C. Publisher C focusses on the narrow topic of automotive news. This could lead to circumstances in which the NRS faces a large number of very specific categories. Collaborative filtering performs well for publishers A and B but not for C. The circular buffer and the trending baseline perform well for all publishers. The circular buffers exhibit little variation depending on the parameter choice. This could be due to the highly skewed frequencies with which readers engage with articles. The distributions exhibit a strong popularity bias which supplies the same small subset of articles to the buffer. As a result, the recommendations would largely coincide among lists of varying lengths. Publishers A and B show improvements for shorter time windows, whereas Publisher C trending baseline performance peaks at 12 hours. Besides, we observe different maximum results among the three publishers. Sequences top Publisher A at 18.5 % (R_x) and 14.7 % (R_s). In contrast, the circular buffer with one hundred elements scores highest for Publisher B with 11.7 % (R_x) and 3.0 % (R_s). The circular buffer with five hundred or a thousand elements performs best for Publisher C with 2.1 % (R_x) and 1.6 % (R_x). Hence, the differences with respect to R_x span 16.4 %. Such substantial differences are unlikely to emerge from the baseline. Instead, we have to assume that other aspects play a vital role, such as the composition of the readership, interface design, and content.

5 CONCLUSION AND FUTURE WORK

News Recommender Systems support readers navigating the news landscape by suggesting which article to read next. Evaluation is necessary to optimise NRS. To estimate how much value a new method adds, evaluation protocols compare their results to baselines. A consensus on what baselines to use has yet to establish. Researchers have used a variety of baselines (cf. Section 2). We have formulated three criteria which baselines have to meet to be considered a viable option. Section 4.1 presents a list of candidate baselines, all of which require manageable effort to implement and need not much additional data. To check the candidate baselines' competitiveness, we have devised an experiment on three publishers with data covering 14 months. The results suggest that the circular buffer and the trending baseline stably provide competitive performance for all publishers. We have observed variations among the performance of baselines for particular publishers. For instance, the sequence baseline has scored exceptionally well for publisher A yet failed for publisher B completely. More research is needed to explore how to transfer our findings to other publishers. We conjecture that similar publishers will confirm the order of performance for the baselines.

Researchers are keen to compare their methods to the most competitive approach. This requires considerable investment in re-implementing previous research. We argue that using the circular buffer or trend-based method already represents a solid baseline. Both have scored higher than collaborative filtering and content-based filtering in our experiments. Researchers ought to highlight the trade-off between predictive accuracy and computational costs. Amatriain and Basilico [1] have highlighted this critical trade-off in the case of streaming service *Netflix*. Their team decided not to implement an ensemble of 107 algorithms as the engineering costs surpassed the added value in prediction accuracy. The accelerated dynamics of collections of news articles raise even

Table 2: Evaluation results. The table shows the two performance scores for each combination of publisher and baseline. The best results for each publishers are highlighted in bold font.

	Publisher A		**Publisher B**		**Publisher C**	
Baseline	$R_x \cdot 10^{-5}$	$R_s \cdot 10^{-5}$	$R_x \cdot 10^{-5}$	$R_s \cdot 10^{-5}$	$R_x \cdot 10^{-5}$	$R_s \cdot 10^{-5}$
random	6.26	6.21	6.16	6.11	1.73	1.69
random (6h)	87.69	85.59	64.30	62.43	13.72	13.45
random (12h)	56.02	55.05	43.05	42.22	9.99	9.81
random (24h)	37.75	37.29	32.69	32.21	8.06	7.88
random (48h)	27.06	26.87	23.80	23.47	6.11	6.04
popular	934.30	653.56	431.17	82.23	2025.96	1581.96
popular (6h)	1009.02	722.28	282.33	53.83	2027.88	1582.75
popular (12h)	1039.16	755.82	526.83	100.99	2031.92	1585.04
popular (24h)	1089.32	774.56	1166.36	216.55	2037.30	1587.64
popular (48h)	1140.12	781.17	1388.02	258.87	2043.20	1590.65
recency	708.23	647.72	1017.66	200.90	38.17	34.36
circular buffer (100)	8069.65	6583.47	**11 747.88**	**3049.56**	2069.87	1583.18
circular buffer (200)	8069.65	6583.47	11 748.00	3049.56	2074.45	1585.22
circular buffer (500)	8069.65	6583.47	11 748.00	3049.56	**2074.76**	**1585.35**
circular buffer (1000)	8067.65	6583.47	11 748.00	3049.56	**2074.76**	**1585.35**
sequences	**18 542.49**	**14 650.12**	0.14	0.03	1523.25	1256.07
content-based	330.10	310.96	286.01	89.93	4.27	4.22
collaborative filtering	5864.39	4861.47	4648.87	845.90	132.68	116.79
trends (2h)	8599.78	6833.47	11 237.24	2066.37	1608.16	1246.95
trends (6h)	6437.37	4743.85	6398.07	1173.67	1784.49	1381.72
trends (12h)	5805.29	4205.00	4481.68	823.05	1795.34	1392.33
trends (24h)	4858.50	3418.07	1580.30	292.20	1723.92	1344.13

stricter constraints on recommender systems than in the case of movies.

We see several directions for future research. First, one could introduce a method to estimate the implementation costs in a more quantitative fashion. This would allow us to address the trade-off more rigorously. Likewise, one could measure the data footprint of different baselines to assess their space and time complexity. Second, one could apply the proposed baselines and possible additions or adaptions on data from other publishers. This would help to quantify the generality of our findings. Finally, evaluation protocols including more than a single recommendation could reveal how ranking metrics compare to our binary scheme. Still, ranking metrics might not display most accurately whether a system performs well unless it presents recommendations in the form of a list.

REFERENCES

[1] Xavier Amatriain and Justin Basilico. 2012. Netflix recommendations: beyond the 5 stars (part 1). *Netflix Tech Blog* 6 (2012).

[2] Erik Brynjolfsson and JooHee Oh. 2012. The Attention Economy - Measuring the Value of Free Digital Services on the Internet. *ICIS* (2012).

[3] Iván Cantador, Pablo Castells, and Alejandro Bellogín. 2011. An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. *International Journal on Semantic Web and Information Systems (IJSWIS)* 7, 1 (2011), 44–78.

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10.

[5] Wei Chu and Seung-Taek Park. 2009. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th international conference on World wide web*. ACM, 691–700.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 191–198.

[7] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 271–280.

[8] Qi Gao, Fabian Abel, Geert-Jan Houben, and Ke Tao. 2011. Interweaving trend and user modeling for personalized news recommendation. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, 100–103.

[9] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 169–176.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 1725–1731. https://doi.org/10.24963/ijcai.2017/239

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 173–182.

[12] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[14] Dhruv Khattar, Vaibhav Kumar, Vasudeva Varma, and Manish Gupta. 2018. Weave&Rec: A Word Embedding based 3-D Convolutional Network for News Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1855–1858.

[15] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. Springer US, Boston, MA, 77–118.

[16] Vaibhav Kumar, Dhruv Khattar, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Neural Architecture for News Recommendation.. In *CLEF (Working Notes)*.

[17] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[18] Lei Li and Tao Li. 2013. News Recommendation via Hypergraph Learning: Encapsulation of User Behavior and News Content. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)*. ACM, New York, NY, USA, 305–314. https://doi.org/10.1145/2433396.2433436

[19] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 125–134.

[20] Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications* 41, 7 (2014), 3168–3177.

[21] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[22] Andreas Lommatzsch. 2014. Real-time news recommendation using context-aware ensembles. In *European Conference on Information Retrieval*, Vol. LNCS 8416. Springer, Springer, 51–62.

[23] Andreas Lommatzsch, Benjamin Kille, Frank Hopfgartner, Martha Larson, Torben Brodt, Jonas Seiler, and Özlem Özgöbek. 2017. CLEF 2017 NewsREEL Overview: A Stream-based Recommender Task for Evaluation and Education. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 239–254.

[24] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*. Springer, Boston, MA, Boston, MA, 73–105.

[25] Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-based collaborative filtering for news topic recommendation. In *Twenty-ninth AAAI conference on artificial intelligence*.

[26] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2016. Learning word embeddings from wikipedia for content-based recommender systems. In *European Conference on Information Retrieval*. Springer, 729–734.

[27] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, New York, NY, USA, 1933–1942. https://doi.org/10.1145/3097983.3098108

[28] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[30] Guy Shani and Asela Gunawardana. 2010. Evaluating Recommendation Systems. In *Recommender Systems Handbook*. Springer US, Boston, MA.

[31] A Swaminathan and T Joachims. 2015. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback.. In *ICML*.

[32] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1633–1642.

[33] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1835–1844. https://doi.org/10.1145/3178876.3186175

[34] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification.. In *IJCAI*. 2915–2921.

[35] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[36] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 167–176. https://doi.org/10.1145/3178876.3185994

[37] Li Zheng, Lei Li, Wenxing Hong, and Tao Li. 2013. PENETRATE: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications* 40, 6 (2013), 2127–2136.