An Empirical Study on News Recommendation in Multiple Domain Settings

Shuichiro Haruta(✉) and Mori Kurokawa

KDDI Research Inc., 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
{sh-haruta,mo-kurokawa}@kddi-research.jp
https://www.kddi-research.jp/english

Abstract. News recommendations using deep neural networks have been a hot research topic. However, most studies on news recommendations are based on the single domain setting. In this paper, we propose a news recommendation framework that uses freezing parameters and fine-tuning techniques for multiple domain settings. Since the model learned by data from multiple news platforms enables the representation of news articles to be much more robust, freezing the parameters of the news encoder effectively works in this setting. Moreover, the characteristics of domain-specific users are captured by fine-tuning the model on each domain data. Our empirical results with a real-world dataset demonstrate that using multiple domain data in the news recommendation results in a better performance. Despite its simplicity, the proposed framework works well, especially for domains where the number of data points is small. This framework has an AUC improvement of about 10% compared with the single domain setting.

Keywords: Recommender system · News recommendation · Deep learning

1 Introduction

Recently, an increasing number of news articles have been provided to us by various news platforms, such as Google news[1] and Gunosy.[2] Since it is impossible for users to read all of them, personalized news recommendation systems have become an important research topic [25]. On behalf of users, personalized news recommendation systems make recommendations by utilizing several pieces of information, such as users and news articles. Personally recommended articles help users save time and improves their user experience.

Although several news recommendation methods have been proposed, most recent research focuses on deep learning techniques [8,9,11,14,19–23] and aims

[1] https://news.google.com.
[2] https://gunosy.co.jp.

T. Hara and H. Yamaguchi (Eds.): MobiQuitous 2021, LNICST 419, pp. 529–540, 2022.
https://doi.org/10.1007/978-3-030-94822-1_34

to achieve higher performance by obtaining distributed representations of both users and articles. For example, in Ref. [14], the authors propose the mechanism that uses denoising autoencoder and triplet loss to represent news articles. The representations of users are based on their browsing history and sessions. NRMS [21] applies word2vec [13] to news titles. The obtained word embeddings are further fed into a multihead self-attention mechanism to capture word relationships. A pretrained language model such as BERT is also used in news recommendations [4,24].

However, to the best of our knowledge, recent research on news recommendations mentioned above deals with the single-domain setting, and there are not enough studies, especially on deep learning-based cross-domain news recommendations. In fact, there is a cross-domain situation where the data of multiple news platforms are available such that a news company deals with multiple news brands (e.g., Japanese news provider "Gunosy" has three brands named "Gunosy", "Newspass", and "Lucra"). In this setting, news articles might be more robustly represented compared with a single-domain setting, and users might show different characteristics in each domain. Therefore, we propose investigating the performance of deep learning-based methods under a cross-domain news recommendation setting.

In this paper, we propose pretraining the model by using all domains' data and freezing news-related model parameters for fine-tuning. It is expected that pretraining with all domains' data enables news embedding to be much more robust, and domain-specific user characteristics are expressed by fine-tuning. As a result of the experiments, the proposed framework worked well, especially in domains where the number of data points was small and increased improvement to 10 % compared with the single domain setting.

The contributions of this paper are as follows:

- To the best of our knowledge, this is the first study in the cross-domain news recommendation field.
- We find that using multiple domain data in the news recommendation brings better performance. Our results are useful for other researchers who would like to know the performance of deep models in this setting.

The remainder of this paper is constructed as follows: We describe the related works on news recommendation and state research questions in Sect. 2. The proposed framework to answer the research questions is described in Sect. 3. In Sect. 4, we evaluate the proposed framework. Finally, we conclude this paper and mention future works in Sect. 5.

2 Related Works

We can roughly classify the news recommendation methods into two approaches, "traditional methods" and "deep learning-based methods". We introduce the representative methods in this section.

2.1 Traditional Methods

In the early stage of research on news recommendations, traditional collaborative filtering methods [1,3,5,16] were representative. In those methods, the news that people with similar preferences liked (clicked) in the past is recommended to a user. However, since the user-user similarity and recommended items are defined based on articles' ID, it is intrinsically difficult to recommend novel news articles, which is also known as the "cold start problem". News recommendation is especially sensitive to this problem since news arrives continuously and users can easily change their preferences. To overcome the cold start problem, the features of news content have been proposed. For example, by using the TF-IDF (Term Frequency-Inverse Document Frequency)-like algorithm, some methods take the contents of news articles into consideration [2,6,7]. Since TF-IDF is a technique that can extract keywords from documents, it is utilized for creating a feature vector of news articles. Furthermore, popularity, categories, sentiment information, and news location are represented as features and incorporated into the model [10,12,15,18]. However, these types of handcrafted features are usually not optimal in representing the semantic information encoded in news texts [22].

2.2 Deep Learning-Based Methods

With the surge of deep learning techniques, almost all recent recommendation models adopt neural networks. In particular, the content of news articles is captured by a deep neural network-based NLP (Natural Language Processing) technique, and users are represented by their browsing history of news articles in many cases. In Ref. [14], the authors propose the mechanism that uses DAE (Denoising Auto Encoder) and triplet loss to represent news articles. The representations of users are based on their browsing history and sessions, which are also considered in other research [8,9]. To obtain a richer representation, CNNs (Convolutional Neural Networks) and attention mechanisms are applied to news content [19–21]. For example, in NRMS [21], it applies word2vec [13] to news titles. The obtained word embeddings are further fed into a multihead self-attention mechanism to capture word relationships. A pretrained language model such as BERT is also used in news recommendations [4,24]. In addition to using a deep NLP model, some works predict user behavior, such as "active-time" and "satisfaction" [11,23]. They learn the models in a multitask fashion and improve performance.

2.3 Question

To the best of our knowledge, recent research on news recommendations mentioned above addresses the single-domain setting, and there are not enough studies, especially on deep learning-based cross-domain news recommendations. Basically, the model should be learned by using a large amount of data with good quality in many machine learning scenarios. In this context, using multiple-domain data might compensate for the amount of data in a single domain. On the other hand, it is easy to assume that users' features are different in each

news platform. We would like to find a good way to tune news articles and user embeddings in a cross-domain setting.

Our interests are summarized as follows;

1. Is news embedding much more robust if we use multiple domain data?
2. Should we consider the characteristics of domain-specific users?

We aim to answer these questions through the proposed framework.

3 Proposed Framework

To clarify the above questions, we propose pretraining the model by using all domain data and freezing news-related model parameters for fine-tuning. It is expected that pretraining with all of the domain data enables news embedding to be more robust, and domain-specific user characteristics to be expressed by fine-tuning.

We show the overview of the proposed framework in Sect. 3.1, and a detailed explanation of the proposed architecture is described in Sect. 3.2.

3.1 Overview

We consider the cross-domain situation where the data of all domains (platforms) are available such that a news company deals with multiple news brands. The data included in news domains is denoted as $\{T_i | 1 \leq i \leq N_{\text{domain}}\}$, where N_{domain} is the number of domains. T_i contains the information of user-news interactions, e.g., displayed news, clicked news, and timestamps. Please note that news is shared in each domain, but users are not shared in our assumption. Furthermore, T_i is divided into training, validation, and test sets. That is, $T_i = T_i^{\text{train}} \cup T_i^{\text{val}} \cup T_i^{\text{test}}$. Let S denote the data in which each T_i is combined. S^{train} is denoted as $S^{\text{train}} = \bigcup_i T_i^{\text{train}}$. The same holds on S^{val} and S^{test}. In our framework, the model is first learned by using S^{train} and S^{val}. Then, the model is fine-tuned by using T_i^{train} and T_i^{val} in each domain. Finally, test sets T_i^{test} and S^{test} are used for the evaluation. Figure 1 shows an example of data utilization flow.

3.2 Architecture and Procedure

We follow the recent models' architecture, which captures the contents of news articles by NLP, and users are represented by a set of browsed news embeddings. These types of methods are mentioned in Sect. 2. The following explanation accompanies Fig. 2. Let the user's browsed articles and the recommendation candidate article denote $\{D_i | 1 \leq i \leq H\}$ and D_{cand}, where H is the number of histories input to the model. Each of the browsed articles is input to news encoders whose projection is defined as f. The $\mathbf{e_i}$, i-th output of the news encoder, is formulated as

$$\mathbf{e_i} = f(D_i; \theta_{\text{news}}), \tag{1}$$

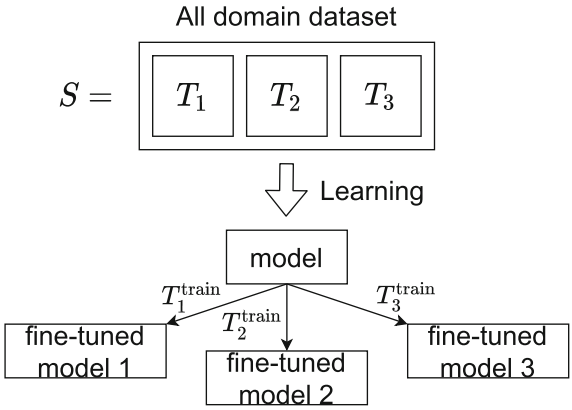

Fig. 1. Overview of the data flow. This is an example of the case where $N_{\text{domain}} = 3$. First, S is used for learning the recommendation model. The model is then fine-tuned in each domain using T_1, T_2, and T_3.

where θ_{news} is the trainable parameter. Let $\mathbf{u}$ denote an embedding of the user. $\mathbf{u}$ is expressed as

$$\mathbf{u} = g_1(E; \theta_1), \tag{2}$$

where g_1, E, and θ_1 are the aggregator & conversion function, set of $\mathbf{e_i}$, and trainable parameter, respectively. Similarly, the click prediction $\hat{p}$ is expressed as

$$\hat{p} = g_2(\mathbf{u}, \mathbf{e}_{\text{cand}}; \theta_2). \tag{3}$$

The model should output higher probability when D_{cand} is the article to be recommended, and vice versa. Since the number of news articles is generally too large, it is difficult to train the model by using all articles. To overcome that, most methods adopt a negative sampling strategy in the training phase. That is, $\mathbf{u}$, D^+ (clicked article), and $D_1^-, D_2^-, ..., D_K^-$, (K-sampled non-clicked articles) are used to train the model. Although the loss calculation depends on the base model, for example, it can be formulated as

$$\text{loss} = \sum_{i=0}^{K} \mathcal{L}(p_i, y_i), \tag{4}$$

where $y_0 = 1$, p_0 is the predicted click probability of the positive sample, $y_1, ..., y_k = 0$, $p_1, ..., p_K$ are those of the negative sample, and $\mathcal{L}$ is a loss function.

In the pretraining process, we use dataset S and train the above model parameters θ_{news}, θ_1, and θ_2 by minimizing loss using optimizers such as SGD (Stochastic Gradient Descent). In fine-tuning, T_i^{train} and T_i^{val} in each domain are used, and θ_{news} is frozen. Thus, we have i models that are dedicated to each domain.

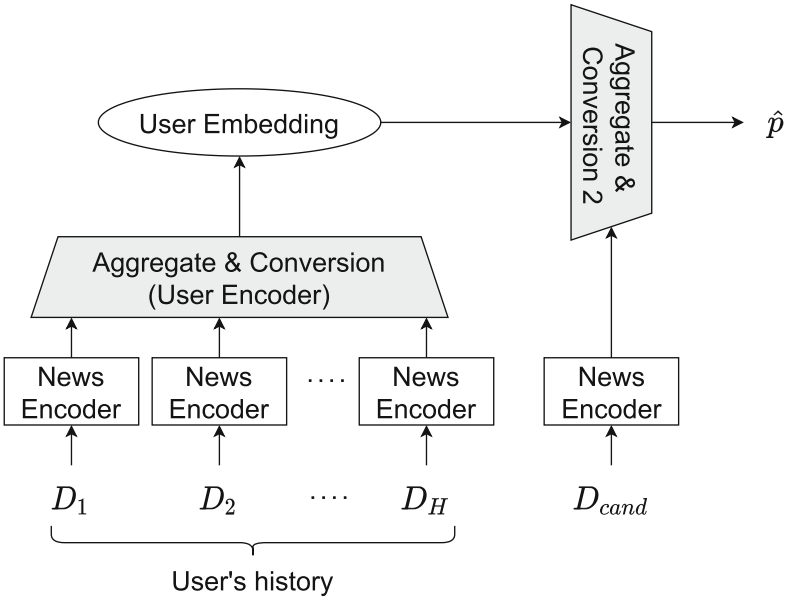

Fig. 2. Overview of the assumed model. Many works (e.g. [14,19–21]) adopt this kind of architecture and we follow that. In this paper, we basically follow NRMS [21], architecture. Parameters in components of gray are fine-tuned in each domain, and other parameters are frozen.

4 Experiments

4.1 Base Model

In the following experiments, we select NRMS [21] as the base model and apply the proposed framework with the same hyper-parameter settings. This is because its architecture is simple and brings effective results. That model adopts an NLP-based news encoder, and user embeddings are based on their browsed history, which also suits our assumption. Please note that other models can also be used for our experiments.

4.2 Dataset

We use three datasets: "Gunosy", "Newspass", and "Lucra", which are the names of news platforms run by a single company. Table 1 shows the details of dataset. As shown in Table 1, the news articles are partly shared in each dataset, and users are not shared. Further, Lucra targets female users. Thus, its user characteristics are supposed to be different from other domains. Since each article is written in Japanese, we used Japanese word embeddings (GloVe) made by asahi.com [17]. We believe they are suitable for the news recommendation task because they are made from newspaper articles. Within the shown period,

the data of 31st July 2020 are only used as the test set, and the others are the training and validation sets. For the training phase, we try two negative sampling strategies based on two sample sources: "All articles" and "Impression". When the source is "all articles", negative samples for a user are randomly chosen from all unread articles in the training set. These datasets include impression data, which is a set of news articles displayed in users' devices and also used as the negative sampling source by randomly choosing unread articles. The number of negative samples K is set to 4 by following NRMS.

Similarly, we make two types of evaluation data from test sets. In the first case, we randomly choose 200 negative samples from all articles in the test set for each user. In the second case, we use all negative samples in the impression data for each user. We denote "A" and "I" as negative sampling source ("All articles" and "Impression", which are mentioned above). For example, (A, I) indicates that train set contains negative samples picked from All articles and test set contains negative samples picked from Impression.

Table 1. The number of users and articles in each dataset. G, N, and L indicate Gunosy, Newspass, and Lucra, respectively. Shared users and articles are represented by X ∩ Y. We picked a maximum of 3,000 users who have a larger number of clicks from each domain and the articles they clicked were extracted.

Dataset	# User	# Article	Period
G	3000	28893	From 27th to 31st in July, 2020.
N	3000	26484	
L	817	10273	
G ∩ N	0	12942	
G ∩ L	0	2439	
N ∩ L	0	2058	
G ∩ N ∩ L	0	1455	

4.3 Metrics

Following NRSM, we evaluate AUC (area under the ROC curve), MRR (mean reciprocal rank), and nDCG@k (normalized discounted cumulative gain at k). All take values between 0.0 to 1.0. When the value is 1.0, it indicates that the model is perfect.

– **AUC:** AUC is a widely used metric and indicates overall performance. It takes a higher value when positive samples tend to rank higher than negative samples in the recommendation list.

- **MRR:** When positive samples tend to hit at the top of the recommendation list, MRR takes a higher value. In contrast, the value hardly improves when positive samples hit the bottom of the list.
- **nDCG@k:** For the top k news articles in the recommendation list, articles that should be highly recommended but appearing lower in a list are penalized. We evaluate the case of $k = 5$ and $k = 10$.

4.4 Performance

Overall Results. Table 2 shows the performance of the model in each dataset pair. As we can see from Table 2, learning with G+N+L and fine-tuning tend to achieve better results regardless of dataset and negative sampling sources. In many cases, the overall performance (AUC) and the quality of recommendation (MRR, nDCG) improved. This reflects the effectiveness of the proposed framework. Especially, AUC of Lucra is effectively improved compared with single domain setting. We consider this is because the number of articles in Lucra is relatively small and it targets female users. This is the just situation where the proposed framework seems to effectively work. Learning with G+N+L enables for news embeddings to be much more robust and the characteristics of users can be captured by domain-specific fine-tuning. On the other hand, the improvements in Gunosy and Newspass are relatively small. This is because Gunosy and Newspass share about 50% news articles and the user characteristics between them seem to be similar. In addition to that, the fact that Lucra's articles are targeting female users might be another reason. Even if the information of Lucra's article is reflected in the model, it does not impact on the recommendation results very much in those domains or works as noise in some cases.

Comparing negative sampling strategies, the models tested by "All articles" sampling (*, A) achieve better performance. In impression data, displayed news articles include the effect of recommendation system which have already been working. Thus, using impression data is more practical case and classification becomes more difficult. Although the results are better when training and testing adopt the same source, we cannot judge the superiority between training with All articles and training with Impression.

From the business perspective, this result implies that it is possible to transfer model trained by many articles to other platform dealing with similar news articles. Since it is unnecessary to share the user information in this framework, there is no privacy concern in providing the model. This is a useful merit and we consider there are many situation that the proposed framework can be applied in real setting.

Table 2. The performance on each dataset. G, N, and L are the same as Table 1. If the models are fine-tuned, the value on the column "FT" is True. NS indicates Negative Samples used in dataset. Test set results are shown. For fine-tuned models, the result of final epoch is shown. Bold values are the best results in the same NS-test pair.

Dataset			FT	AUC	MRR	nDCG	
NS	Train	Test	–	–	–	@5	@10
(A, A)	G	G	False	0.7310	0.0538	0.0996	0.1128
	G+N+L	G	False	0.7295	0.0577	0.1198	0.1283
	G+N+L	G	**True**	**0.7341**	**0.0612**	**0.1329**	**0.1351**
	N	N	False	0.7002	0.0631	0.1408	0.1395
	G+N+L	N	False	0.7035	0.0589	0.1409	0.1283
	G+N+L	N	**True**	**0.7136**	**0.0642**	**0.1628**	**0.1485**
	L	L	False	0.7173	0.0646	0.1879	0.1712
	G+N+L	L	False	0.7674	0.0952	**0.2877**	0.2272
	G+N+L	L	**True**	**0.7791**	**0.1013**	0.2873	**0.2428**
	G+N+L	G+N+L	False	0.7270	0.0581	0.1235	0.1293
(A, I)	G	G	False	0.6130	0.0441	0.0888	0.0867
	G+N+L	G	False	0.6269	0.0475	0.0978	0.1020
	G+N+L	G	**True**	**0.6276**	**0.0495**	**0.1048**	**0.1045**
	N	N	False	0.5646	0.0301	0.0486	0.0527
	G+N+L	N	False	0.5706	0.0298	**0.0544**	0.0457
	G+N+L	N	**True**	**0.5794**	**0.0327**	0.0492	**0.0532**
	L	L	False	0.5107	0.0290	**0.0509**	0.0409
	G+N+L	L	False	0.5584	**0.0297**	0.0345	**0.702**
	G+N+L	L	**True**	**0.5641**	0.0294	0.0476	0.0530
	G+N+L	G+N+L	False	0.6165	0.0444	0.0893	0.0934
(I, I)	G	G	False	0.6594	0.0605	0.1484	0.1367
	G+N+L	G	False	0.6467	0.0579	0.144	0.1294
	G+N+L	G	**True**	**0.6513**	**0.0628**	**0.1578**	**0.1422**
	N	N	False	0.5843	0.0394	0.0743	0.0716
	G+N+L	N	False	0.5945	**0.0415**	**0.0910**	0.0759
	G+N+L	N	**True**	**0.5976**	0.0406	0.0744	**0.0772**
	L	L	False	0.5472	0.0299	0.0616	0.0469
	G+N+L	L	False	0.5615	0.0314	0.0578	0.0733
	G+N+L	L	**True**	**0.5696**	**0.0377**	**0.0845**	**0.0962**
	G+N+L	G+N+L	False	0.6372	0.0544	0.1315	0.1181
(I, A)	G	G	False	0.724	0.0661	0.1562	0.1529
	G+N+L	G	False	0.7164	0.0643	0.1523	0.1453
	G+N+L	G	**True**	**0.7225**	**0.0688**	**0.1666**	**0.1580**
	N	N	False	0.6569	0.0562	0.1227	0.118
	G+N+L	N	False	0.6867	**0.0752**	0.2021	0.1729
	G+N+L	N	**True**	**0.6916**	0.0724	**0.2029**	**0.1732**
	L	L	False	0.6375	0.0723	0.1402	0.1621
	G+N+L	L	False	0.7236	0.0783	0.2291	0.1930
	G+N+L	L	**True**	**0.7317**	**0.0845**	**0.2788**	**0.2082**
	G+N+L	G+N+L	False	0.7134	0.0657	0.1587	0.1499

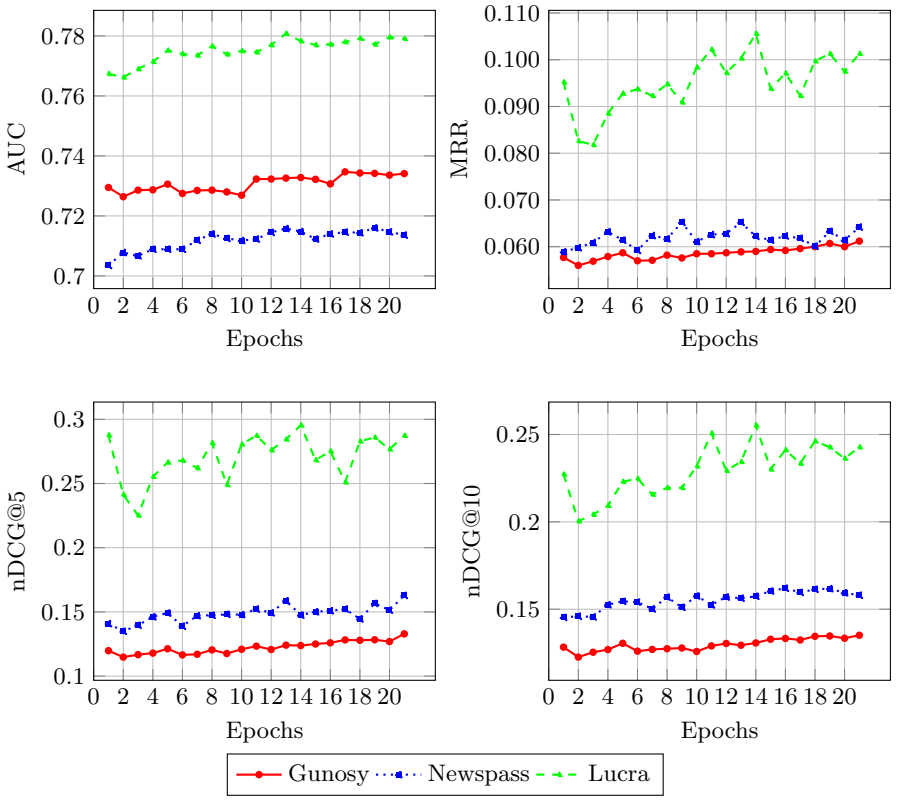

Fig. 3. The performance (AUC, MRR, nDCG@5, and nDCG@10) under fine-tuning. Gunosy(A, A), Newspass(A, A) and Lucra(A, A) are used as dataset.

Fine-Tuning Performance. Figure 3 shows the performance under fine-tuning. We only show the result of fine-tuning in Gunosy(A, A), Newspass(A, A), and Lucra(A, A) since the tendency is almost the same in other negative sampling pairs. In this experiment, although recommendation quality measures keep rising as epochs increases, we stop fine-tuning in epoch 20 since there is no large improvement.

As we can see from Fig. 3, the larger epochs become, the larger the metrics are. While the improvement is relatively large in Lucra, the results are saturated in other domains. This is the same reason mentioned in Sect. 4.4. We can say that domain-specific fine-tuning is effective in the domain whose user characteristics are different from pre-trained model and it is marginal in other domains. In practical use, domain-specific fine-tuning can be skipped in this kind of domains.

5 Conclusion

In this paper, we described a simple model for news recommendation tasks in multiple domain settings that uses freezing parameters and fine-tuning. Through experiments using the proposed framework on a real-world dataset, we found that a learning model using multiple domain data is effective for obtaining robust news embeddings. Moreover, our empirical results imply that the characteristics of domain-specific users can be captured from a multiple domain model by fine-tuning the domain. In particular, the proposed framework is effective in domains whose number of data points is small. As a future work, we would like to try additional experiments by changing the base model and clarify the effective type of models for cross-domain news recommendations.

Acknowledgements. The authors would like to thank Kojiro Iizuka (Gunosy) and Yoshifumi Seki (Gunosy) for their support and discussions. This research was partially supported by JST CREST Grant Number JPMJCR21F2, Japan.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Capelle, M., Frasincar, F., Moerland, M., Hogenboom, F.: Semantics-based news recommendation. In: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, pp. 1–9 (2012)
3. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web, pp. 271–280 (2007)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Dwivedi, S.K., Arya, C.: A survey of news recommendation approaches. In: 2016 International Conference on ICT in Business Industry & Government (ICTBIG), pp. 1–6. IEEE (2016)
6. Gershman, A., Wolfe, T., Fink, E., Carbonell, J.G.: News personalization using support vector machines. Carnegie Mellon Univ. J. Contrib. (2011). https://www.semanticscholar.org/paper/News-Personalization-using-Support-Vector-Machines-Gershman-Wolfe/c665575cabc19aaba2cdf775602494cdd46c59fb
7. Goossen, F., IJntema, W., Frasincar, F., Hogenboom, F., Kaymak, U.: News personalization using the CF-IDF semantic recommender. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics, pp. 1–12 (2011)
8. Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 843–852 (2018)
9. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
10. Lee, H.J., Park, S.J.: MONERS: a news recommender for the mobile web. Expert Syst. Appl. **32**(1), 143–150 (2007)

11. Liu, R., Peng, H., Chen, Y., Zhang, D.: HyperNews: simultaneous news recommendation and active-time prediction via a double-task deep neural network. In: IJCAI, pp. 3487–3493 (2020)
12. Liu, S., Dong, Y., Chai, J.: Research of personalized news recommendation system based on hybrid collaborative filtering algorithm. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 865–869. IEEE (2016)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
14. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1933–1942 (2017)
15. Parizi, A.H., Kazemifard, M., Asghari, M.: EmoNews: an emotional news recommender system. J. Digit. Inf. Manag. **14**(6), 392–402 (2016). https://www.researchgate.net/profile/Mohammad-Kazemifard/publication/313574529_Emonews_An_emotional_news_recommender_system/links/5e3bcdb5299bf1cdb9116783/Emonews-An-emotional-news-recommender-system.pdf
16. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186 (1994)
17. Taguchi, Y., Tamori, H., Hitomi, Y., Nishitoba, J., Kikuta, K.: 同義語を考慮した日本語の単語分散表現の学習. IPSJ SIG Technical Report 2017-NL-233, no. 17, pp. 1–5 (2017)
18. Tavakolifard, M., Gulla, J.A., Almeroth, K.C., Ingvaldesn, J.E., Nygreen, G., Berg, E.: Tailored news in the palm of your hand: a multi-perspective transparent approach to news recommendation. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 305–308 (2013)
19. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 1835–1844 (2018)
20. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2576–2584 (2019)
21. Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6389–6394 (2019)
22. Wu, C., Wu, F., Huang, Y.: Personalized news recommendation: a survey. arXiv preprint arXiv:2106.08934 (2021)
23. Wu, C., Wu, F., Qi, T., Huang, Y.: User modeling with click preference and reading satisfaction for news recommendation. In: IJCAI, pp. 3023–3029 (2020)
24. Wu, C., Wu, F., Qi, T., Huang, Y.: Empowering news recommendation with pre-trained language models. arXiv preprint arXiv:2104.07413 (2021)
25. Zhong, E., Liu, N., Shi, Y., Rajan, S.: Building discriminative user profiles for large-scale content recommendation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2277–2286 (2015)