该文参考了微软亚洲研究院的多数新闻推荐的模型方法，提出了基于图神经网络的多视角学习方法，其主要包含新闻内容角度建模和user-news关系视角建模。该文用到的用户表示方法和新闻表示方法与NAML相似，可以看作是其之上加入图网络的改进版本。

MVL: Multi-View Learning for News Recommendation

T.Y.S.S.Santosh
santoshtyss@gmail.com
IIT Kharagpur, India

Avirup Saha
saha.avirup@gmail.com
IIT Kharagpur, India

Niloy Ganguly
ganguly.niloy@gmail.com
IIT Kharagpur, India

Abstract

In this paper, we propose a Multi-View Learning (MVL) framework for news recommendation which uses both the content view and the user-news interaction graph view. In the content view, we use a news encoder to learn news representations from different information like titles, bodies and categories. We obtain representation of user from his/her browsed news conditioned on the candidate news article to be recommended. In the graph-view, we propose to use a graph neural network to capture the user-news, user-user and news-news relatedness in the user-news bipartite graphs by modeling the interactions between different users and news. In addition, we propose to incorporate attention mechanism into the graph neural network to model the importance of these interactions for more informative representation learning of user and news. Experiments on a real world dataset validate the effectiveness of MVL.

CCS Concepts

• **Information systems** → *Collaborative and social computing systems and tools*; *Social tagging systems*; **Recommender systems**.

Keywords

news recommendation, graph view, multi-view learning

ACM Reference Format:
T.Y.S.S.Santosh, Avirup Saha, and Niloy Ganguly. 2020. MVL: Multi-View Learning for News Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401294

1 Introduction

With the rapid development of the Internet, people's news reading habits have gradually shifted from traditional media such as newspapers and TV to the Internet [5]. Therefore, personalized news recommendation have become a necessity to help users find their news of interest among the myriad of news produced online [6].

As a result, in recent times several deep learning based methods [1, 5–9, 11] have been proposed for task of recommendation. [5] proposed to learn news representations from news bodies via denoising auto-encoders, and learn user representations from the representations of their browsed news via a gated recurrent unit (GRU) network. [6] proposed to learn news representations from news titles via a knowledge-aware convolutional neural network (CNN) and learn user representations from news representations using the similarity between candidate news and browsed news. [11] uses a deep attention based neural network by considering the users click sequence information. [7] proposed to learn news representations by incorporating titles, bodies and categories as different views of news. [9] proposed to apply multi-head self-attentions to learn contextual word and news representations by modeling the interactions between words and news. [1] proposed an approach to learn both long-and short-term user representations where it learns long-term representations of users from the embeddings of their IDs and short-term representations of users from their recently browsed news via a GRU network. [8] proposed a personalized attention network which exploits the embeddings of user ID as the queries of the word- and news-level attention networks. There are also matrix factorization based model [10] to process vectors of users and news articles. The model ignores the news contents and take the implicit feedback as its input.

However all the above approaches do not explicitly capture the **user-user** and **news-news relatedness** in the modelling process which can be obtained from **user-news interaction graph**. In this paper, we propose a **Multi-View Learning (MVL)** framework which uses both the content and the user-news interaction graph. There are two major modules in MVL. **(a).** A content view module to learn representations of candidate news articles and users with the help of news content view; news representations is learned from various information like titles, bodies and categories. We learn user representations using representations of historical news browsed by user conditioned on the news article to be recommended, **(b).** A graph view module where we propose to use a graph neural network to learn representations of users and news articles from user-news interaction graph view by modeling the second-order interactions in the user-news graph. Finally we use an output layer to predict the probability of a user browsing that candidate news article. Experiments on a real world dataset validate the effectiveness of MVL.

2 Our Approach: MVL

We first define the problem and then elaborate the various modules one by one.

2.1 Problem Definition and Output Layer

Given a user u_x along with his recently browsed news articles and a candidate news item n_x to be recommended, our goal is to predict the probability of the user browsing that candidate news. The final representations of user u and news n are the concatenation of the representations learned from the content view and graph view (discussed next) i.e., $u = [u_c, u_g]$ and $n = [n_c, n_g]$. The click probability score y is calculated by the inner product of the representation vectors of user u_x and the candidate news n_x as $y = u_x^T n_x$.

2.2 Content View $[n_c, u_c]$

In the core of this module is a news encoder which is used to learn the representations of the news article and also the representation

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401294

of the user with the aid of historical news browsed by him/her conditioned on the news article to be recommended.

2.2.1 News Encoder [n_c] The news encoder is used to learn representations of news from different kinds of information such as title, body and topic categories. Since different kinds of news information have different characteristics, instead of simply merging them for news representation, we learn unified news representations by aggregating them using attention mechanism .

Title encoder. The first component is the title encoder, which is used to learn news representations from their titles. We first convert a news title to a word sequence $[w_1^t, w_2^t, ..., w_n^t]$, where n is the length of this title, into a sequence of low-dimensional vectors $[e_1^t, e_2^t, ..., e_n^t]$ using pre-trained word embeddings. We then feed them into a convolutional neural network (CNN) [4] for learning local contexts of words in news titles. We obtain the contextual representation of the i^{th} word as c_i^t, which is calculated as

$$c_i^t = ReLU(F_t \times e^t_{(i-K):(i+K)} + b_t) \tag{1}$$

where $e^t_{(i-K):(i+K)}$ is the concatenation of word embeddings from position (i-K) to (i+K). F_t and b_t are the kernel and bias parameters of the CNN filters. This finally outputs a sequence of contextual word representations $[c_1^t, c_2^t, ..., c_n^t]$. We use a word-level attention network to select important words within the context of each news title to learn more informative news representations.

$$a_i^t = q_t^T \tanh(V_t \times c_i^t + b_t) \tag{2}$$

where V_t and b_t are the projection parameters, q_t denotes the attention query vector and is learned during training. We obtain the attention weight of the i^{th} word in a news title α_i^t as:

$$\alpha_i^t = \frac{\exp(a_i^t)}{\sum_{j=1}^n \exp(a_j^t)} \tag{3}$$

The final representation of a news title r^t is the summation of the contextual representations of its words weighted by their attention weights.

$$r^t = \sum_{j=1}^n \alpha_j^t c_j^t \tag{4}$$

Body and Category Encoder. The next components in news encoder module are body encoder, which is used to learn a news representation from the content present in the body and category encoders, which proposes to incorporate the category information in news representation. Similar to the title encoder, the body and category encoders obtain the contextual representations using CNN which takes word embeddings as input and then output final representation as r^b for a news body and r^c for a news category respectively as the summation of the contextual word representations weighted by their attention weights.

Attentive Network. We propose a news-level attention network to model the informativeness of the above three different categories. First we obtain the attention weights of title, body, category α_t, α_b and α_c respectively. The attention weight of the title view is calculated as

$$a_t = q_v^T \tanh(U_v \times r^t + u_v) \;\&\; \alpha_t = \frac{\exp(a_t)}{\exp(a_t) + \exp(a_b) + \exp(a_c)} \tag{5}$$

where U_v and u_v are projection parameters, q_v is the attention query vector and is learned during training. The attention weights of body and category can be computed in a similar way. The final unified news representation n_c learned by the news encoder module is the summation of the news representations from different views weighted by their attention weights as

$$n_c = \alpha_c r^c + \alpha_t r^t + \alpha_b r^b. \tag{6}$$

2.2.2 Obtaining user representation [u_c] We obtain representation of user u_c with the help of his/her browsed news conditioned on the candidate news article to be recommended n_x. Consequently, the user-level attention network used to model the different impacts of the user's recently clicked news is conditioned on n_x. Let the obtained news representations of his browsed news and candidate news article to be recommended from news encode be $[n_1, n_2, ..., n_l]$ and n_x respectively. We compute the attention weight α^j of the j^{th} news that the user u browsed w.r.t. n_x as

$$a_j = \tanh(W_1 n_j + b_1) \;\&\; a = \tanh(W_2 n_x + b_2) \tag{7}$$

$$\alpha_j = \frac{\exp(v^T(a + a_j))}{\sum_{k=1}^l \exp(v^T(a + a_k))} \tag{8}$$

We obtain the final content representation of user u_c as summation of the news representations browsed by this user weighted by their attention weights.

$$u_c = \sum_{j=1}^l \alpha_j n_j \tag{9}$$

2.3 Graph View [u_g, n_g]

In the content view, we tried to implicitly capture first order interactions for users using their browsed articles. To directly capture second-order interactions for users and both first and second orders for news, we propose a hierarchical attentive graph neural network for learning representation. We first enhance the static user-news bi-partite graph by representing each node (either user or news) with the embedding obtained from content view. Then use this graph to mine the user-user and news-news relatedness by modeling the first and the second-order (hop) interactions of users and news in the graph.

Learning u_g : Let the news that a user u browsed be $[n_1, n_2, ..., n_P]$ and the users who browsed the news n_i be $[u_{i1}, u_{i2}, ..., u_{iK}]$. We first learn the representations of each news from the representations of the users who have browsed this news. Usually different users who browsed the same news may have different informativeness in representing this news. Thus, we propose to use an attentive graph neural network to model the importance of the users that are connected to the news node. The attention weight α_{ij}^d of the j^{th} user who browses the news n_i is computed as:

$$a_{ij}^d = \tanh(w_g \times u_{ij} + b_g) \;\&\; \alpha_{ij}^d = \frac{\exp(a_{ij}^d)}{\sum_{l=1}^K \exp(a_{il}^d)} \tag{10}$$

where w_g and b_g are projection parameters. The user-based representation n_i^u of the news n_i is the summation of the embeddings of its related user weighted by their attention weights, which is

formulated as

$$n_i^u = \sum_{l=1}^{K} \alpha_{il}^d u_{il} \tag{11}$$

Once the embedding of the news entity is obtained, we then learn the final graph-based representation u_g of the user u from the representations of its neighbor news nodes in an attentive manner. We compute the attention weight α_i^d of the i^{th} news that the user u browsed as

$$a_i^d = \tanh(w_h \times n_i^u + b_h) \;\&\; \alpha_i^d = \frac{\exp(a_i^d)}{\sum_{l=1}^{P} \exp(a_l^d)} \tag{12}$$

where w_h and b_h are projection parameters. The graph-based representation u_g of the user u is calculated as summation of the news representations associated with this user weighted by their attention weights.

$$u_g = \sum_{l=1}^{P} \alpha_l^d n_l^u \tag{13}$$

Graph view of the news n_g is computed similarly in a hierarchical manner.

2.4 Model Training

We use negative sampling techniques for model training. For each news browsed by a user which is regarded as a positive sample, we randomly sample K news articles which are not clicked by this user as negative samples. We then jointly predict the click probability scores of the positive news y^+ and the K negative news $[y_1^-, y_2^-, ..., y_K^-]$. In this way, we formulate the news click prediction problem as a pseudo K+1 way classification task. We normalize these click probability scores using softmax to compute the posterior click probability of a positive sample as follows:

$$p_i = \frac{\exp(y_i^+)}{\exp(y_i^+) + \sum_{j=1}^{K} \exp(y_{i,j}^-)} \tag{14}$$

where y_i^+ is the click probability score of the i^{th} positive news, and $y_{i,j}^-$ is the click probability score of the j^{th} negative news in the same session with the i^{th} positive news. The loss function L in MVL for model training is the negative log-likelihood of all positive samples, which can be formulated as

$$L = -\sum_{i=1}^{S} \log(p_i) \tag{15}$$

where S is the size of the set of the positive training samples.

3 Experiments

We conduct experiments on a real-world online news dataset Adressa [2] from a Norwegian news portal where two versions named Adressa-1week and Adressa-10week are available, which separately collect as long as 1 week (from 1 January to 7 January 2017) and 10 weeks (from 1 January to 31 March 2017). For the Adressa-1week dataset, we split the data as the first 6 days history data for training pairs and the last day for testing. Similarly, for the Adressa-10week dataset, in training period, we use the previous 50 days'data for training, the rest 20 days for testing. We set historical news browsed by user length in both the content view and the graph view to 15. In our experiments, we have used pre-trained word2vec word embeddings which are 100-dimensional. CNN network consists of 400 filters and their window size is kept to 3. The dimension of attention query vectors was kept to 100. The negative sampling ratio K is set to 4. We have used Adam as the optimization algorithm and the batch size is set to 64. We applied 20% dropout to each layer in our approach to mitigate overfitting. These hyper-parameters were selected according to the validation set. The metrics used for result evaluation in our experiments include F1 and AUC. We repeated each experiment 10 times and reported the average results.

3.1 Performance Evaluation

We evaluate the performance of MVL by comparing it with the following baselines: DMF [10], DeepFM [3] DKN [6], DAN [11], NAML [7],NRMS [9], LSTUR [1], NPA [8],

According to Table 1, we observe that the methods based on neural networks outperform traditional matrix factorization methods. This can be because neural networks can learn better news and user representations than traditional matrix factorization methods. We observe that our model performs better than all the baselines. Different from baseline methods, MVL uses a multi-view learning framework to incorporate different views. This result validates the effectiveness of MVL.

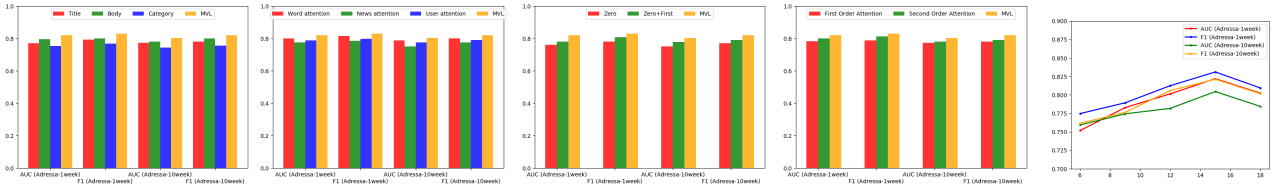
3.2 Effectiveness of Content View

In this section, we conducted several experiments to validate the effectiveness of Content View module in our model. First, we explore the effectiveness of incorporating various information of content in MVL. The performance of MVL and its variants with different combinations is shown in Fig 1a. First, the model with the body information achieves better performance than those with the title or category information only. This is intuitive because the bodies of news usually contain the original information of news and can provide rich information for modeling news topics. Second, the title and category information are also informative for news recommendation. This is probably because titles usually have decisive influence on users' reading behaviors. Thus, incorporating news titles is useful for modeling the characteristics of news and users. In addition, since categories of news are important clues of news topics, incorporating the category view is also useful for recommendation. Third, combining all three information further improved the performance of MVL. These results validate the effectiveness of incorporating various information in our content view.

Next, we conducted experiments to validate the effectiveness of the attention mechanism at word-level, news-level and user-level. The performance of MVL and its variants with different combinations of attention networks is shown in Fig. 1b. We observe that the word-level attention network achieves better performance than those with the news or user attention only. This is probably because words are basic units in titles and bodies to convey their meanings and different words usually have different informativeness for learning news representations. Our approach can recognize and highlight the important words, which is useful for learning more informative news representations. The improvements via news-level attention and user-level attention are marginal

3.3 Effectiveness of Graph View

In this section, we conducted several experiments to validate the effectiveness of Graph View in our approach. First we explore the effectiveness of incorporating user-item interaction through graph.



(a) Information of cv (b) Attention in cv (c) Depth in graph view (d) Attention in graph view (e) Length of user history

Figure 1: Effectiveness of content view, graph view and length of user history. cv represents content view.

We compare the performance of MVL with its variants involving only the zero-order or with both the zero and first-order interactions in Fig. 1c. We observe that the performance of MVL can be consistently improved as the depth increases. This is because the first-order information of graph contains the interactions between users and items and the second-order information can reveal the user-user and item-item relatedness. Thus, more information can be incorporated as the depth increases, which can benefit user and item representation learning.

Next, we conducted experiments to validate the effectiveness of the attention network in the graph-view. We compare our MVL with its variants by removing the attention networks in the first-order or second-order encoder to validate their effectiveness and the results are shown in Fig. 1d. We find that the attention networks in both first- and second-order encoders are important in our MVL.

3.4 Effect of length and variety of user history

In the dataset for each user generally a very long length (> 200) of historical information is provided. However, we check whether such long history is useful or not. We conduct several experiments to study the effect of length of user history on the performance by restricting the number of historical news browsed by user in the model. From Fig. 1e, we observe that the performance of our model first improves with the increase of length of user history (upto around 15) and then drops as the length further increases. This shows that while too small history length has insufficient capability of capturing the necessary information regarding the interests of the user, too large history length introduces unnecessary noise and reduces generalization ability.

We built a difficult history subset via selecting only users who have historically browsed more categories of news than the median of the original dataset (i.e. these users have diverse interest). MVL achieves AUC of 0.6234 and F1 of 0.6522 whereas the state of the art model NAML achieves AUC of 0.5916 and F1 of 0.6274. This indicates that as expected the performance of MVL deteriorates but that is also true for baselines. In fact, in this zone the proportion of improvement of MVL over baselines is slightly better than average.

4 Conclusion

The Multi-View Learning framework proposed beats the baseline by 3-4% which is a significant improvement considering the number of works already done in the space. This improvement can be ascribed to the deft design of incorporating both the content and the graph view through recursive use of attentive network. We make a thorough ablation study and note interesting observations e.g. just pumping the system with huge historical data may not be useful. Also we find that avoidance of explicit feature engineering provides the necessary flexibility whereby MVL's performance improvement over baselines is more than average when a user changes her preferences frequently.

Methods	Adressa-1week		Adressa-10week	
	AUC	F1	AUC	F1
DMF	0.5566	0.5646	0.5320	0.5415
DeepFM	0.6909	0.6148	0.7404	0.6582
DKN	0.7557	0.7611	0.7432	0.7229
DAN	0.7593	0.7401	0.7676	0.7165
LSTUR	0.7615	0.7512	0.7712	0.7465
NRMS	0.7735	0.7686	0.7809	0.7639
NAML	0.7918	0.8041	0.7970	0.7822
NPA	0.7897	0.7805	0.7922	0.7747
MVL	**0.8220**	**0.8311**	**0.8046**	**0.8212**

Table 1: Performance comparison of different methods

References

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.

[2] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *Proceedings of the international conference on web intelligence*. 1042–1048.

[3] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[4] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[5] 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*. 1933–1942.

[6] 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*. 1835–1844.

[7] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. *arXiv preprint arXiv:1907.05576* (2019).

[8] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2576–2584.

[9] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. 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)*. 6390–6395.

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

[11] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5973–5980.