MIND News Recommendation Competition

U Kang
Department of Computer Science
Seoul National University
DeepTrade Inc.
ukang@snu.ac.kr

Abstract

How can we accurately recommend news to users? Microsoft released MIND, a large-scale and high-quality news dataset constructed from user click logs in a Microsoft news platform. This report is the summary of the work of "dtsnu" in MIND news recommendation competition. We elaborate on how we achieved a high recommendation performance on MIND dataset.

1 Introduction

This report is the summary of the work of "dtsnu" on MIND news recommendation competition which was organized by Microsoft News and Microsoft Research Asia teams. The goal of the competition is to predict the rankings of candidate news. The details of the MIND dataset and competition are available at https://msnews.github.io/competition.html. Our final model achieves AUC: 0.7114, MRR: 0.3568, NDCG@5: 0.3916, NDCG@10: 0.4485 on the full TEST dataset. In the rest of this report, we describe state-of-the-art news recommendation algorithms in Section 2, our proposed method in Section 3, experiments in Section 4, and conclusion in Section 5.

2 State of the Art News Recommendation Algorithms

In news recommender systems, three are three crucial tasks: 1) learning representations of news, 2) learning representations of users, and 3) learning relations between users and candidate news. To address these three tasks, previous news recommender systems use three core modules: *news encoder*, *user encoder*, and *click predictor*. We discuss previous state-of-the-art news recommender systems according to how they model the three core modules.

2.1 Neural News Recommendation with Attentive Multi-View Learning [5]

Wu et al. proposed a neural news recommendation approach with attentive multi-view learning (NAML) [5] which learns representations of news and users by exploiting different types of news information.

News Encoder The news encoder of NAML learns the representation of news from its title, body, and category. To encode each type of information, the news encoder is composed of three components: *title encoder*, *body encoder*, and *category encoder*. Title encoder translates a sequence of words in a title into a single title vector; body encoder translates a sequence of news in a user history into a single body vector; category encoder translates a category into a single category vector. The title encoder takes embedding vectors of words in a title and learns contextual word representations by a convolutional neural network (CNN) [2]. The encoded word vectors then are aggregated to a single vector by a word-level attention network. The body encoder takes embedding vectors of words in a body, learns contextual word representations by a CNN, and aggregates the encoded word vectors

using a word-level attention network. The category encoder takes an embedding vector of a category and learns a hidden representation using a fully-connected neural network. The news encoder then aggregates encoded vectors of the three components by an attentive multi-view network.

User Encoder The user encoder of NAML learns the representation of a user from his browsed news history. Each news browsed by a user is encoded into a single vector by the news encoder. The user encoder then aggregates encoded vectors of browsed news using a news-level attention network to recognizing important news in the history.

Click Predictor The click predictor of NAML predicts scores between a user and candidate news. Each candidate news is encoded into a single vector by the news encoder and the score is predicted by an inner-product between a user vector encoded by the user encoder and a candidate news vector.

2.2 Neural News Recommendation with Multi-Head Self-Attention [6]

Wu et al. proposed a neural news recommendation approach with multi-head self-attention (NRMS) [6] which learns the representations of news and users by using multi-head self-attention networks [4].

News Encoder The news encoder of NRMS learns representations of news from their titles. The interactions between words in a title are crucial to learning news representations. The news encoder takes embedding vectors of words in a title and encodes them using a multi-head self-attention network to learn the complicated contextual representations. The encoded word vectors then are aggregated to a single vector by a word-level attention network to recognize important words in the title.

User Encoder The user encoder of NRMS learns the representation of a user from his browsed news history. Similar to the interactions between words in a title, those between news in a user history are crucial to learning user representations. The user encoder takes news vectors encoded by the news encoder in a user history and encodes them using a multi-head self-attention network to learn the contextual representation of a user. The encoded news vectors then are aggregated to a single vector by a news-level attention network to recognize important news in the history.

Click Predictor NRMS predicts scores between a user and candidate news in the same way as NAML. Each candidate news is encoded into a single vector by the news encoder and the score is predicted by an inner-product between a user vector encoded by the user encoder and a candidate news vector.

3 Proposed Method

3.1 Overview

Our final method is an ensemble of two state-of-the-art algorithms. We use multi-head self-attention (NRMS) with attentive multi-view learning (NAML) for extracting news-embeddings, and BERT or Glove for extracting word embeddings from the titles and abstracts. The details of each combination are explained in the following sections.

3.2 NRMS+NAML+Glove

A multi-head self-attention network is capable of representing more complicated patterns in a text than Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN) since it reflects the information of all word-wise pairs in the sequence. In addition, auxiliary information such as abstract and category of news help the generalization of the recommendation performance. To combine the advantages of the two aforementioned approaches, we propose a novel news encoder combining that of NRMS and that of NAML. The news encoder is composed of four components: *title encoder*, *abstract encoder*, *category encoder*, and *sub-category encoder*. The title encoder and the abstract encoder take embedding vectors of words in a title and in an abstract; they then learn contextual word representations in the title and the abstract by multi-head self-attention networks, respectively.

We use the Glove [3] for initialization of the word embedding vectors; we also use two-layered multi-head self-attention networks for both title and abstract encoders based on experimental results. The category encoder and the sub-category encoder take embedding vectors of a category and a sub-category; they then learn hidden representations using fully-connected neural networks. The hidden vectors encoded by the four encoders are aggregated to a single vector by an attentive multi-view network. We construct a user encoder and a click predictor in the same way as those of NRMS.

3.3 NRMS+NAML+BERT

As a second approach, we utilize large BERT [1] to get more sophisticated word embeddings, and we get the pre-trained model from the transformers package[1]. However, the enormous number of parameters over 300M incurs a problem of too long training time; it requires more than two days for a single epoch when we finetune the pre-trained large BERT model. Instead, we fix the parameters of BERT and stack three MLP layers with non-linear activation functions on top of BERT to efficiently extract useful features. Other settings of the model including two-layered multi-head self-attention and attentive multi-view learning are the same as the first model explained in Section 3.2.

3.4 Ensemble

We generate ensembles of models using the two base models just discussed. Although there are only two base models, we generate many members of ensembles by selecting different sets of hyperparameters.

4 Experiments

4.1 Overview

We conduct extensive experiments to maximize the performance of our model. We find the best hyperparameters for each model using demo dataset, which is the smallest dataset provided by the competition organizers. Then, we train the final models on the large dataset with the best hyperparameter configurations and generate ensembles using them. We compare four metrics - AUC, MRR, NDCG@5, and NDCG@10 - according to the rule of the competition. All of our experiments are performed on workstations with RTX 2080Ti and GTX 1080Ti.

4.2 Hyperparameter Search

NRMS+NAML+Glove (NNG) We compare the models trained with learning rate $l \in \{0.0001, 0.0002\}$, dropout rate $r \in \{0.16, 0.2, 0.24\}$, and depth of multi-head self-attention layers $n_l \in \{1, 2\}$. As a result, we find that learning rate $l = 0.0001$, dropout rate $r = 0.2$, and self-attention layer $n_l = 2$ are the best hyperparameters.

NRMS+NAML+BERT (NNB) We compare the models trained with word embedding size $w_s \in \{200, 256, 400\}$, learning rate $l \in \{0.0001, 0.0002\}$, dropout rate $r \in \{0.16, 0.2, 0.24\}$, and depth of multi-head self-attention layers $n_l \in \{1, 2\}$. As a result, we find that word embedding size $w_s = 256$, learning rate $l = 0.0002$, dropout rate $r = 0.16$, and self-attention layer $n_l = 2$ are the best ones.

4.3 Ensemble

We select two NRMS+NAML+Glove models and four NRMS+NAML+BERT models as final models to generate ensembles. The hyperparameter descriptions and performance on DEV dataset of our final models are summarized in Table 1. The models M1-1 and M1-2 are two variants of the NNG model, with one and two multi-head self-attention layers, respectively. The model M2-1 is a primitive model that uses only title information, thus it does not use attentive multi-view learning. We include this model since it shows competitive performance to other models over 0.695 AUC. M2-2 is the NNB model trained with the best hyperparameters found in Section 4.2. The models M2-3 and M2-4 are the replicas of the M2-2 with different random seeds since M2-2 is the first model that achieves AUC over 0.7.

[1] https://huggingface.co/transformers/pretrained_models.html

Table 1: The hyperparameter description and performance on DEV dataset of each final model for ensemble. NB* represents NRMS+BERT model which is trained only with title embeddings.

Abbreviation	M1-1	M1-2	M2-1	M2-2	M2-3	M2-4
Model	NNG	NNG	NB*	NNB	NNB	NNB
Word embedding size	300	300	300	256	256	256
Learning rate	0.0001	0.0001	0.0001	0.0002	0.0002	0.0002
Dropout rate	0.2	0.2	0.2	0.16	0.16	0.16
Depth of multi-head self-attention layers	1	2	1	2	2	2
Random seed	0	0	42	42	18	1
AUC	0.6914	0.6982	0.6959	0.7067	0.6973	0.7028
MRR	0.3323	0.3358	0.3388	0.3466	0.3411	0.3446
NDCG@5	0.3694	0.3976	0.3733	0.3854	0.3765	0.3818
NDCG@10	0.4345	0.4393	0.4385	0.4487	0.4414	0.4459

Table 2: Results of the best four ensemble models submitted to the leaderboard.

Abb.	M1-1	M1-2	M2-1	M2-2	M2-3	M2-4	type	AUC (DEV)	AUC (TEST (10%))
E1	X	O	O	O	X	O	z-score	0.7109	0.7087
E2	**O**	**O**	**O**	**O**	**X**	**O**	**z-score**	**0.7107**	**0.7113**
E3	X	X	O	O	X	O	s-score	0.7097	0.7075
E4	O	O	X	O	X	O	rank	0.7091	0.7038

When generating ensembles, we use the following scores.

- **Rank.** We compute the average of ranks predicted by each model.
- **S-score.** We compute the average of softmax value of the raw score of each model.
- **Z-score.** We compute the average of z-normalized value of the raw score of each model.

We test all 189 combinations of ensemble models and compare the performance on the DEV dataset. Then, we finally select the four final ensemble models E1 to E4. The performance of the final ensemble models on both DEV and TEST (10%) dataset is summarized in Table 2. Finally, we select E2 as our best model and submit it to the system. The performance of E2 model on each dataset is summarized in Table 3. The ensemble model turns out to generalize very well, since it achieved a better performance on TEST dataset than DEV dataset.

5 Conclusion

In this paper, we summarize our work in MIND News Recommendation Competition. We utilize BERT and Glove to extract word embeddings from titles, categories, subcategories, and abstracts. Then, we use a two layered multi-head self-attention technique to extract more sophisticated embeddings of titles and abstracts. We integrate embeddings of several content sources using attentive multi-view learning. Our final submitted model is an ensemble of base models, and we achieve AUC: 0.7114, MRR: 0.3568, NDCG@5: 0.3916, NDCG@10: 0.4485 on the full TEST dataset.

Acknowledgement

We thank Hyunsik Jeon, Seungcheol Park, and Yuna Bae for helpful discussions.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[2] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

Table 3: Final result of E2 model.

	AUC	MRR	NDCG@5	NDCG@10
DEV	0.7107	0.3477	0.3867	0.4514
TEST (10%)	0.7113	0.3572	0.3920	0.4487
TEST	0.7114	0.3568	0.3916	0.4485

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[5] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In *IJCAI*, 2019.

[6] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In *EMNLP-IJCNLP*, 2019.