UNBERT: User-News Matching BERT for News Recommendation

Qi Zhang, **Jingjie Li***, **Qinglin Jia**, **Chuyuan Wang**, **Jieming Zhu**, **Zhaowei Wang** and **Xiuqiang He**

Huawei Noah's Ark Lab

{zhangqi193, lijingjie1, jiaqinglin2, wangchuyuan, jamie.zhu, wangzhaowei3, hexiuqiang1}@huawei.com

Abstract

Nowadays, news recommendation has become a popular channel for users to access news of their interests. How to represent rich textual contents of news and precisely match users' interests and candidate news lies in the core of news recommendation. However, existing recommendation methods merely learn textual representations from in-domain news data, which limits their generalization ability to new news that are common in cold-start scenarios. Meanwhile, many of these methods represent each user by aggregating the historically browsed news into a single vector and then compute the matching score with the candidate news vector, which may lose the low-level matching signals. In this paper, we explore the use of the successful BERT pre-training technique in NLP for news recommendation and propose a BERT-based user-news matching model, called UNBERT. In contrast to existing research, our UNBERT model not only leverages the pre-trained model with rich language knowledge to enhance textual representation, but also captures multi-grained user-news matching signals at both word-level and news-level. Extensive experiments on the Microsoft News Dataset (MIND) demonstrate that our approach consistently outperforms the state-of-the-art methods.

1 Introduction

Online news platforms such as Google News[1] and MSN News[2] have become a prevalent way for users to access news information [Das *et al.*, 2007; Lavie *et al.*, 2010]. The large quantity of news articles generated everyday makes it hard for users to hunt for their interested news contents. Therefore, recommendation systems become necessary to provide personalized news recommendations to users according to their preferences [Phelan *et al.*, 2011; IJntema *et al.*, 2010; De Francisci Morales *et al.*, 2012].

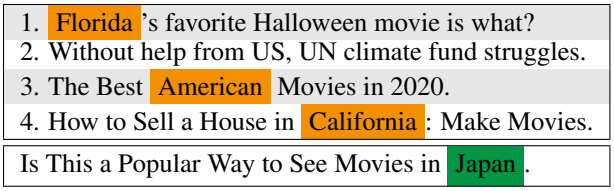

Figure 1: A negative example: several news browsed by a user (upper box) and a candidate news (lower box). Orange bars represent the important signals related with green bar that should be captured.

The quality of news recommendation depends heavily on the understanding of the rich textual contents in news articles. However, traditional recommendation methods often use categorical features (e.g., news ids, news categories) or bag-of-words (tokens or n-grams) to model the news contents. Typical examples include LibFM [Rendle, 2012] and DeepFM [Guo *et al.*, 2017], which are classic recommendation models based on factorization machines. Many recent studies [Wang *et al.*, 2018; An *et al.*, 2019; Wu *et al.*, 2019a; Wu *et al.*, 2019c; Wang *et al.*, 2020] are devoted to the direction of neural news recommendation and propose to learn news representations in an end-to-end fashion. In these models, words are first embedded to low-dimensional embedding vectors, and then they leverage popular network architectures (e.g., CNNs and attention mechanisms) to learn hidden news representations for recommendation. Neural news recommendation models have made surprising performance improvements, but they still suffer from the cold-start problem. As we know, the cold-start problem is severe in new recommendation due to the rapid updates and short timeliness of news articles. To alleviate this issue, several recent studies [Wu *et al.*, 2019c; Wang *et al.*, 2020] propose the use of pre-trained word embeddings (e.g., Word2Vec [Mikolov *et al.*, 2013] and Glove [Pennington *et al.*, 2014]) to initialize the embedding layers of their models, which has shown non-negligible improvements. However, these pre-trained word embeddings are mostly context-independent and their effectiveness may be further weaken during training with a randomly initialized downstream model. Instead, in this paper, we take one step further to apply a pre-trained BERT model [Devlin *et al.*, 2018] in news recommendation. BERT is one of the most successful pre-trained language models in natural language processing (NLP), and has been widely used

*Corresponding Author

[1]https://news.google.com/

[2]https://www.msn.com/en-us/news

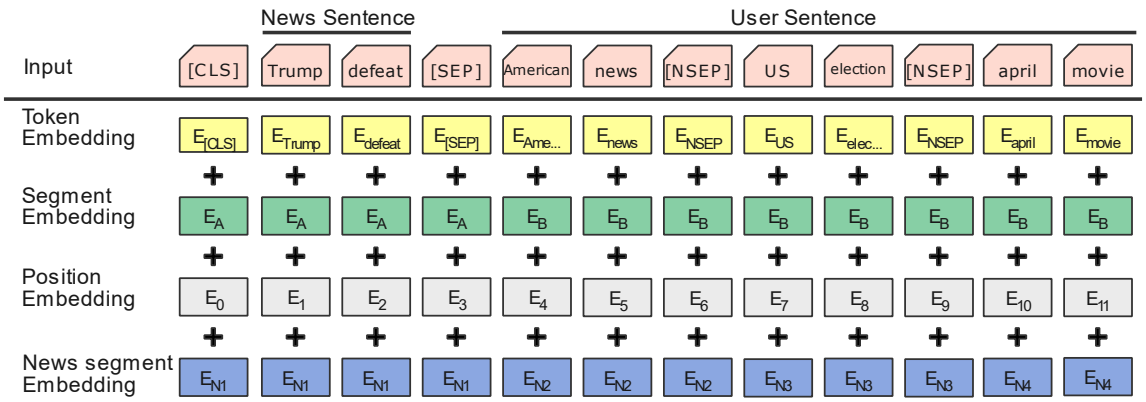

Figure 2: UNBERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, the position embeddings, and the News segmentation embeddings.

for various NLP tasks. We argue that BERT-based models could enhance the deep semantic modeling of news to better mitigate the cold-start problem, since both word embeddings and model parameters are pre-trained on the Web-scale textual data full of general-domain knowledge. Meanwhile, we adopt the pretain-finetune strategy to fine-tune the entire model in our in-domain news data for recommendation.

How to precisely match users' interests and candidate news is another key factor for news recommendation [Wang *et al.*, 2020]. A lot of research has been done toward this goal. For instance, DKN [Wang *et al.*, 2018] and NPA [Wu *et al.*, 2019b] learn user representations based on the similarity between candidate news and previously clicked news. LSTUR [An *et al.*, 2019] models short-term and long-term user interests from the clicked news using GRUs. NAML [Wu *et al.*, 2019a] and NRMS [Wu *et al.*, 2019c] apply the attention networks for learning user representations. After obtaining each user vector and item vector separately, these models then match these two vectors to predict click probability. However, treating each news as a whole vector and encoding users and items separately may ignore some low-level matching signals (e.g., word-level relations) between users' interests and candidate news. As shown in Figure 1, the candidate news has a strong semantic similarity to the 1st and 3rd browsed news because all of them are related to movies at the news level. But at the words level, some words such as "Florida" and "American" in the browsed news may mismatch the "Japan" in candidate news, and the user actually does not click the news. As such, the word-level matching signal which tells a user's location is not well utilized for news recommendation. In this paper, we propose a User-News matching BERT (namely **UNBERT**) model for news recommendation, which not only leverages the powerful pre-trained BERT model but also captures user-news matching signals at both news-level and word-level. The contributions of our work are summarized as follows:

- To the best of our knowledge, UNBERT is the first work to introduce the pre-trained BERT to capture user-news matching signals for news recommendation that takes full advantage of the out-domain knowledge.
- UNBERT proposes the idea of representing user by raw text of the browsed news directly, and learns user-news matching representation at both word-level and news-level to capture multi-grained user-news matching signals through two matching modules.
- Extensive experiments on the real-world dataset show that our approach can effectively improve the performance of news recommendation.

2 Related Work

Traditional methods utilize manual feature engineering to represent news and user for matching [Son *et al.*, 2013; Bansal *et al.*, 2015]. For example, CCTM [Bansal *et al.*, 2015] leverages article and comment content through topic modeling and the co-commenting pattern of users through collaborative filtering to build news and user representations. In recent years, several deep learning methods are proposed for news and user representations, and achieve better performance than traditional methods. For example, DKN [Wang *et al.*, 2018] and NPA [Wu *et al.*, 2019b] use CNN and personalized attention mechanism respectively to learn news representations, LSTUR [An *et al.*, 2019], NAML [Wu *et al.*, 2019a] and NRMS [Wu *et al.*, 2019c] apply the attention networks in news representation, and these methods then learn the user representation by aggregating user's browsed news. In addition, matching-based method FIM [Wang *et al.*, 2020] performs fine-grained matching between segment pairs of each browsed news and the candidate news at each semantic level via stacked dilated convolutions.

Different from these methods, our method introduces a capable pre-trained BERT model to alleviate the cold-start problem by introducing out-domain knowledge, learns the matching representations at both the word-level and news-level via self-attention to capture the multi-grained user-news matching signals.

3 Our Approach

In this section, we first formulate the problem of news recommendation, and then elaborate on our UNBERT including the input and output as well as the model architecture.

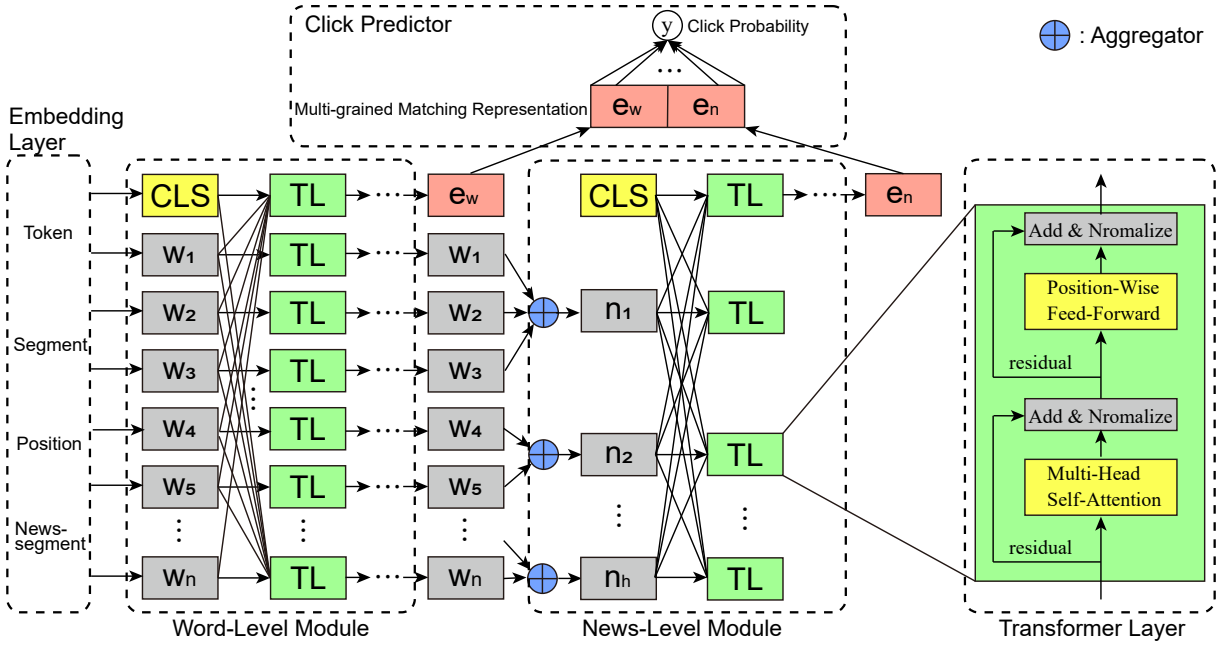

Figure 3: The overall architecture of our UNBERT approach.

3.1 Problem Statement

Given a user u and a set of candidate news $V_u = \{v_1, v_2, \ldots, v_{|V_u|}\}$, where $u \in \mathcal{U}, v \in \mathcal{V}$, $\mathcal{U}$ and $\mathcal{V}$ denote the set of user and news. Formally, our model aims to generate scores for all candidate news by predicting their click probability of u. The score of the i-th candidate news v_i for user u is denoted by $\hat{y}_i = f(u, v_i)$.

3.2 Input and Output

For a given user u and a candidate news v, we denote clicked news of user u as $[n_1^u, n_2^u, \ldots, n_{|n_u|}^u]$, where $n_j^u (j = 1, \ldots, |n_u|)$ is the j-th clicked news by user u, and n_v represents the candidate news v. In this study, each news is represented by the title[3] which is composed of a sequence of words, i.e., $n_u = [w_1, w_2, \ldots]$, and w_i represent the i-th word.

For news representation, its sequence of words is used as "News Sentence" as shown in Figure 2. And for user representation, we simply concatenate the words of user's clicked news into a whole sequence as "User Sentence", where a segment token `[NSEP]` is added at begin of each news as the separated signal. As shown in Figure 2, we add the special classification token `[CLS]` followed by these two sentences and a special token `[SEP]` to separate them. Finally, the "Input" token sequence to UNBERT is a combination of "News Sentence" and "User Sentence" with a set of special symbols.

The output of UNBERT is a matching score that stands for the probability that the user u will click the candidate news v.

[3]In this paper, we only adopt news titles as input, since title is the decisive factor affecting users' choice of reading. But note that our approach can be easily generalized to any sort of news-related texts like the abstract.

3.3 Model Architecture

In this section, we present the architecture details of our UNBERT approach as shown in Figure 3, including *Embedding Layer*, *Word-Level Module*, *News-Level Module* and *Click Predictor*.

Embedding Layer

For a given input token, the corresponding embeddings including token, segment, position and news segment are generated and the input representation is constructed by summing these embeddings in our study. A visualization of this construction can be seen in Figure 2.

The token, segment and position embeddings are pre-trained using masked LM by masking 15% of all word-piece tokens in each sequence at random and predicting the masked words. The news segment embedding is randomly initialized and further updated in our fine-tuning task. The final input token representation E_t in UNBERT is constructed by summing its corresponding token, segment, position and news segment embedding[4]:

$$E_t = E_{token} + E_{seg} + E_{pos} + E_{nseg} \quad (1)$$

Word-Level Module

As illustrated in Figure 3, the Word-Level Module (`WLM`) applies multiple Transformer Layers (`TL`) iteratively to compute the hidden representations at each layer for each word and propagate the matching signal at word-level simultaneously. Each `TL` contains two sub-layers: a Multi-Head Self-Attention sub-layer and a Position-wise Feed-Forward network [Vaswani *et al.*, 2017].

[4]The position embeddings is not used in our approach because the performance will be worse when using it.

	MIND-small			MIND-large		
	Train	Dev	Test	Train	Dev	Test
# User	162,898	47,187	88,898	711,222	255,990	702,005
# News	76,904	53,897	57,856	101,527	72,023	120,961
# Impressions	199,998	50,002	100,000	2,232,748	376,471	2,370,727
# Positive samples	300,357	75,183	153,963	3,383,656	574,845	-
# Negative samples	7,060,083	1,779,492	3,740,561	80,123,718	13,510,712	-

Table 1: Statistics of the datasets.

Multi-Head Self-Attention applies *Scaled Dot-Product Attention* as the attention function to learn the combination weights of the output:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where Q, K and V represents the query, key and value matrix correspondingly, which are projected from E_t matrix with different learned projection matrices as in Eq.(3), and $\frac{1}{\sqrt{d_k}}$ is a scaling factor to avoid extremely small gradients by producing a softer attention distribution. Multi-Head Self-Attention applies h attention functions in parallel to produce the output representations which are concatenated:

$$head_i = Attention\left(E_t W_i^Q, E_t W_i^K, E_t W_i^V\right) \quad (3)$$

$$MultiHead(Q, K, V) = [head_1; \ldots; head_h]\, W^O \quad (4)$$

where W^O, W_i^Q, W_i^K and W_i^V are learnable parameters, i represents i-th head. With the multi-head, the information from different representation subspaces at different positions of word-level is jointly learnable which would be very helpful for capturing the matching signal between different words.

Position-Wise Feed-Forward network is a fully connected feed-forward network (FFN) applied to each position separately and identically in order to endow the model with nonlinearity and interactions between different dimensions. This consists of two linear transformations with a ReLU activation in between:

$$FFN(x) = \text{ReLU}\left(xW_1 + b_1\right)W_2 + b_2 \quad (5)$$

where W_1, W_2 and b_1, b_2 are learnable parameters and shared across all positions.

In addition, the residual connection is introduced into each of the two sub-layers, as shown in Figure 3, in order to make the model deeper and reduce training difficulty. Moreover, the layer normalization is applied to each sub-layer to normalize the inputs over all the hidden units in the same layer for stabilizing and accelerating the network training.

News-Level Module

We aggregate the word's hidden representation of each news from `WLM` to the news representation, and then implement the other multiple Transformer Layers to capture the news-level matching signal in News-Level Module (`NLM`).

Denote the hidden representation for the i-th word obtained from `WLM` as w_i, n_j is the j-th news representation aggregated from its sequence of words S_j where $i \in S_j$. Here, we obtain the news representation n_j from the word representation w_i using the following three types of aggregators:

- *NSEP Aggregator* directly uses the embedding of the special token `[NSEP]` for news representation, due to the final hidden state corresponding to this token is designed to represent the token sequence [Devlin *et al.*, 2018]:

$$n_j = w_i \quad \text{where} \quad i = [\texttt{NSEP}]_j \quad (6)$$

- *Mean Aggregator* averages the words' embedding directly to form the news embedding:

$$n_j = \sum_{i \in S_j} w_i \;/\; |S_j| \quad (7)$$

- *Attention Aggregator* uses a light-weight attention network [Bakhtin *et al.*, 2018] to learn the combination weights of the word embedding matrix w. It applies a fully connected neural network with W_h and b_h, *tanh* as hidden layer activation function, and then another fully connected neural network with W_o and b_o to learn combination weights f defined in Eq.(8) and finally computes the linear combination of words embedding in Eq.(9).

$$f = \tanh\left(wW_h + b_h\right)W_o + b_o \quad (8)$$

$$n_j = \sum_{i \in S_j} f_i w_i \;/\; \sum_{i \in S_j} f_i \quad (9)$$

We then stack the same transformer layers as `WLM` to gather the information propagated from these news embeddings, and explore the news-level matching representation in `NLM`,

Click Predictor

The click predictor module is used to predict the probability of a user clicking a candidate news. Denote the word-level matching representation e_w and the news-level matching representation e_n as shown in Figure 3, we concatenate these two representation vectors as multi-grained matching representation before applying a full connection layer:

$$y = \text{softmax}([e_w; e_n]\, W^c + b^c) \quad (10)$$

where e_w and e_n refer to the word-level and new-level matching representation correspondingly, W^c and b^c are the learnable parameters.

4 Experiments

4.1 Datasets

We conduct experiments on a real-world news recommendation dataset MIND[5] [Wu *et al.*, 2020] collected from MSN

[5]https://msnews.github.io

Method	MIND-small				MIND-large			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
LibFM	0.5974	0.2633	0.2795	0.3429	0.6185	0.2945	0.3145	0.3713
DeepFM	0.5989	0.2621	0.2774	0.3406	0.6187	0.2930	0.3135	0.3705
DKN	0.6175	0.2705	0.2890	0.3538	0.6407	0.3042	0.3292	0.3866
NPA	0.6321	0.2911	0.3170	0.3781	0.6592	0.3207	0.3472	0.4037
NAML	0.6550	0.3039	0.3308	0.3931	0.6646	0.3275	0.3566	0.4140
LSTUR	0.6438	0.2946	0.3189	0.3817	0.6708	0.3236	0.3515	0.4093
NRMS	0.6483	0.3001	0.3252	0.3892	0.6766	0.3325	0.3628	0.4198
FIM	0.6502	0.3026	0.3291	0.3910	0.6787	0.3346	0.3653	0.4221
UNBERT	**0.6762**	**0.3172**	**0.3475**	**0.4102**	**0.7068**	**0.3568**	**0.3913**	**0.4478**
%Improv.	2.12	1.33	1.67	1.71	2.81	2.22	2.60	2.57
UNBERT-en$^{\triangle}$	-	-	-	-	0.7183	0.3659	0.4020	0.4581

Boldface indicates the best results (the higher, the better), while the second best is underlined. UNBERT-en$^{\triangle}$ represents the ensemble score based on UNBERT which is at the top of https://msnews.github.io/#leaderboard.

Table 2: The overall performance of different methods on MIND.

News[6] logs. There are two versions of the MIND datasets named MIND-large and MIND-small, where the MIND-small is a small version of the MIND-large by randomly sampling the daily behavior logs with equal probability from the MIND-large. The detailed statistics of the datasets are shown in Table 1.

4.2 Evaluation Metrics

Several standard evaluation metrics in the recommendation field are used including: AUC, MRR and nDCG@K with $K = 5, 10$. The performance is the average of these metrics on all impression logs. Since test set labels of MIND-large are not provided, the test performance is obtained through submission on the MIND News Recommendation Competition[7].

4.3 Models and Training Details

In our experiments, MIND-small dataset is used to determine the parameter settings, then we train and evaluate on both small and large dataset. The *bert-base-uncased* is used as the pre-trained model to initialize the word-level module. We apply negative sampling with ratio 4 in consideration of being consistent with other baselines as well as the training efficiency, Adam [Kingma and Ba, 2014] is used for model optimization. The batch size is set to 128, the learning rate is set to $2e^{-5}$, and 2 epochs are trained. All the hyper-parameters are tuned on the validation set.

4.4 Performance Evaluation

The performance of our approach is evaluated by comparing with the following methods: (1) **LibFM** [Rendle, 2012], which extracts TF-IDF [Beel *et al.*, 2016] features from users' browsed news and candidate news, and concatenates them as the input; (2) **DeepFM** [Guo *et al.*, 2017]: a deep factorization machine with the same features as LibFM; (3) **DKN** [Wang *et al.*, 2018], a deep news recommendation method based on knowledge-aware CNN; (4) **NPA** [Wu *et al.*, 2019b], a neural news recommendation method with personalized attention mechanism; (5) **NAML** [Wu *et al.*, 2019a], a neural news recommendation approach with attentive multi-view learning; (6) **LSTUR** [An *et al.*, 2019], a neural news recommendation method using GRU to learn user representations; (7) **NRMS** [Wu *et al.*, 2019c], a neural news recommendation method using multi-head self-attention to learn user and news representations; (8) **FIM** [Wang *et al.*, 2020], a fine-grained interest matching method for neural news recommendation; (9) **UNBERT**, our approach. For fair comparison, only title of news is adopted for all methods, in particular, we only use the TF-IDF features extracted from all news titles for LibFM and DeepFM. The results of these methods on two datasets are summarized in Table 2.

We have several observations from Table 2. First, the methods which learn word representations in an end-to-end manner such as DKN, NPA, NAML, LSTUR and NRMS outperform the neural recommendation methods that craft the features manually such as LibFM and DeepFM, which tells that these end-to-end methods are more suitable for news representation learning than the methods based on manual feature engineering.

Second, the methods of learning news representation using the attention mechanism (e.g., NPA, NAML, LSTUR and NRMS) outperform DKN. This is probably because the attention mechanism can model the interaction between the words, and learn the news representations more accurately by capturing relative importance of the interaction. Particularly, NRMS outperforms other attention-based methods because it adopts the useful multi-head self-attention to capture the relatedness between users' browsed news and the candidate news in news-level.

Third, FIM outperforms other methods because its pair-

[6]https://www.msn.com/en-us/news

[7]https://competitions.codalab.org/competitions/24122#participate

	AUC	MRR	nDCG@5	nDCG@10
$\text{UNBERT}_{\text{word}}$	0.6733	0.3138	0.3429	0.4058
$\text{UNBERT}_{\text{news}}$	0.6735	0.3147	0.3455	0.4079
UNBERT	0.6762	0.3172	0.3475	0.4102

Table 3: Ablation study of UNBERT single word-level signal and news-level signal.

	AUC	MRR	nDCG@5	nDCG@10
$\text{UNBERT}_{\text{nseg}}$	0.6746	0.3162	0.3460	0.4088
$\text{UNBERT}_{\text{mean}}$	0.6758	0.3151	0.3453	0.4079
$\text{UNBERT}_{\text{att}}$	0.6762	0.3172	0.3475	0.4102

Table 4: The ranking performance with different aggregator.

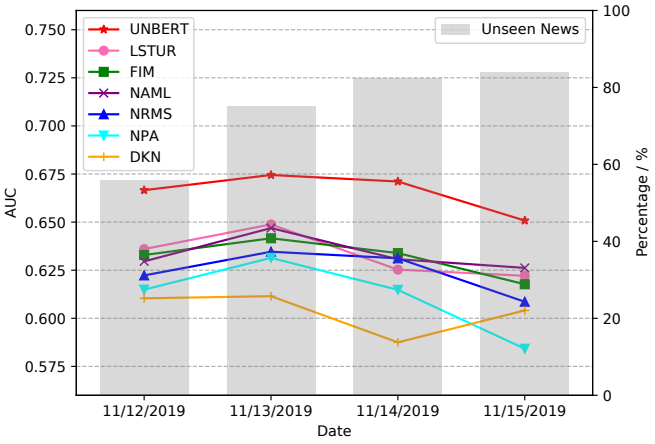

Figure 4: Performance trends of different methods on different days.

wise multi-level matching architecture can detect fine-grained matching signals not just news-level matching information.

Finally, our approach can consistently outperform other baseline methods in terms of all metrics. The significant improvement indicates that the leverage of out-domain data through the pre-trained model can introduce rich language knowledge and enhance the textual representation. Moreover, this also validates advantage of the multi-grained user-news matching signals at both word-level and news-level via self-attention to predict the probability of a user clicking a candidate news.

4.5 Ablation Study of WLM and NLM

In order to study the effectiveness of word-level module and news-level module which aims to capture the multi-grained user-news matching signal, we test our model by disabling one part to evaluate another, yielding UNBERT with word-level signal ($\text{UNBERT}_{\text{word}}$), UNBERT with news-level signal ($\text{UNBERT}_{\text{news}}$) and the complete UNBERT, as shown in Table 3. The experiment is conducted on the MIND-small due to the submission limit on the MIND-large.

Firstly, we observe that $\text{UNBERT}_{\text{word}}$ and $\text{UNBERT}_{\text{news}}$ achieve a pretty good performance close to the full version of UNBERT, which confirms the effectiveness of these two level matching signals. Secondly, $\text{UNBERT}_{\text{news}}$ outperforms $\text{UNBERT}_{\text{word}}$, which proves that the word-level is insufficient for its weakness on capturing news structure. Finally, the full version of UNBERT performs best, which tells that the multi-grained matching signal is necessary for news recommendation.

4.6 Impact of Aggregator Type

We also study the impact of different aggregator type in NLM of UNBERT, on the final ranking performance in Table 4 .

We observe that *Attention Aggregator* achieves the best performance, *Mean Aggregator* follows and *NSEG Aggregator* is the worst. Since *NSEG Aggregator* uses the embedding of a special token that can represent news from word embeddings to some extent, while *Mean Aggregator* aggregates word representation to news representation by mean reducing explicitly, which leads to a better performance of *Mean Aggregator*. *Attention Aggregator* applies attention mechanism and further improves the performance, which indicates that the relatedness between the words is a vital factor for news representation.

4.7 Effectiveness on Cold Start

This experiment studies the performance of different news recommendation methods on unseen news. Recall that the large quantity of unseen news are generated everyday, and effective method should be able to alleviate this problem. Therefore, we evaluate UNBERT on the test set of different days, and compare it with other models. We divide the MIND-small dataset into 7 groups by day, train and validate models on the first 3 days, and then test models on each day of the remaining 4 days.

As shown in figure 4, our method consistently outperforms other methods in a large margin on different days, which proves the effectiveness of UNBERT. In addition, with the cold news growing over time, UNBERT performs stably, while the performances of other methods drop significantly, especially from 11/13/2019 to 11/14/2019. This illustrate the advantage of out-domain knowledge on alleviating the cold-start problem. The abnormal that the overall performance increases while the percentage of unseen news increases from 11/12/2019 to 11/13/2019 is due to the token distribution has not changed much between these two days.

5 Conclusion

In this paper, we propose a novel method named User-News matching BERT (UNBERT) for news recommendation. Our method introduces a pre-trained model on the out-domain data to alleviate the cold-start problem, and captures multi-grained user-news matching signals at both word-level and news-level through WLM and NLM. The experimental results show that UNBERT outperforms the state-of-the-art methods on the real-world dataset in terms of AUC, MRR, nDCG@5 and nDCG@10, and UNBERT ranks the first in the leaderboard of MIND competition.

References

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

[Bakhtin *et al.*, 2018] Anton Bakhtin, Arthur Szlam, Marc'Aurelio Ranzato, and Edouard Grave. Lightweight adaptive mixture of neural and n-gram language models. *arXiv preprint arXiv:1804.07705*, 2018.

[Bansal *et al.*, 2015] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 195–202, 2015.

[Beel *et al.*, 2016] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

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

[De Francisci Morales *et al.*, 2012] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 153–162, 2012.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[IJntema *et al.*, 2010] Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 1–6, 2010.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lavie *et al.*, 2010] Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. User attitudes towards news content personalization. *International journal of human-computer studies*, 68(8):483–495, 2010.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[Phelan *et al.*, 2011] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. In *European Conference on Information Retrieval*, pages 448–459. Springer, 2011.

[Rendle, 2012] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22, 2012.

[Son *et al.*, 2013] Jeong-Woo Son, A-Yeong Kim, and Seong-Bae Park. A location-based news article recommendation with explicit localized semantic analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 293–302, 2013.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2018] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1835–1844, 2018.

[Wang *et al.*, 2020] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 836–845, 2020.

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

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

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

[Wu *et al.*, 2020] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, 2020.