Transformer-based DKN for News Recommendation

Hanwei Xia*, Inwhee Joe**
*Dept. of Computer Science, Hanyang University
**Dept. of Computer Science, Hanyang University

ABSTRACT

In recent years, deep learning has been widely used in news recommendation systems. In the previous personalized news recommendation, a large number of CF-based methods, content-based or hybrid methods have been proposed. But most of the works are only modeling the user's interaction history, ignoring the hidden meaning of the user's continuous behaviors. In this paper, we propose to adopt the powerful Transformer model in order to understand the hidden meaning of the user's continuous behaviors in news recommendations. The experimental results prove the superiority of the transformer, and the AUC has been significantly improved as compared to the original model.

1. INTRODUCTION

In many real-world applications, the current interests of users are intrinsically dynamic and evolving, and are affected by their historical behaviors. For example, after reading the news of large-scale dissemination of covid-19, the user will want to know the symptoms and preventive measures of covid-19, but he/she will not read such news under normal circumstances.

In the era of deep learning, Google's WDL [1] embeds a large number of raw features as vectors into low-dimensional spaces and then fed into fully connected layers that are multi layer perceptron (MLP) to predict whether a user will click on an item. But they just concatenate all features without learning the hidden information between the user's continuous behaviors. Alibaba's DIN [2] is improved based on WDL, and they proposed to use attention mechanism to compare the similarities between the target item and the previously clicked items of a user. Deep Knowledge-Aware Network (DKN) [3] also uses DNN as an attention network to distinguish the impact of different news in the user's click history on target news. None of them takes into account the hidden meaning of the user's continuous behaviors.

To solve the above problem, we try to incorporate continuous information of the user's behavior sequence into the DKN. Inspired by the great success of the Transformer in natural language processing (NLP) [4], we treat news in a user's behavior sequence as words in a sentence, and then use the self-attention mechanism to learn a better representation for each news in a user's behavior sequence, and then feed them into MLPs to predict the probability that the user will click on target news. The advantage of the Transformer is that it can use the self-attention mechanism to better capture the relevance among words in sentences. In other words, the Transformer can extract the "relevance" among news in a user's behavior sequences. We used the real data set attached to the DKN for experiments. The results show that the Transformer has made significant progress in deep learning-based recommendation methods, and AUC has improved significantly.

2. RELATED WORK

Personalized News Recommendation. In personalized news recommendation, CF-based methods [5] often have the cold-start problem because news items are frequently replaced. Therefore, many content-based or hybrid methods have been proposed [6, 7]. Recently, researchers have also tried to combine topic models [8] and recurrent neural networks [9] into news recommendations. The major difference between prior work and ours is that we use the Transformer to extract the "dependency" among news in a user's behavior sequences, in order to better understand the hidden meaning of user behavior sequences.

Attention Mechanism. The Transformer lets people see the power of attention mechanism in machine translation [4] and text classification. Recently, researchers try to employ the attention mechanism to improve recommendation performances and interpretability [3, 10]. For example, DKN [3] uses DNN as an attention network to distinguish the impact of different news in the user's click history on target news. DKN treats attention mechanism as an additional component to the original models. In contrast, the Transformer is built solely on multi-head self-attention and achieves great success in text sequence modeling. So Transformer can better extract the "relevance" among news in a user's behavior sequences.

Sequential Recommendation. Recently, RNN, Gated Recurrent Unit (GRU) [11] and Long Short-Term Memory (LSTM) [12] are widely used in modeling user behavior sequences [14]. These methods usually encode the user's previous records into vectors with various recurrent architectures and loss functions. Other than RNN, deep

learning models are also introduced for sequential recommendation. For example, BST [13] also uses the Transformer model. The major difference between BST and ours is that We have different architectures in the feature embedding layer.

3. METHODOLOGY

Fig 1 is the overview architecture of this paper. We improved on the basis of DKN [3], using KCNN to convert news headlines into news embedding, and embed positional features as low-dimensional vectors. After concatenating news embedding and positional embedding, use the Transformer layer to learn the deeper representation of each news in the sequence. We connect the embedding of Other Features with the output of the Transformer layer, then use three fully connected layers to learn the interactions of the mixed features, and then use the sigmoid function to generate the final output.

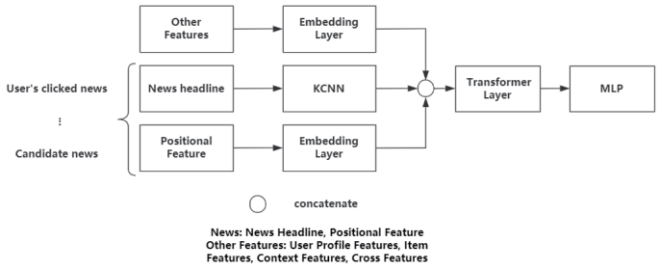


Fig.1: overview architecture

3.1 Knowledge Distillation and Knowledge-aware CNN

First, DKN [3] split the news headline into a set of words, and then link the words in the headline with the entities of the knowledge base. If the entity corresponding to the word can be found, then all adjacent entities within one hop from the linked entity are found, and these adjacent entities are called context entities. The word2vec model can be used to obtain the word embedding, and the knowledge graph embedding model can be used to obtain the knowledge entity embedding. Context embedding is to average multiple entity embeddings.

KCNN. DKN maps entity embedding and context embedding to the same vector space through a non-linear transformation. Then use word embedding, entity embedding, and context embedding as the multi-channel CNN input, and obtain news embedding of the news headline through the convolution operation.

3.2 Embedding layer

The role of the embedding layer is to embed all the input features into a fixed-size low-dimensional vector. We concatenate user profile features, item features, context features, and the combination of different features and embed them into low-dimensional vectors. Then we concatenate the matrix embedding of these features with the matrix embedding output by Transformer as the input of the MLP layer.

Positional embedding. Position feature is equivalent to the positional encoding in [4], which is the position information of the news in the users' behavior sequence. The purpose is to introduce timing information to the users' behavior sequence. We use $pos(v_i) = t(v_t) - t(v_i)$ to calculate the position value of news v_i, and then project it into a low-dimensional vector after discretization. Where $t(v_t)$ represents the recommending time and $t(v_i)$ the timestamp when user click news v_i. Then we concatenate news embedding and positional embedding as the input of the Transformer.

3.3 Transformer layer

Fig 2 is the architecture of the Transformer Encoder lay er.

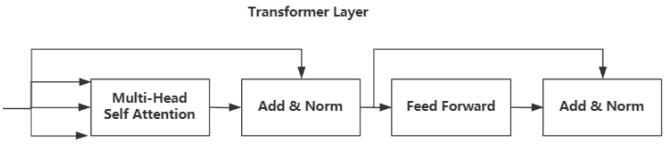


Fig.2: Transformer Encoder Layer

Self-attention layer. Below is the scaled dot-product attention [4]:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{\mathrm{d}}}\right)\mathbf{V}, \quad (1)$$

where **Q** is the queries, **K** is the keys, and **V** is the values. The self-attention operation takes the embedding of items as input, transforms them into three matrices by linear projection, and enters them into an attention layer. Following [4], we use the multi-head attention:

$$\mathbf{S} = \text{MH}(\mathbf{E}) = \text{Concat}(head_1, head_2, \dots, head_h)\mathrm{W}^{\mathrm{H}}, \quad (2)$$

$$head_i = \text{Attention}(\mathbf{E}\mathbf{W}^{\mathbf{Q}}, \mathbf{E}\mathbf{W}^{\mathbf{K}}, \mathbf{E}\mathbf{W}^{\mathbf{V}}) \ , \quad (3)$$

where the projection matrices $\mathbf{W}^{\mathrm{Q}}, \mathbf{W}^{\mathrm{K}}, \mathbf{W}^{\mathrm{V}} \in \mathbb{R}^{d\times d}$, and E is the embedding matrices of all news, and h is the number of heads.

Point-wise Feed-Forward Networks. Following [4], We define point-wise Feed-Forward Networks (FFN) as follows:

$$\mathbf{F} = FFN(\mathbf{S}). \quad (4)$$

We use dropout and LeakyReLU in self-attention and FFN to avoid overfitting and learn hidden features hierarchically. Below is the output of the self-attention and FFN layers:

$$\mathbf{S}' = \text{LayerNorm}(\mathbf{S} + \text{Dropout}(\text{MH}(\mathbf{S}))), \quad (5)$$

$$\mathbf{F} = \text{LayerNorm}(\mathbf{S}' + \text{Dropout}(\text{LeakyReLU}(\mathbf{S}'\mathbf{W}^{(1)} + b^{(1)})\mathbf{W}^{(2)} + b^{(2)})), \quad (6)$$

Where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, b^{(1)}, b^{(2)}$ are the learnable parameters, and LayerNorm is the standard normalization layer.

Stacking the FNN blocks. It aggregates all the embeddings of previous news to learn the complex relationship hidden in the news sequences after the first self-attention block. Below is the definition of the self-building blocks and the b-th block:

$$\mathbf{S}^b = \text{SA}(\mathrm{F}^{(b-1)}), \quad (7)$$

$$\mathbf{F}^b = FFN(\mathbf{S}^b), \forall i \in 1,2,\dots,n. \quad (8)$$

The output of the Transformer layer is the matrix F, which represents the news embedding in the users' behavior sequence, and the candidate news that contains the characteristics of the users' behavior sequence.

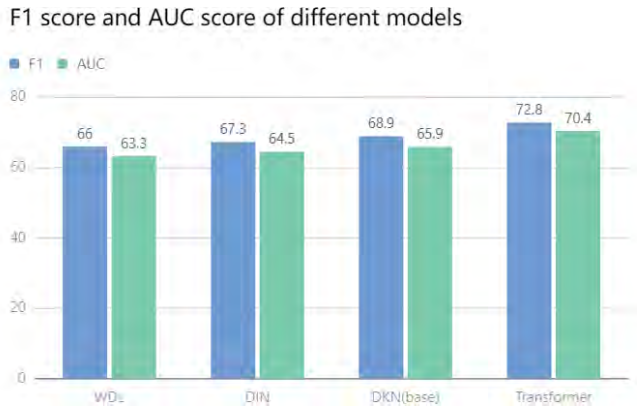
3.4 MLP layer

Due to the good performance of WDL [1] and DIN [2], we connect the embedding of Other Features with the output of

the Transformer layer, then use three fully connected layers to learn the mutual effects among the hybrid features. Finally, we use the sigmoid function to output the probability that the user clicks on the target news because whether users will click on the candidate news is a dichotomy classification problem.

4. PERFORMANCE EVALUATION

The results are shown in Fig 3, from which we can see the superiority of the Transformer model over other models. In specific, the F1 score is improved from 66.0(WDL), 67.3(DIN), and 68.9(DKN) to 72.8(Transformer). The AUC is improved from 63.3(WDL), 64.5(DIN) and 65.9(DKN) to 70.4(Transformer). In contrast, the Average RT of Transformer has not increased a lot, which guarantees the feasibility of using the Transformer model in real news recommendations.



Methods	F1	AUC	Average RT(ms)
WDL	66.0 ±1.2 (-2.9%)	63.3 ± 1.5 (-2.6%)	14
DIN	67. 3±1.3 (-1.6%)	64.5±1.1 (-1.4%)	16
DKN(base)	68.9 ± 1.5	65.9 ± 1.2	19
Transformer	**72.8±1.5(+3.9%)**	**70.4±1.4(+4.5%)**	21

Fig.3: F1 scores and AUCs of different methods

5. CONCLUSION

In this paper, we introduced the use of transformer as an attention mechanism to dynamically calculate the total historical performance of users on the basis of DKN [3]. Experimental data show the superiority of this model in modeling user behavior sequences. If there is a better model of the attention mechanism, you can try to replace it.

REFERENCES

[1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. 2016.

[2] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. 2018. In KDD. 1059–1068.

[3] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. DKN: Deep Knowledge Aware Network for News Recommendation. 2018. In Proceedings of The 2018 Web Conference (WWW 2018). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186175

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. In NIPS. 5998–6008.

[5] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. 2011. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 448–456.

[6] Michal Kompan and Mária Bieliková. Content-Based News Recommendation. 2010. In EC-Web, Vol. 61. Springer, 61–72.

[7] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. 2009. In Proceedings of the third ACM conference on Recommender systems. ACM, 385–388.

[8] Tapio Luostarinen and Oskar Kohonen. Using topic models in contentbased news recommender systems. 2013. In Proceedings of the 19th Nordic Conference of Computational Linguistics. Linköping University Electronic Press, 239–251.

[9] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based News Recommendation for Millions of Users. 2017. In KDD. ACM, 1933–1942.

[10] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural Attentive Session-based Recommendation. 2017. In Proceedings of CIKM. ACM, New York, NY, USA, 1419–1428.

[11] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. 2014. In Proceedings of EMNLP. Association for Computational Linguistics, 1724–1734.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. 1997. Neural Computation 9, 8 (Nov. 1997), 1735–1780.

[13] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, Wenwu Ou. Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. 2019. arXiv: 1905.06874v1 [cs.IR]

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based Recommendations with Recurrent Neural Networks. 2016. In Proceedings of ICLR.