Neural News Recommendation with Multi-Head Self-Attention

(Summary)

Abstract

The core of NRMS is a news encoder and a user encoder. In the news encoder, we use multi-head self-attentions to learn news representations from news titles by modeling the interactions between words. In the user encoder, we learn representations of users from their browsed news and use multi-head self-attention to capture the relatedness between the news. Besides, we apply additive attention to learn more informative news and user representations by selecting important words and news.

Keywords: news encoder, user encoder, multi-head self-attention

1 Introduction

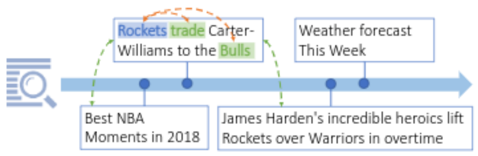

Rockets trade Carter-Williams to the Bulls
Weather forecast This Week
Best NBA Moments in 2018
James Harden's incredible heroics lift Rockets over Warriors in overtime

Figure 1: Several news browsed by an example user Orange and green dashed lines represent the interac tions between words and news respectively.

Our work is motivated by several observations.

- First, the interactions between words in news title are important for understanding the news. For example, in Fig. 1, the word "Rockets" has strong relatedness with "Bulls". Besides, a word may interact with multiple words, e.g., "Rockets" also has semantic interactions with "trade".
- Second, different news articles browsed by the same user may also relatedness. For example, in Fig. 1, the second news is related to the first and the third news.
- Third, different words may have different importance in representing news. In Fig. 1, the word "NBA" is more informative than "2018". Besides, different news articles browsed by the same user also have different importance in representing this user. For example, the first three news articles are more informative than the last one.

2 Our Approach

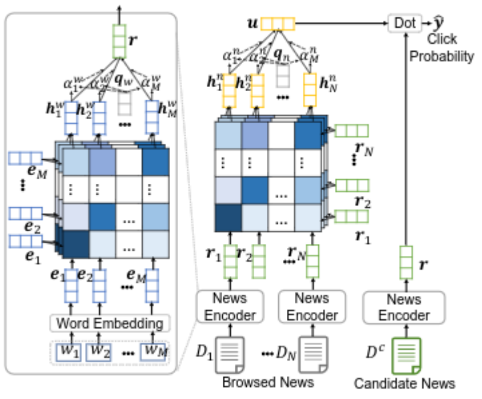


Figure 2: The framework of our *NRMS* approach.

Our NRMS approach for news recommendation is shown in Fig. 2. It contains three modules, i.e., news encoder, user encoder, click predictor.

2.1 News Encoder

The news encoder module is used to learn news representations from news titles. It contains three layers.

- The first one is word embedding, which is used to convert a news title from a sequence of words into a sequence of low-dimensional embedding vectors, $[w_1, w_2, ..., w_M] \rightarrow [e_1, e_2, ..., e_M]$
- The second layer is word-level multi-head self-attention network. The representation of the i_{th} word learned by the k_{th} attention head is computed as:

$$\alpha_{i,j}^k = \frac{exp(e_i^T Q_k^w e_j)}{\sum_{m=1}^{M} exp(e_i^T Q_k^w e_m)}$$

$$h_{i,k}^w = V_k^w (\sum_{j=1}^{M} \alpha_{i,j}^k e_j)$$

  The multi-head representation h_i^w of the i_{th} word is the concatenation of the representations produced by h separate self-attention heads, i.e., $h_i^w = [h_{i,1}^w; h_{i,2}^w; ...; h_{i,h}^w]$.

- The third layer is an additive word attention network. The attention weight α_i^w of the i-th word in a news title is computed as:

$$a_i^w = q_w^T tanh(V_w \times h_i^w + v_w)$$
$$\alpha_i^w = \frac{exp(a_i^w)}{\sum_{j=1}^{M} exp(a_j^w)}$$

  The final representation of a news is formulated as:

$$r = \sum_{i=1}^{M} \alpha_i^w h_i^w$$

2.2 User Encoder

The user encoder module is used to learn the representations of users from their browsed news. It contains two layers.

- The first one is a news-level multi-head self-attention network. The representation of the i_{th} news learned by the k_{th} attention head is formulated as follows:

$$\beta_{i,j}^k = \frac{exp(r_i^T Q_k^n r_j)}{\sum_{m=1}^{M} exp(r_i^T Q_k^n r_m)}$$

$$h_{i,k}^n = V_k^n (\sum_{j=1}^{M} \beta_{i,j}^k r_j)$$

  The multi-head representation of the i_{th} news is the concatenation of the representations output by h separate self-attention heads, i.e., $h_i^n = [h_{i,1}^n, h_{i,2}^n, ..., h_{i,h}^n]$.

- The second layer is an additive news attention network. The attention weight of the i-th news is computed as:

$$a_i^n = q_n^T tanh(V_n \times h_i^n + v_n)$$
$$\alpha_i^n = \frac{exp(a_i^n)}{\sum_{j=1}^{N} exp(a_j^n)}$$

  The final representation of a user is formulated as:

$$u = \sum_{i=1}^{N} \alpha_i^n h_i^n$$

2.3 Click Predictor

$$\hat{y} = u^T r^c$$

2.4 Model Training

We use negative sampling techniques for model training. For each news browsed by a user (positive sample), we randomly sample K news which are shown in the same impression but not clicked by the user (negative samples), denote the click probability score as $\hat{y}^+$ and $[\hat{y}_1^-, \hat{y}_2^-, ..., \hat{y}_K^-]$. These scores are normalized by the softmax function

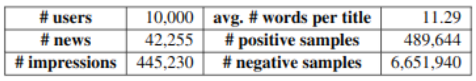
$$p_i = \frac{exp(\hat{y}^+)}{exp(\hat{y}^+) + \sum_{j=1}^{K} exp(\hat{y}_{i,j}^-)}$$

We re-formulate the news click probability prediction problem as a pseudo $(K+1)$-way classification task, and the loss function is the negative log-likelihood of all positive samples S:

$$\mathcal{L} = -\sum_{i \in S} log(p_i)$$

3 Experiments

3.1 Datasets and Experimental Settings

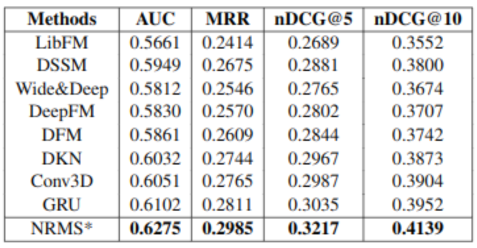


# users	10,000	avg. # words per title	11.29
# news	42,255	# positive samples	489,644
# impressions	445,230	# negative samples	6,651,940

Table 1: Statistics of our dataset.

- Datasets: MIND, collected from MSN News logs in one month (Dec. 13, 2018 to Jan. 12, 2019), the logs in the last week were used for test and the rest were used for training, we randomly sampled 10% of training data for validation.
- Word embeddings: 300-dimensional (GLove), 20 dropout
- Self-attention: 16 heads, output of each head is 16-dimensional
- Additive attention query vectors: 200-dimensional
- K = 4
- Optimizer: Adam
- Batch size = 64
- Machine: with Xeon E5-2620 v4 CPUs and a GTX1080Ti GPU
- Evaluation Metrics: AUC, MRR, nDCG@5 and nDCG@10

3.2 Performance Evaluation



Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	0.5661	0.2414	0.2689	0.3552
DSSM	0.5949	0.2675	0.2881	0.3800
Wide&Deep	0.5812	0.2546	0.2765	0.3674
DeepFM	0.5830	0.2570	0.2802	0.3707
DFM	0.5861	0.2609	0.2844	0.3742
DKN	0.6032	0.2744	0.2967	0.3873
Conv3D	0.6051	0.2765	0.2987	0.3904
GRU	0.6102	0.2811	0.3035	0.3952
NRMS*	**0.6275**	**0.2985**	**0.3217**	**0.4139**

Table 2: The results of different methods. *The improvement is significant at $p < 0.01$.

4 Conclusion and Future Work

Conclusion: The core of our approach is a news encoder and a user encoder. In both encoders we apply multi-head self-attention to learn contextual word and news representations by modeling the interactions between words and news. In addition, we use additive attentions to select important words and news to learn more information news and user representations.

Future Work: considering the positional information of words and news, incorporating multiple kinds of news information (title, abstract, body, ...)

Chu Dinh Duc
24/12/2022