User Modeling with Click Preference and Reading Satisfaction
for News Recommendation

Chuhan Wu[1] , **Fangzhao Wu**[2] , **Tao Qi**[1] and **Yongfeng Huang**[1]

[1]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China
wuchuhan15@gmail.com, fangzwu@microsoft.com,
qit16@mails.tsinghua.edu.cn, yfhuang@tsinghua.edu.cn

Abstract

Modeling user interest is critical for accurate news recommendation. Existing news recommendation methods usually infer user interest from click behaviors on news. However, users may click a news article because attracted by its title shown on the news website homepage, but may not be satisfied with its content after reading. In many cases users close the news page quickly after click. In this paper we propose to model user interest from both click behaviors on news titles and reading behaviors on news content for news recommendation. More specifically, we propose a personalized reading speed metric to measure users' satisfaction with news content. We learn embeddings of users from the news content they have read and their satisfaction with these news to model their interest in news content. In addition, we also learn another user embedding from the news titles they have clicked to model their preference in news titles. We combine both kinds of user embeddings into a unified user representation for news recommendation. We train the user representation model using two supervised learning tasks built from user behaviors, i.e., news title based click prediction and news content based satisfaction prediction, to encourage our model to recommend the news articles which not only are likely to be clicked but also have the content satisfied by the user. Experiments on real-world dataset show our method can effectively boost the performance of user modeling for news recommendation.

1 Introduction

News recommendation is an integral part of many online news services to target user interest and alleviate information overload [Wang *et al.*, 2018]. User interest modeling is critical for personalized news recommendation. Existing news recommendation methods usually model interest of users based on their click behaviors on news platforms. For example, Wang et al. [2018] proposed to learn user representations from clicked news articles based on their similarities to candidate news. Zhu et al. [2019] proposed to learn user representations from clicked news using a combination
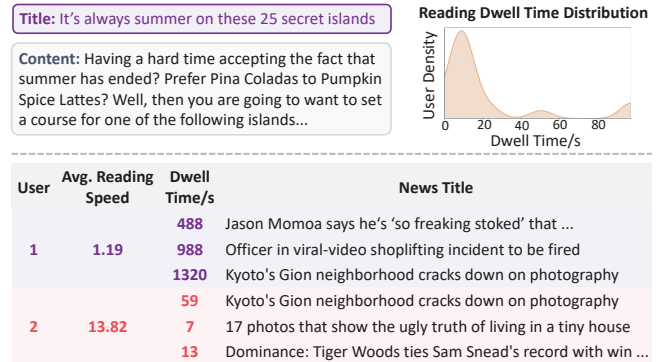

Figure 1: The reading dwell time distribution of an example news and news clicked by two users. The speed unit is words per second.

of CNN and attentive LSTM. Users usually make click decisions based on news titles displayed on news websites. However, since news titles are usually short and the information they condensed is very limited and even noisy (e.g., click-baits), click behaviors based on news titles may not necessarily indicate user interest. In many cases, users are not satisfied with the content of the clicked news and close the news pages quickly. For example, as shown in the top figure of Fig. 1, most users close the page of this news in less than 20 seconds, which means that the content of this news is not attractive for many users who click it. Thus, modeling user interest based on their news click behaviors may not be accurate enough.

Besides the click behaviors, the news reading behaviors can provide important information of whether the users are satisfied with the news content. A natural way to incorporate reading behaviors is using the time users spending on reading the content of news, in other words, the dwell time [Kim *et al.*, 2014]. Long dwell time on a news page usually indicates higher interest and satisfaction on this news content than short dwell time [Lu *et al.*, 2018]. However, news articles usually have huge difference in content length, and longer articles usually take more time to read. Thus, the absolute dwell time may be not optimal for modeling reading satisfaction. A direct improvement of dwell time is reading speed which considers both dwell time and news content length [Lu *et al.*, 2019]. However, as shown in the bottom figure of Fig. 1, different users may have different habits in news reading, and

their average reading speed has huge difference. Thus, reading speed may also be not optimal for satisfaction modeling.

In this paper, we propose a user modeling method for news recommendation which can model user interest from both news click behaviors and news reading behaviors.[1] First, we propose a personalized reading speed metric to measure the reading satisfaction of users with news content, which is computed based on the reading dwell time, news content length and the average historical news reading speed of a user. We learn embeddings of users from the news content they previously read with a content-satisfaction attention network to incorporate their satisfaction into the modeling of their interest on news content. In addition, we learn another user embedding from the news titles clicked by users to model their click preferences in news titles. We synthesize these two kinds of user embeddings into a unified representation for news recommendation via a behavior attention network. We propose to train our model via two supervised tasks based on user behaviors, i.e., news title based click prediction and news content based satisfaction prediction, which can encourage our model to recommend news not only with higher probabilities to be clicked, but also are satisfied by users after reading their content. Extensive experiments on real-world dataset validate that our approach can effectively improve the performance of user modeling for news recommendation.

The main contributions of this paper include:

- We propose an effective method to infer user interest for news recommendation from both click behaviors and satisfaction-aware reading behaviors.

- We propose a personalized reading speed metric to measure users' reading satisfaction with news content.

- We train the user models using both title based click prediction and content based satisfaction prediction tasks.

- We conduct extensive experiments to explore the performance of the proposed method on real-world dataset.

2 Related Work

2.1 User Modeling for News Recommendation

News recommendation is a critical technique for online news websites, and has been extensively studied over years. How to accurately model user interest is a core problem in news recommendation. Many existing methods for news recommendation are based on news clicks to represent users [Liu *et al.*, 2010; Li *et al.*, 2010; Li *et al.*, 2011; Lian *et al.*, 2018]. For example, Liu et al. [2010] proposed to represent users based on the topic categories of their clicked news and interest features generated by a Bayesian model. Li et al. [2010] proposed to use demographics, geographic features and behavior categories on Yahoo! to represent users. Lian et al. [2018] proposed to represent users from their profiles such as demographics and behavior features extracted from their clicked news. However, these methods rely on manual feature engineering to construct user representations, which requires massive expertise.

In recent years, there are several methods based on deep learning to learn user representations [Okura *et al.*, 2017; Wang *et al.*, 2018; Wu *et al.*, 2019b; An *et al.*, 2019; Wu *et al.*, 2019a; Wu *et al.*, 2019d; Zhu *et al.*, 2019; Wu *et al.*, 2019c; Ge *et al.*, 2020]. For example, Okura et al. [2017] proposed to learn representations of users from their historical clicked news via a gated recurrent unit (GRU) network. Wang et al. [2018] proposed to learn user representations from clicked news according to their similarity to candidate news. Wu et al. [2019a] proposed an attentive multi-view learning framework to learn user representations from the unified news representations synthesized from different views of news. Zhu et al. [2019] proposed to use CNNs and an attentive LSTM network to learn user representations from clicked news. However, these methods only consider news click behaviors of users, which may be inaccurate in modeling user interest.

There are only a few news recommendation methods which consider reading satisfaction to fill the gaps between news clicks and user interest [Yi *et al.*, 2014]. For example, Yi et al. [2014] proposed to use dwell time to weight clicked news to form user representations. However, news content length and user reading customs have huge impacts on the reading dwell time, which is not considered by these methods. In addition, the weighting method is also manually designed, which may not be optimal. Different from these methods, we propose a personalized reading speed metric to model user satisfaction, and we propose to consider user satisfaction in both user interest modeling and user model training.

2.2 User Satisfaction Modeling

Modeling user satisfaction in personalized online services is important to better understand user preferences. Some methods model user satisfaction based on the ratings given by users [Zhao *et al.*, 2019]. However, explicit feedback such as user ratings may not be available in many scenarios such as search and web browsing. Thus, many methods model user satisfaction based on implicit feedback such as dwell time of user behaviors [Kim *et al.*, 2014; Liu *et al.*, 2016; Zhou *et al.*, 2018]. For example, Kim et al. [2014] proposed to predict user satisfaction for search engine based on browsing dwell time and the topic of the displayed webpage. Zhou et al. [2018] proposed to jointly predict the probability of user click and dwell time to consider user satisfaction in recommender systems. However, dwell time may be heavily influenced by the characteristics of users and displayed content, which may not be optimal to model user satisfaction. There are also several studies to employ other kinds of implicit feedback to model user satisfaction. For example, factors such as reading speed, reading ratio, scroll interval have been shown correlated with user satisfaction [Lu *et al.*, 2018; Lu *et al.*, 2019]. In addition, the contexts of user behaviors such as the sequential patterns of moves and clicks may also have relatedness with user satisfaction [Mehrotra *et al.*, 2017]. However, different users may also have different using customs, and these factors of different users may also correlate to satisfaction differently. Different from these methods, we propose to use personalized reading speed to model user satisfaction in user interest modeling and model training.
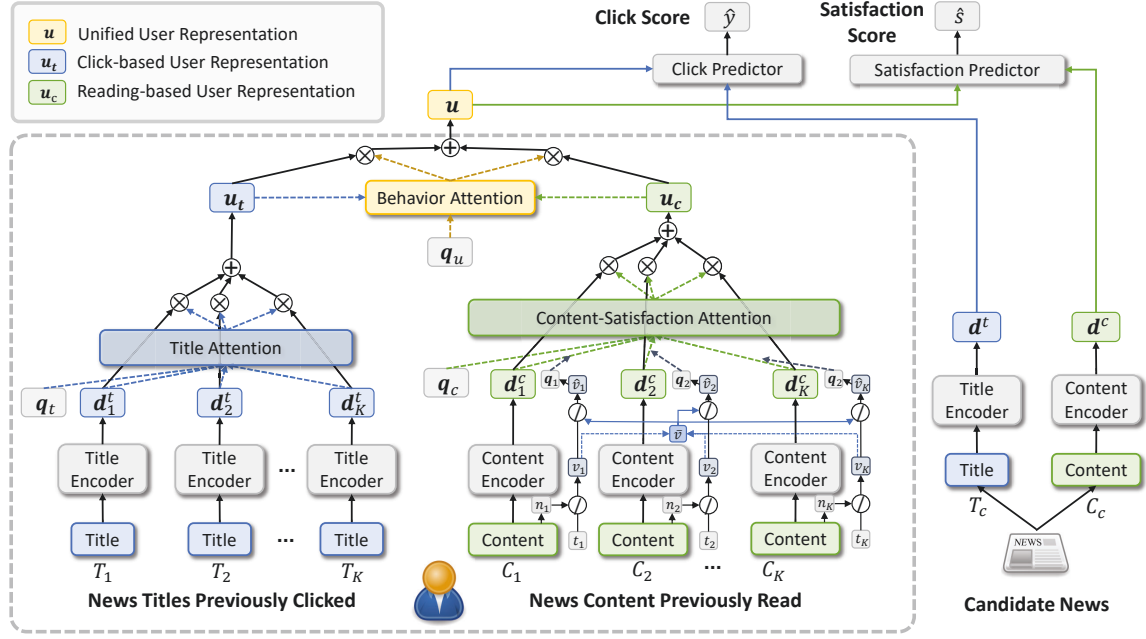
---

[1]The source codes of our CPRS method are available at https://github.com/wuch15/IJCAI2020-CPRS.

Figure 2: The overall framework of our *CPRS* approach.

3 Our Approach

In this section, we introduce our user modeling approach with **c**lick **p**reference and **r**eading **s**atisfaction (CPRS) for news recommendation. The overall framework of *CPRS* is shown in Fig. 2. Its details are introduced in the following sections.

3.1 Satisfaction Modeling

We first introduce how to model reading satisfaction of users with news content. In many existing methods, user satisfaction is modeled by dwell time of behaviors [Kim *et al.*, 2014; Zhou *et al.*, 2018]. However, the length of news content may have impact on the dwell time of news reading, making it inaccurate to directly use dwell time for satisfaction modeling. Thus, it may be more appropriate to use reading speed to combine the information of dwell time and content length. Denote the reading dwell time of the i_{th} news as t_i and the length of its content as n_i. Then the reading speed v_i of this news is computed by $v_i = n_i/t_i$. However, reading speed is still insufficient to model satisfaction accurately, because users are diverse in their reading speeds due to their different reading habits. Thus, we propose to measure user satisfaction via personalized reading speed by using the ratio of reading speeds to the average historical reading speed of this user. The personalized reading speed $\hat{v}_i$ of the i_{th} news is computed by $\hat{v}_i = v_i/(\frac{1}{K}\sum_{j=1}^{K} v_j)$, where K is the number of historical clicked news.

3.2 User Modeling

Next, we introduce how to model user interests by considering both click preference and reading satisfaction. There are three modules in the user modeling part.

The first one is a content-satisfaction attention network, which is used to learn reading-based user representations from news content read by users by considering their satisfaction into interest modeling. Due to the characteristics such as topics and quality of news content, the informativeness of the content of different news in modeling user interest is different. Thus, we use content attentions to learn user representations from previously read news content in a selective manner. Denote the representation vector of the i_{th} news content as $\mathbf{d}_i^c$. Its content attention weight α_i is calculated as follows:

$$\alpha_i = \frac{\exp(\mathbf{q}_c^T \mathbf{d}_i^c)}{\sum_{j=1}^{K} \exp(\mathbf{q}_c^T \mathbf{d}_j^c)}, \tag{1}$$

where $\mathbf{q}_c$ is the content attention query. To incorporate user satisfaction into the modeling of user interest, we propose to use satisfaction attentions to help recognize the news content satisfied by users. To compute satisfaction attention of the i_{th} news content, we first map its real-valued personalized reading speed $\hat{v}_i$ into a discrete variable by $\hat{v}_i' = \lfloor \log_2(\hat{v}_i) \rfloor$. Then, we use the embedding of v_i' to generate the satisfaction attention query $\mathbf{q}_i$ via a linear transformation. Finally, its satisfaction-based attention weight β_i is calculated as:

$$\beta_i = \frac{\exp(\mathbf{q}_i^T \mathbf{d}_i^c)}{\sum_{j=1}^{K} \exp(\mathbf{q}_j^T \mathbf{d}_j^c)}. \tag{2}$$

The final reading-based user representation $\mathbf{u}_r$ is the summation of content representations weighted by their content and satisfaction attention weights, i.e., $\mathbf{u}_c = \sum_{i=1}^{K}(\alpha_i + \beta_i)\mathbf{d}_i^c$.

The second one is a title attention network, which is used to learn clicked-based user representations from clicked news titles to capture users' click interest. News with different titles may have also different informativeness for modeling click preference [Wu *et al.*, 2019c]. Thus, we use an attention network to form click-based user representations by selecting
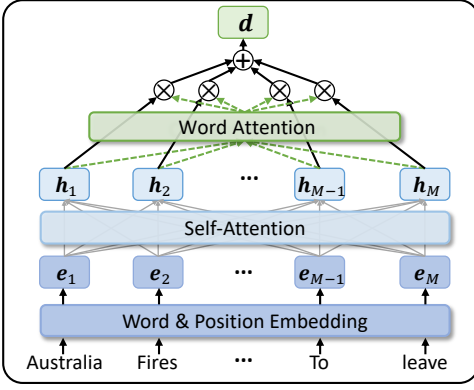
Figure 3: The architecture of the text encoder.

important news titles. Denote the representation of the i_{th} clicked news title as $\mathbf{d}_i^t$. Its attention weight γ_i is computed in a similar way with content attentions. The click-based user representation $\mathbf{u}_t$ is the summation of title representations weighted by their attention weights, i.e., $\mathbf{u}_t = \sum_{i=1}^{K} \gamma_i \mathbf{d}_i^t$.

The third one is a behavior attention network. Usually the click and reading behaviors may have different importance in representing user preference. For example, if a user scans most news very quickly but reads a few news articles carefully, the reading behavior of this user may be more important than click behavior in representing her interest. Thus, we use a behavior attention network to form unified user representations from the click-based and reading-based ones. Denote the attention weights of click behavior and reading behavior as δ_t and δ_c. For instance, δ_t is computed as follows:

$$\delta_t = \frac{\exp(\mathbf{q}_u^T \mathbf{u}_t)}{\exp(\mathbf{q}_u^T \mathbf{u}_t) + \exp(\mathbf{q}_u^T \mathbf{u}_r)}, \quad (3)$$

where $\mathbf{q}_u$ is a behavior attention query. The final user representation $\mathbf{u}$ is computed by $\mathbf{u} = \delta_t \mathbf{u}_t + \delta_c \mathbf{u}_c$.

3.3 Text Encoder

In this section, we briefly introduce how to implement the text encoders (i.e., title encoders and content encoders) in Fig. 2 to learn news title and content representations. Motivated by Wu et al. [2019c], we use a combination of word attention and multi-head self-attention networks to encode news titles and news contents into their representations. In addition, we use position embeddings to keep the order information of words. As shown in Fig. 3, there are three layers in a text encoder.

The first one is a word and position embedding layer. It is used to convert the words in an input text into their embeddings. To capture the order information of words, we use a summation of the semantic embedding and position embedding to represent each word. The output of this layer is a sequence of vector, denoted as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{M-1}, \mathbf{e}_M]$, where M is the number of words.

The second layer is a multi-head self-attention network [Vaswani et al., 2017]. The words in the same text may have inherent relatedness. Thus, we use a multi-head self-attention network to learn representations of words by capturing their interactions with each other. The input of

this layer is the word embedding sequence $\mathbf{E}$, and the output of this layer is a word representation sequence $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{M-1}, \mathbf{h}_M] = MultiHeadAttention(\mathbf{E})$.

The third layer is a content attention network. Different words in the same news may be diverse in their informativeness. Thus, we use a word attention network to learn more informative text representations by selecting important words. The attention weight μ_i of each word is computed in a similar way with title and content attention. The output text representation $\mathbf{d}$ is the summation of word representations weighted by their attention weights, i.e., $\mathbf{d} = \sum_{i=1}^{M} \mu_i \mathbf{h}_i$.

3.4 Click-Satisfaction Prediction

In this section, we introduce the details of label prediction and modeling training, which take both click signals and satisfaction signals into consideration. Denote the representations of the candidate news title and content obtained by text encoders as $\mathbf{d}^t$ and $\mathbf{d}^c$, respectively. Since users' click decisions are mainly based on titles, we predict the probability score $\hat{y}$ of a user clicks the candidate news D_c based on the representation of user and the title of candidate news, i.e., $\hat{y} = \mathbf{w}_t^T[\mathbf{u}; \mathbf{d}^t]$, where $\mathbf{w}_t$ is a parameter vector. Similar to the historical clicked news, modeling user satisfaction with candidate news may also benefit user interest modeling. However, the dwell time of candidate news is not available for test, and it is inappropriate to directly incorporate them as input. Thus, we propose a news content based satisfaction prediction task to predict the satisfaction scores of candidate news, which encourages our model to recommend news not only tend to be clicked but also are satisfied by users in terms of their content. The satisfaction score $\hat{s}$ of a candidate news is predicted by $\hat{s} = \mathbf{w}_s^T[\mathbf{u}; \mathbf{d}^c]$, where $\mathbf{w}_s$ are parameters.

Following [Huang et al., 2013; Wu et al., 2019c], we use negative sampling techniques for model training. For each clicked news (regarded as a positive sample), we randomly select N non-clicked news that displayed in the same impression as negative samples. The loss function for recommendation is formulated as:

$$\mathcal{L}_{rec} = -\sum_{i=1}^{|\mathcal{S}|} \log\left(\frac{\exp(\hat{y}_i)}{\exp(\hat{y}_i) + \sum_{j=1}^{N} \exp(\hat{y}_{i,j})}\right), \quad (4)$$

where $\mathcal{S}$ is the training set, $\hat{y}_i$ and $\hat{y}_{i,j}$ are the click scores of the i_{th} positive sample and the associated j_{th} negative sample. The gold satisfaction score of the i_{th} positive news is computed as $s_i = \log_2(\hat{v}_i)$, where $\hat{v}_i$ is the personalized reading speed. Denote the predicted satisfaction score of this news as $\hat{s}_i$, then the loss function for content satisfaction prediction is formulated as:

$$\mathcal{L}_{sat} = \sum_{i=1}^{|\mathcal{S}|} |s_i - \hat{s}_i|. \quad (5)$$

We jointly train our model in both tasks. The final loss function to be optimized is a weighted summation of the loss functions of two tasks:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{sat}, \quad (6)$$

where λ is a coefficient to control the relative importance of the news content based satisfaction prediction task.

4 Experiments

4.1 Datasets and Experimental Settings

Since there is no publicly available news recommendation dataset that contains the full dwell time information of all news logs, we constructed one by collecting the 500,000 news impression logs from Microsoft News[2] during one month from 10/12/2019 to 11/13/2019. The logs in the last week were used for test, and the rest were used for training. We further randomly sampled 10% of the training data for validation. The statistics of this dataset are summarized in Table 1.

# Users	285,563	Avg. # Words Per News Title	11.65
# News	95,628	Avg. # Words Per News Content	648.31
# Impressions	500,000	Avg. Reading Speed (Word/s)	16.71

Table 1: Detailed dataset statistics.

In our experiments, word embeddings were 300-dimensional, and the personalized read speed embedding was 200-dimensional. We used Glove [Pennington *et al.*, 2014] to initialize the word embeddings. Following Wu et al. [2019c], the self-attention networks had 16 heads, and each head was 16-dimensional. The attention query was 200-dimensional. The negative sampling ratio N was 4. The intensity of dropout was 20%. The loss coefficient λ was 0.4. Adam [Kingma and Ba, 2015] was used as the optimizer. The batch size was set to 32. These hyperparameters were tuned on the validation set. Each experiment was repeated 10 times to mitigate occasionality. The metrics used in our experiments are the average AUC, MRR, nDCG@5 and nDCG@10 scores over all impressions.

4.2 Performance Comparison

To validate the effectiveness of our *CPRS* method, we compare it with many baseline methods, including: (1) *LibFM* [Rendle, 2012], a feature-based matrix factorization method. (2) *Wide&Deep* [Cheng *et al.*, 2016], a neural recommendation method which consists of a wide linear part and a deep neural part. (3) *DeepFM* [Guo *et al.*, 2017], a neural recommendation method which combines factorization machines and deep neural networks. (4) *DSSM* [Huang *et al.*, 2013], deep structured semantic model, which utilizes character trigram hashing and deep neural networks. (5) *DFM* [Lian *et al.*, 2018], a neural news recommendation approach, which integrates of neural networks with different depths. (6) *GRU* [Okura *et al.*, 2017], a neural news recommendation method which uses GRU to learn user representations and autoencoders to learn news representations. (7) *DKN* [Wang *et al.*, 2018], deep knowledge-aware network for news recommendation. (8) *DAN* [Zhu *et al.*, 2019], a neural news recommendation approach which uses CNN to learn news representations and uses attentive LSTM to learn user representations. (9) *NAML* [Wu *et al.*, 2019a], a neural news recommendation approach with attentive multi-view learning. (10) *NRMS* [Wu *et al.*, 2019c], a neural news recommendation approach with hierarchical multi-head self-attention. (11) *DTW* [Yi *et al.*, 2014], using dwell time to weight clicked

---

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

| Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM	57.68±0.65	17.93±0.59	18.68±0.62	23.12±0.60
Wide&Deep	59.16±0.47	18.79±0.46	19.21±0.44	24.04±0.45
DeepFM	58.34±0.51	18.14±0.47	18.75±0.52	23.59±0.59
DSSM	60.01±0.36	19.68±0.33	20.02±0.36	24.61±0.36
DFM	59.29±0.45	18.87±0.41	19.13±0.43	23.95±0.46
GRU	60.79±0.26	20.14±0.24	20.57±0.22	25.23±0.25
DKN	60.12±0.31	19.77±0.26	19.94±0.29	24.74±0.33
DAN	61.30±0.34	20.55±0.30	21.01±0.30	25.87±0.31
NAML	61.96±0.28	20.96±0.25	21.35±0.24	26.20±0.27
NRMS	62.59±0.31	21.38±0.29	21.76±0.26	26.69±0.28
DTW	63.24±0.24	21.43±0.24	22.28±0.20	27.31±0.18
CPRS*	**64.78**±0.22	**22.34**±0.18	**23.20**±0.16	**28.34**±0.16
CPRS-click	64.03±0.20	21.89±0.19	22.67±0.15	27.81±0.16

Table 2: News recommendation results. *Improvements over baselines are significant ($p < 0.01$).

news in the user encoder of our approach. (12) *CPRS-click*, a variant of our *CPRS* method with only click behaviors for model training. For fair comparison, all these methods use the titles and bodies of news to learn news representations.

The performance of different methods is summarized in Table 2. We have several findings from the results. First, compared with methods based on handcrafted features to represent news and users (e.g., *LibFM*, *DSSM* and *DFM*), the methods that utilize neural networks to learn news and user representations from news content perform better. This may be because handcrafted features may be sub-optimal in representing news and users, and using neural networks to build their representations may be more suitable for news recommendation. Second, the methods which consider the reading satisfaction of users (e.g., *CPRS* and *CPRS-click*) outperform other ones based on news clicks only. This is probably because news clicks cannot exactly reflect user interest, and incorporating reading satisfaction can model user interest more accurately. Third, the methods that use personalized reading speed (*CPRS* and *CPRS-click*) outperform the *DTW* method that uses dwell time to weight clicked news. This is because the latter method ignores the influence of news content lengths and user reading habits, which is inaccurate. Fourth, our *CPRS* approach outperforms its variant *CPRS-click*. This is probably because *CPRS* can consider the satisfaction signals into model training to capture user interest better.

4.3 Effectiveness of Reading Satisfaction Modeling

In this section, we conduct experiments to validate the effectiveness of modeling reading satisfaction via personalized reading speed. We compare the performance of *CPRS* using different ways to indicate reading satisfaction, i.e., dwell time, reading speed, and personalized reading speed. The results are shown in Fig. 4(a). From the results, we find that using dwell time is inferior to using reading speed in modeling reading satisfaction. This is because the difference in the length of news content is not considered by dwell time, which may be inaccurate to model satisfaction with different news. In addition, compared with using personalized reading speed, simply using reading speed is also not the optimal way. This is because the reading speed of different users can be very different, and regarding different users equally may be inaccurate to model the satisfaction of different users.
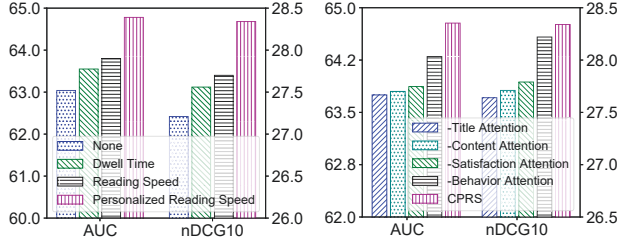
(a) Different satisfaction model-ing methods.

(b) *CPRS* and its variants with-out one kind of attention.

Figure 4: Effects of satisfaction modeling and attention networks.

Figure 5: Performance of *CPRS* w.r.t. different settings of λ.

Figure 6: Visualization of content and satisfaction attention. PRS denotes personalized reading speed.

4.4 Effectiveness of Attention Networks

In this section, we conducted several experiments to vali-date the effectiveness of different attention networks in our approach. We use the leave-one-out scheme to evaluate the contribution of each kind of attention network to *CPRS*. The results are shown in Fig. 4(b). From the results, we find that the title and content attention networks in our approach play important roles in our approach. It indicates that modeling the informativeness of news title and content is critical for learn-ing informative news and user representations. In addition, the satisfaction attention network in our approach can also ef-fectively improve the performance of our approach. It shows that selecting important news according to reading satisfac-tion can calibrate the user interest modeling based on news clicks. Besides, the behavior attention network is also useful. It shows that modeling the contributions of click and reading behaviors can help learn more accurate user representations. These results validate the effectiveness of the attention net-works in our approach.

4.5 Hyper-parameter Analysis

In this section, we study the influence of the loss coefficient λ in Eq.(6) on the performance of our approach. The perfor-mance of *CPRS* w.r.t. different values of λ is shown in Fig. 5. From the results, we find that the performance of *CPRS* im-proves quickly with the growth of λ. This is probably because when λ is too small, the model is not well-trained in the sat-isfaction prediction task, and useful satisfaction signals are not fully utilized. When λ goes too large, the performance starts to decline. This may be because the satisfaction pre-diction task is over-emphasized and the click prediction task does not receive sufficient attention. Thus, setting λ to a mod-erate value (e.g., 0.3) may be more suitable for our approach.

4.6 Case Study

In this section, we conduct several case studies to visually explore the effectiveness of our approach. We visualize the satisfaction attention weights and the content attention weights with and without satisfaction modeling of the his-torical clicked news from a randomly selected user. The re-sults are shown in Fig. 6. From the visualization results, we find the content attention without satisfaction signals may be inaccurate. For example, the fifth news is assigned a high attention weight, but the user reads the content of this news
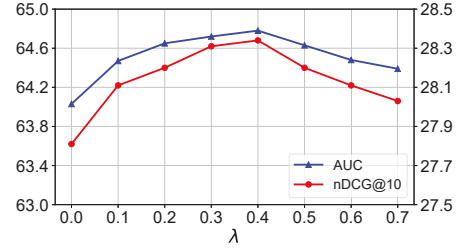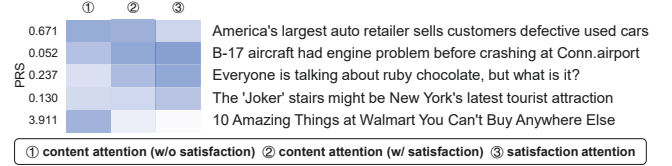
very fast before closing it, which is probably an indication of disappointment. Thus, it is insufficient to model user inter-ests based on news clicks only. After considering user satis-faction, the satisfaction attention mechanism can select news that the user reads more carefully and the content attention mechanism can also focus more on the news satisfied by this user. It shows that modeling reading satisfaction is beneficial for modeling user interest more accurately, and can further help recommend news that are satisfied by users.

5 Conclusion

In this paper, we propose a user modeling approach with click preference and reading satisfaction for news recommen-dation. We propose a personalized reading speed metric to model user satisfaction, which is based on dwell time, news content length, and the average speed of historical news read-ing. We learn a user representation from the news content read by a user with a content-satisfaction attention network to incorporate reading satisfaction into user interest model-ing. In addition, we learn another user representation from clicked news titles using an attention network to grasp click preferences in news titles. We fuse both kinds of user repre-sentations into a unified one in an attentive way. Besides, we propose to jointly train our model in two tasks, i.e., click pre-diction based on news title and satisfaction prediction based on news content, to encourage our model to recommend news that are satisfied by users rather than tend to be clicked only. Experimental results on real-world dataset validate that in-corporating reading satisfaction can effectively improve the quality of user modeling for news recommendation.

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 *ACL*, pages 336–345, 2019.

[Cheng *et al.*, 2016] 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. In *DLRS*, pages 7–10, 2016.

[Ge *et al.*, 2020] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Graph enhanced representation learning for news recommendation. In *WWW*, pages 2863–2869, 2020.

[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. In *AAAI*, pages 1725–1731, 2017.

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using click-through data. In *CIKM*, pages 2333–2338, 2013.

[Kim *et al.*, 2014] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM*, pages 193–202, 2014.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

[Li *et al.*, 2011] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *SIGIR*, pages 125–134, 2011.

[Lian *et al.*, 2018] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach. In *IJCAI*, pages 3805–3811, 2018.

[Liu *et al.*, 2010] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *IUI*, pages 31–40, 2010.

[Liu *et al.*, 2016] Yiqun Liu, Xiaohui Xie, Chao Wang, Jian-Yun Nie, Min Zhang, and Shaoping Ma. Time-aware click model. *TOIS*, 35(3):1–24, 2016.

[Lu *et al.*, 2018] Hongyu Lu, Min Zhang, and Shaoping Ma. Between clicks and satisfaction: Study on multi-phase user preferences and satisfaction for online news reading. In *SIGIR*, pages 435–444, 2018.

[Lu *et al.*, 2019] Hongyu Lu, Min Zhang, Weizhi Ma, Ce Wang, Feng xia, Yiqun Liu, Leyu Lin, and Shaoping Ma. Effects of user negative experience in mobile news streaming. In *SIGIR*, pages 705–714, 2019.

[Mehrotra *et al.*, 2017] Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan Awadallah, Ahmed El Kholy, and Madian Khabsa. User interaction sequences for search satisfaction prediction. In *SIGIR*, pages 165–174, 2017.

[Okura *et al.*, 2017] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In *KDD*, pages 1933–1942, 2017.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Rendle, 2012] Steffen Rendle. Factorization machines with libfm. *TIST*, 3(3):57, 2012.

[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 *NIPS*, 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 *WWW*, pages 1835–1844, 2018.

[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. In *IJCAI*, pages 3863–3869, 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 *KDD*, 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 *EMNLP-IJCNLP*, pages 6390–6395, 2019.

[Wu *et al.*, 2019d] Chuhan Wu, Fangzhao Wu, Junxin Liu, Shaojian He, Yongfeng Huang, and Xing Xie. Neural demographic prediction using search query. In *WSDM*, pages 654–662, 2019.

[Yi *et al.*, 2014] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *RecSys*, pages 113–120, 2014.

[Zhao *et al.*, 2019] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *RecSys*, pages 43–51, 2019.

[Zhou *et al.*, 2018] Tengfei Zhou, Hui Qian, Zebang Shen, Chao Zhang, Chengwei Wang, Shichen Liu, and Wenwu Ou. Jump: a joint predictor for user click and dwell time. In *IJCAI*, pages 3704–3710, 2018.

[Zhu *et al.*, 2019] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. Dan: Deep attention neural network for news recommendation. In *AAAI*, volume 33, pages 5973–5980, 2019.