Learning to Select Historical News Articles for Interaction based Neural News Recommendation

Anonymous Author(s)

ABSTRACT

The key to personalized news recommendation is to match the user's interests with the candidate news precisely and efficiently. Most existing approaches embed user interests into a representation vector then recommend by comparing it with the candidate news vector. In such a workflow, fine-grained matching signals may be lost. Recent studies try to cover that by modeling fine-grained interactions between the candidate news and each browsed news article of the user. Despite the effectiveness improvement, these models suffer from much higher computation costs online. Consequently, it remains a tough issue to take advantage of effective interactions in an efficient way. To address this problem, we proposed an end-to-end **S**elective **F**ine-grained **I**nteraction framework (SFI) with a learning-to-select mechanism. Instead of feeding all historical news into interaction, SFI can quickly select informative historical news w.r.t. the candidate and exclude others from following computations. We empower the selection to be both sparse and automatic, which guarantees efficiency and effectiveness respectively. Extensive experiments on the publicly available dataset MIND validates the superiority of SFI over the state-of-the-art methods: with only five historical news selected, it can significantly improve the AUC by 2.17% over the state-of-the-art interaction-based models; at the same time, it is four times faster.

KEYWORDS

News Recommendation, Interaction-based, Selection

1 INTRODUCTION

Nowadays, people are overwhelmed with information, exhausted to seek things they're interested in. Online news platforms e.g. MSN News[1] greatly alleviate this information overload problem by recommending news articles according to user's specific interests [16, 19, 31, 41]. The key technology of these news platforms is personalized news recommendation [13]. Due to the particular large-scale and time-sensitive property of news, the news recommenders must be both effective and efficient so that it can be deployed in real production systems.

A lot of existing news recommendation approaches [1, 1, 16, 21, 29–33, 39] follow a representation-based matching strategy. They learn a representation vector for the candidate news and encode the user's history news into a vector to form the user representation in the same semantic space. The matching score between these vectors is calculated as the click probability. However, the user vector is an aggregation of multiple historical articles, so it hardly keeps the fine-grained information and may contain noise in the articles. For example in Figure 1, the candidate news matches user's fine-grained interests *Biden* and *Hillary* (fine-grained interaction happens), which motivates the user's current click. Unfortunately, the aggregated user vector mixes all terms in $\mathbf{h}_1, \mathbf{h}_2$ and $\mathbf{h}_3$ and blurs these fine-grained interests, thus degrades the capacity of user modeling. Even worse, noises such as *wind* and *wildfires* are also included and they are unrelated to the current click. Though some recent "multi-channel" methods [2, 20] attempt to cover richer information by maintaining multiple representation vectors, they are still limited to modeling fine-grained interaction in an explicit and reliable manner.

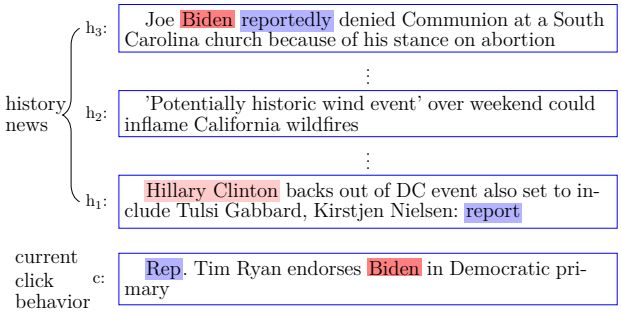

Figure 1: Example of a user's behavior log in MSN. Three historical news articles of the user are shown and c is the news she actually clicked afterward. The text marked in the same color is the fine-grained (term-term) matching signals. The darker the color, the more they match.

In order to capture fine-grained matching signals between the candidate news and the user, Wang et al. [26] proposed an interaction-based model. It computes similarity matrices between the candidate news and every historical news piece of the user at word level to derive the click probability. Despite the effectiveness improvement, the model is especially slow. It has to recompute term-level interaction matrices with every historical article when scoring each candidate, which is far more expensive than dot product used in representation-based methods. Intuitively, **we shouldn't involve all the historical articles in interaction**. For example, $\mathbf{h}_2$ should be excluded within this click because it is irrelevant to the current candidate $\mathbf{c}$ and there would be no interactions between them. Tailoring the user history to several recent browsed news items seems to be a straightforward solution to save efficiency. However, **blindly interacting with only the latest browsed news limits the recommendation effectiveness**, mainly because: 1) When the length of the kept browsing history is short, there isn't sufficient

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

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

news for the model to learn the user's interests well. Back to the Figure 1, if we cut off earlier history news h_2 and h_3, the interaction quality would greatly decrease since the most informative one, h_3, is lost. 2) When the capacity becomes larger, irrelevant historical news, such as h_2, is involved. As mentioned above, such news is harmful to the recommendation accuracy. It hardly contributes to motivate the click and would act like noise, reducing the matching score of the candidate that the user's truly interested in.

To tackle the above problem, in this paper, we propose **SFI**, a **S**elective **F**ine-grained **I**nteraction framework. The key idea of it is to **select a small number of historical news articles with higher informativeness, then perform fine-grained interaction over them only.**

The biggest challenge of SFI is to select the informative history news sparsely, precisely, and efficiently. Most previous works about feature selection employ gating operator [7, 18, 40]. But it cannot be directly used in SFI because gating doesn't eliminate zero entries so the total computations are not lessened. Two-stage training [8] is not desirable either, because no ground-truth label (indicates which historical news interacts with the candidate) is available to train the selector. Last but not least, we'd better manipulate news-level representations to guide selection for the sake of efficiency. In this work, we design the **learning-to-select** mechanism to fulfill all our goals. Specifically, SFI learns a selection vector for each news article, and computes the cosine similarity between candidate news and every history news in the selection space, taking the result as the informativeness of each historical article. Next, we design two successive selection networks. The hard-selection network enforces sparsity. It selects K most informative news and excludes others from following interactions. Within the output of the hard-selection, the following soft-selection network masks news whose informativeness is below a given threshold and attaches different weights to the unmasked ones. This refinement allows the gradient flow through to optimize the selection vectors, so that the model can learn to highlight valuable features for selection hence achieve higher effectiveness.

Extensive experiments on the publicly available dataset MIND show that SFI outperforms all baselines in terms of both effectiveness and efficiency: with only five historical news articles selected, it significantly improves the recommendation effectiveness by 2.17% over the state-of-the-art interaction-based models with four times faster speed (almost reaches the fastest speed of representation-based methods and outperforms it by 2.71% in AUC). We also comprehensively compare SFI with its naive recent K counterpart and investigate the efficiency effectiveness trade-off brought by SFI.

The main contributions of this paper can be summarized into three aspects:

(1) We propose SFI, a selective fine-grained interaction framework, to take full advantage of the fine-grained interaction in a highly efficient way.

(2) We design the learning-to-select mechanism to sparsely and automatically select informative historical news w.r.t. the candidate.

(3) We conduct extensive ablation studies to verify the advantage of selection; and further investigate the efficiency-effectiveness trade-off that SFI achieves.

2 RELATED WORK

In this section, we first review the traditional recommendation methods, then the neural news recommendation methods.

2.1 Traditional News Recommendation

A lot of traditional recommendations methods [4, 12, 15] are based on collaborative filtering (CF). CF-based methods cluster users by "co-visitation" relationships to recommend news to similar users [4]. Another line of CF studies apply Matrix Factorization [12] and Factorization Machine [23] to model the interaction between users and items. However, these methods face the problem of cold-start and sparsity, which is severe in news domain. They also require difficult and labor-consuming feature engineering. As the counterpart of CF, content-based recommendation methods become the main focus of news recommendation [14, 19] because of the rich text information in news articles.

2.2 Neural News Recommendation

In recent years, deep learning techniques are widely used in news recommendation systems and achieve better results than traditional methods. They can be categorized as follows:

2.2.1 Feature-based Methods. Following traditional recommendation approaches, feature-based methods feed the model with news content together with manually designed features, then employ neural networks to model the complex interactions among all the features [3, 6, 17]. For example, Cheng et al. [3] propose to combine shallow and deep neural networks to extract valuable information from a variety of manual features. Guo et al. [6] add deep layers over the factorization machine to model high order interactions.

2.2.2 Representation-based Methods. More methods proposed to learn representations of news and users from raw texts and browsing histories respectively [1, 10, 16, 21, 29–33, 39]. Numerous well designed models are proposed: multi-layer perceptron over trigrams [10], denoising auto-encoder [21], convolution neural networks [1, 16, 29–32, 39], and various attention-based methods [29, 30, 33]. Multi-channel structure is also explored [20]. Besides, some approaches [9, 34, 36] focused on using graph neural networks to represent news and users with their neighbors. Several methods [27, 28] proposed to incorporate knowledge to construct knowledge-aware representations of news and users.

In spite of the improvements these methods have made, all of them embed the news and the user into one or several one-fold vectors in the semantic space, where the fine-grained information is limited. And the representations can only meet each other in the prediction phase, which may impair fine-grained matching signals between the user and the candidate news.

2.2.3 Interaction-based Methods. To address the above problem, interaction-based models, which match user's interests with the candidate news at more delicate levels, are proposed. Wang et al. [26] designed the state-of-the-art interaction-based method for news recommendation. They constructed segment-to-segment similarity matrices between the candidate news and every historical news article of the user from 3 different granularities. Then they

use 3D-CNN to highlight salient matching signals to make recommendations. Although FIM achieves better results feature- and representation-based methods, its main drawback is the especially slow inference speed.

In this work, **we explore the interaction-based methods and aim to efficiently select fewer historical articles with higher value to perform interactions**. Several works in other fields have proposed to select important features for interaction [7, 18, 40], but they all employ the gating operator, which fails to discard the unselected items hence cannot be directly used to achieve our goal. The most related work [8] splits selection to another training stage, forbidding the model learning to select. Our proposed **learning-to-select** mechanism effectively addresses these issues.

3 OUR APPROACH

First we formulate the news recommendation problem. Given a user u, we have a set of historical news articles browsed by her at the platform, denoted as $\mathcal{H} = \{\mathrm{h}_1, \mathrm{h}_2, \cdots, \mathrm{h}_M\}$. For a candidate news c, our goal is to infer the probability that the user clicks this news article based on her browsing history $\mathcal{H}$, denoted as $p(\mathrm{c}|\mathcal{H})$.

The architecture of SFI is presented in Figure 2. Specifically, it contains four major modules. The **news encoder module** learns word- and news-level representations, the fine-grained ones are intended for interaction and the coarse-grained ones are further transformed for selection. The following **history selector module** manipulates news-level representation to efficiently and precisely select informative news from the user's browsing history. The fine-grained representations of selected news are fed into the **news interactor module** to compute interactions. The coarse-grained matching signals are also modeled in this module. Finally the **click predictor module** incorporates all matching signals to predict the click probability $p(\mathrm{c}|\mathcal{H})$. Next, we introduce each component in our model, especially the history selector.

3.1 News Encoder Module

Since users' click decisions on news platforms are usually based on the title of news articles [30], the *news encoder* learns the news representation from title only. Denote the word sequence of a news title as $S = \{\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_N\}$, where N is the length of S. First of all, we transform S into a sequence of vectors $E = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N\}$ by word embedding matrix $\mathbf{W_e} \in \mathbb{R}^{V \times D}$. V is the vocabulary size and D is the dimension of embeddings. Usually, the local contexts of a word across different spans play a big role in representing the word [26, 29, 30]. Therefore, we employ a hierarchical dilated convolution [38] to extract context features from different semantic granularities. In the l-th convolution layer, the representation of the i-th word is calculated as:

$$r_i^l = \mathrm{ReLU}\left(\mathbf{F_w} \times \bigoplus_{k=0}^{w} \mathbf{e}_{i \pm k\delta} + \mathbf{b}\right) \in \mathbb{R}^{f_s}, \tag{1}$$

where $\bigoplus$ means the concatenation operation for vectors. $\mathbf{F_w}$ is the convolution kernel of size $2w+1$, δ denotes dilation rate, $\mathbf{b}$ denotes bias and f_s denotes the number of filters. A detailed description of dilated convolution can be found in [26]. By hierarchically stacking dilated convolutions with expanding dilation rate, local contexts of different distances are fused into the word representations. Afterward, the output of each convolution layer is appended to the final representation of i-th word: $\mathbf{r}_i = \{\mathbf{r}_i^l\}_{l=0}^{L}$, where $\{\cdot\}$ denotes the vertical alignment of a matrix, and L denotes the total number of the stacked convolution layers.

The representation of each semantic level may contain information of different importance for matching. For example, in the news title "Restaurants to Satisfy Late Night Cravings in Louisville and Beyond", phrase-level local contexts "Late Night Craving" for the word "Night" matter more than that of sentence-level, e.g. "Restaurants ... Night ... And". Therefore, we use an attentive pooling technique [1, 29, 39] to highlight the important local contexts of a single word. Specifically, a trainable vectors $\mathbf{q_l} \in \mathbb{R}^{f_s}$ is introduced as the query of attention. The representation $\mathbf{r}'_i$ of the word w_i that fuses information across every semantic level is computed as:

$$\mathbf{r}'_i = \sum_{l=1}^{L} a_{il}\mathbf{r}_i^l, \qquad \text{where} \quad a_{il} = \frac{\exp(\mathbf{q_l}^\top \mathbf{r}_i^l)}{\sum_{j=1}^{L}\exp(\mathbf{q_l}^\top \mathbf{r}_j^l)}. \tag{2}$$

Similarly, different words may contribute differently in expressing the news. For example, "Louisville" is more informative than "Beyond" because it reveals the location. We use another query vector $\mathbf{q}_w \in \mathbb{R}^{f_s}$ to highlight the informative words in the news title and obtain the overall representation of the entire news:

$$\mathbf{r} = \sum_{i=1}^{N} a_i \mathbf{r}'_i, \qquad \text{where} \quad a_i = \frac{\exp(\mathbf{q_w}^\top \mathbf{r}'_i)}{\sum_{j=1}^{L}\exp(\mathbf{q_w}^\top \mathbf{r}'_j)}. \tag{3}$$

So far, the fine-grained representation of each word $\mathbf{r}_i = \{\mathbf{r}_i^l\}_{l=0}^{L}$ and the coarse-grained representation of the entire news $\mathbf{r}$ are generated by *news encoder*. We further explore other kinds of state-of-the-art encoders, and study their performance in Section 5.3.

3.2 History Selector Module

The *history selector* is the core component of our model. It selects the informative historical news sparsely, automatically, and efficiently with a *learning-to-select* mechanism. Then the selected news pieces are fed into next module for fine-grained interactions.

Denote the news-level representation of i-th clicked news in the user's history as $\mathbf{h}_i$, and that of candidate news as $\mathbf{c}$. Recall that the news-level representation is attentively aggregated from the word vectors in the title, which are optimized for the final matching and thus aren't selection-oriented. Therefore, directly using $\mathbf{h}_i$ and $\mathbf{c}$ for selection may lead to sub-optimal results. In learning-to-select, we project all the news-level representations into the same selection space using a fully-connected network to mitigate such conflicts:

$$\mathrm{Proj}(\mathbf{r}) = \mathbf{W_p}\mathbf{r} + \mathbf{b}, \tag{4}$$

where $\mathbf{r}$ could be either $\mathbf{h}_i$ or $\mathbf{c}$. Considering selection efficiency, we define the candidate-aware informativeness of every historical article as the cosine similarity between selection vectors:

$$\mathbf{s} = \{\cos\left(\mathrm{Proj}(\mathbf{h}_i), \mathrm{Proj}(\mathbf{c})\right)\}_{i=1}^{M}. \tag{5}$$

In selection, no supervision signals are available for $\mathbf{s}$, so it's critical to optimize the parameters end-to-end to allow the model learn valuable features for selection. Besides, sparsity must be enforced, otherwise the model would still be slow. Keeping both constraints in mind, we design two complementary selection networks.

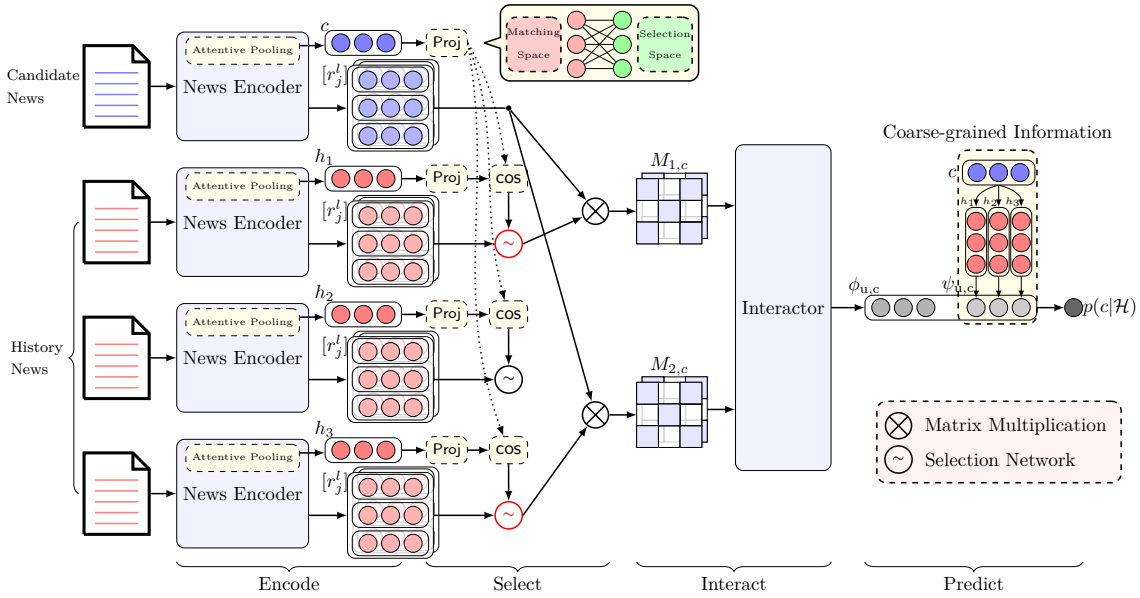

Figure 2: The architecture of our SFI model.

3.2.1 Hard-Selection Network. This sub-module enforces sparsity: it keeps the top K most informative history news and discards others. Formally,

$$\mathbf{x} = \text{argTOPK}(\mathbf{s}) \in \mathbb{R}^K, \tag{6}$$

where argTOPK($\mathbf{s}$) gets the index of the top K value in the vector $\mathbf{s}$. The corresponding history news sliced by $\mathbf{x}$ is selected and its fine-grained representations will get involved in the fine-grained interaction later. To do that, we first transform $\mathbf{x}$ to one-hot encoding matrix $\mathbf{X} = \text{one\_hot}(\mathbf{x}) \in \mathbb{R}^{K\times M}$, where the i-th row in $\mathbf{X}$ is the one-hot vector of $\mathbf{x}_i$, then use matrix multiplication to prune the browsing history to a smaller size:

$$\hat{H} = \mathbf{X} \otimes H, \qquad \hat{\mathbf{s}} = \mathbf{X} \otimes \mathbf{s}. \tag{7}$$

where $H = \{\{\mathbf{h}_i^l\}_{l=0}^L\}_{i=1}^M \in \mathbb{R}^{M\times L\times f_s}$ is the fine-grained representation tensor of user's browsed news, $\hat{H} \in \mathbb{R}^{K\times L\times f_s}$ is that of the selected news. By regulating hyper parameter K, we can elastically control the model's efficiency.

However, this sub-module has three defects that may limit the effectiveness: 1) Because $\mathbf{X}$ is sparse, the gradient cannot be passed to optimize $\mathbf{W_p}$; 2) Because the informativeness distributions of different users vary greatly, some noisy news articles are not filtered out among top K; 3) All of the selected news articles are weighted equally even though some of them are more informative. To tackle the above problems, a soft refinement is proposed.

3.2.2 Soft-Selection Network. This sub-module makes the gradient flow through selection. It is essentially a gating operator with a threshold [40], which further rules out noise (namely authentically uninformative news) and improves effectiveness. Given the output of hard-selection, the soft-selection network masks the news whose informativeness is below the threshold and attaches different importance to the unmasked ones:

$$\tilde{H} = \hat{H} \odot \text{Expand}(\tilde{\mathbf{s}}), \qquad \tilde{\mathbf{s}} = \sigma(\hat{\mathbf{s}}), \tag{8}$$

$$\sigma(s_i) = \begin{cases} 0 & s_i < \gamma, \\ h_i & \text{otherwise.} \end{cases} \tag{9}$$

$\odot$ is Hadamard Product, and γ denotes the threshold. $\sigma(\cdot)$ is element-wise. Expand($\tilde{\mathbf{s}}$) repeats the elements in $\tilde{\mathbf{s}}$, expanding it into $\mathbb{R}^{K\times L\times f_s}$. $\tilde{H}$ is the refined $\hat{H}$, where all representations of the news whose informativeness is lower than γ are masked as 0.

The number of the authentically informative news articles is floating per candidate, so a dynamic quantity of news items is kept. This entitles more flexibility to the selection operation. Meanwhile, with $\tilde{s}$ attached to $\hat{H}$, the *news interactor* can attend to more informative ones, and the parameters in Proj($\cdot$) can be optimized by the selecting step since the element-wise multiplication is differentiable. This helps SFI to learn features that are important for selection and will enhance the effectiveness remarkably.

In back propagation, gradient from the loss function is applied to the *news interactor*, then to the selected fine-grained representation tensor $\tilde{H}$. For simplification, $\tilde{H}$ is reshaped into a vector $\tilde{\mathbf{R}} \in \mathbb{R}^{1\times(K\times d)}$ where $d = L \times f_s$, together with its gradient $\nabla_{\tilde{\mathbf{R}}} = \text{reshape}(\nabla_{\tilde{H}}) \in \mathbb{R}^{1\times(K\times d)}$. The same operation is taken for $\hat{H}$, forming $\hat{\mathbf{R}} \in \mathbb{R}^{1\times(K\times d)}$. Then the gradient for $\mathbf{W_p}$ is:

$$\nabla_{\mathbf{W_p}} = \mathbf{Z}^\top \otimes \nabla_{\mathbf{s}} \otimes \mathbf{c}_1' + \mathbf{c}^\top \otimes \nabla_{\mathbf{s}}^\top \otimes \mathbf{Z}_1, \tag{10}$$

$$\begin{aligned} \nabla_{\mathbf{s}} &= \mathbf{X}^\top \otimes \nabla_{\hat{\mathbf{s}}} \\ &= \mathbf{X}^\top \otimes (\nabla_{\tilde{\mathbf{s}}} \odot g(\hat{\mathbf{s}})) \\ &= \mathbf{X}^\top \otimes \left(\left\{ \sum_{j=1}^{d} \left(\nabla_{\tilde{\mathbf{R}}} \otimes \text{Diag}(\hat{\mathbf{R}}) \right) [i,j] \right\}_{i=1}^{K} \odot g(\hat{\mathbf{s}}) \right), \end{aligned} \tag{11}$$

where $\mathrm{Diag}(\hat{\mathbf{R}})$ is the matrix with the elements of $\hat{\mathbf{R}}$ as the diagonal. $\mathbf{Z} = \{\mathbf{h}_i\}_{i=1}^{M} \in \mathbb{R}^{M \times f_s}$ is the news-level representation matrix of the historical news, and $\mathbf{Z}_1 = \{\mathrm{Proj}(\mathbf{h}_i)\}_{i=1}^{M}, \mathbf{c}_1 = \mathrm{Proj}(\mathbf{c})$ are the corresponding selection vectors. $g(s_i)$ is the derivative for $\sigma(s_i)$:

$$g(s_i) = \begin{cases} 0 & s_i < \gamma, \\ 1 & \text{otherwise.} \end{cases}$$

In this way, the gradient safely flows through the selection stage and reaches $\mathbf{s}$, to increase the score of the useful news pieces and vice versa. It is further spread to optimize $\mathbf{W_p}$ to achieve the above adjustment, however, only from the selected entries.

3.3 News Interactor Module

The selected historical news articles are fed into this module to perform fine-grained interactions with the current candidate. We denote the representation of the words in v-th selected news as $\mathbf{d}_v = \{\mathbf{t}_i\}_{i=1}^{N}$ where $\mathbf{t}_j = \{\mathbf{t}_j^l\}_{l=0}^{L} \in \mathbb{R}^{L \times f_s}$ is the stacked representation of j-th word. Similarly, the representation of each word in the current candidate news is $\mathbf{c}^{\mathrm{f}} = \{\mathbf{p}_j\}_{j=1}^{N}$. Resembling FIM [26], we construct pair-to-pair similarity matrix $\mathbf{M}_{v,\mathrm{c}}^{l}$ of l-th semantic granularity, where each entry is the scaled dot product between the fine-grained representations of v-th selected news and the candidate news:

$$\mathbf{M}_{v,\mathrm{c}}^{l}[i,j] = \frac{\mathbf{t}_i[l]^\top \mathbf{p}_j[l]}{\sqrt{f_s}} \in \mathbb{R}^{N \times N}. \tag{12}$$

Next, the similarity matrices of each granularity across all the selected history news are fused into a 3D cube $O \in \mathbb{R}^{L \times K \times N \times N}$, where a series of 3D CNN and 3D max pooling is applied to highlight the significant matching signals. Outputs of the final pooling layer are flattened as the vector containing fine-grained interactive information across the user and candidate news, denoted as $\phi_{\mathrm{u,c}}$. Other state-of-the-art interactors are studied in Section5.3.

In SFI, fine-grained matching information $\phi_{\mathrm{u,c}}$ only engage selected news articles. However, it is important not to leave out the unselected ones. Although conducting fine-grained interactions on them is unnecessary, we still value the coarse-grained matching signals of them, which come from the matching between news-level representations:

$$\psi_{\mathrm{u,c}} = \{\psi_{\mathbf{h}_1,\mathbf{c}}, \psi_{\mathbf{h}_2,\mathbf{c}}, \cdots, \psi_{\mathbf{h}_M,\mathbf{c}}\}, \qquad \psi_{\mathbf{h}_i,\mathbf{c}} = \mathbf{h}_i^\top \mathbf{c}. \tag{13}$$

$\psi_{\mathrm{u,c}}$ gives an overall matching degree of the user and the candidate news and is complementary to $\phi_{\mathrm{u,c}}$. It facilitates the model to learn more precise correspondences between the matching signals and the click probability. Another critical point is that by involving $\psi_{\mathrm{u,c}}$ to score the candidate, the gradient can be delivered by all of the historical news articles rather than only the selected ones.

3.4 Click Predictor

The *click predictor* module incorporates the output from *news interactor* then predicts the probability of a user clicking on a candidate news article. The news articles with higher click probability are ranked higher in the final user interface.

Given vectors containing course- and fine-grained matching information, $\psi_{\mathrm{u,c}}$ and $\phi_{\mathrm{u,c}}$ respectively, we propose to incorporate both by:

$$y_{\mathrm{u,c}} = \mathbf{W_c}\left\{\phi_{\mathrm{u,c}}, \psi_{\mathrm{u,c}}\right\} + \mathbf{b}. \tag{14}$$

Table 1: Statistics of the MIND dataset

#users	1,000,000	**#news**	161,013
#impressions	15,777,377	**#clicks**	24,155,470
avg. title len	11.52	**avg. his len**	32.99

Following [10, 30], we use negative sampling to simulate the unbalanced distribution of clicked news in an impression. For each ground-truth candidate, we randomly sample m news that is not clicked by her in the same impression as negative samples:

$$\hat{p}(\mathrm{c}|\mathcal{H}) = \frac{\exp(y_{\mathrm{u,c}}^{+})}{\exp(y_{\mathrm{u,c}}^{+}) + \sum_{j=1}^{m} \exp(y_{\mathrm{u,c}}^{-})}. \tag{15}$$

Thus, it is converted to a $m+1$ classification problem, and the negative log likelihood loss is going to be minimized when training:

$$\mathcal{L} = -\sum_{\mathrm{c} \in \mathcal{S}} \log \hat{p}(\mathrm{c}|\mathcal{H}), \tag{16}$$

where $\mathbf{c}$ is the ground-truth news piece which the user clicked, and $\mathcal{S}$ denotes all training samples.

Finally, we jointly train the *news encoder*, *history selector*, *news-interactor* and *click predictor* through the final click signal. In such a way, the model can better learn dependencies among modules.

4 EXPERIMENTAL

4.1 Datasets and Experimental Settings

Our experiments are conducted on MIND [35], a large-scale dataset collected from the users' click logs of the Microsoft News platform from Oct. 12 to Nov. 22, 2019. The statistics of MIND are shown in Table 1. We use the same training-testing partition as [35].

In our experiments, the dimension D of word embeddings is set to 300. We use the pre-trained Glove embeddings [22], to initialize the embedding matrix $\mathbf{W_e}$. The maximum length of news titles is set at 20. the maximum number of clicked news for learning user representations was set to 50. In the *news encoder*, we stack 3 convolution layers with dilation rates $[1, 2, 3]$. The kernel size and the number of filters is set to 3 and 150 respectively. We employ a 2-layer composition for *news-interactor* module, the output channels and the window size is set at $32 - [3, 3, 3]$ and $16 - [3, 3, 3]$. Each convolution component is followed by a max pooling layer with size $[3, 3, 3]$ and stride $[3, 3, 3]$. We apply the dropout strategy [25] to the word embedding layer to mitigate overfitting. The dropout rate is set at 0.2. Adam [11] is used as the optimization algorithm.

The batch size is set to 100 when training and 400 when predicting, and the encoding process is executed offline when predicting. Since there are 40 kernels in total on our machine, we set 40 parallel threads to load data in order to minimize the latency caused by processing data. We independently repeat each experiment for 5 times and report the average performance. We conduct all experiments on a machine with Xeon(R) Silver 4114 CPUs and a TITAN V GPU[2].

[2]We will release the code and scripts based upon the acceptance of the paper.

4.2 Evaluation Metrics

Following existing studies, we use the average AUC, MRR, nDCG@5, and nDCG@10 scores over all impressions to evaluate the effectiveness of the models. All results come from the official test entry. Moreover, given the same batch size, we use the prediction speed i.e. iterations per second to evaluate the efficiency. In one iteration, the batch size of candidate news articles is scored.

4.3 Baselines

We compare SFI with the following baseline methods:

(1) General Recommendation Methods: **LibFM** [24], a state-of-the-art feature-based matrix factorization approach for recommendation[3]; **DSSM** [10], a deep structured semantic model that uses multiple dense layers upon tri-grams. All of the users' clicked news are concatenated as the query, and the candidate news is regarded as documents; **Wide&Deep** [3], a widely used recommendation method that uses the combination of a wide channel and a deep channel for memorization and generalization; **DeepFM** [6], a popular neural recommendation method which combines factorization machine with deep neural networks;

(2) Representation-based Methods: **DFM** [17], which uses dense layers for different channels and attentively fuse outputs; **GRU** [21], which learn news representations with an auto-encoder and utilizes GRU to learn user representations; **Hi-Fi Ark** [20], a multi-channel representation approach for recommendation; **NPA** [30], which highlights informative words and news with personalized attention; **NRMS** [33], which learns delicate representations of news and users by multi-head self-attention; **LSTUR** [1], which models long- and short-term user interests with GRU;

(3) Interaction-based Methods: **FIM** [26], the state-of-the-art interaction-based approach for neural news recommendation, which encodes news by hierarchical dilated CNN and performs interaction between each of the user browsed news articles and the candidate. **Recent(K)**, the naive counterpart of SFI, which keeps the recent K historical news for interaction only (Recent(50) equals FIM).

5 EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Overall Results of Effectiveness

The overall recommendation effectiveness of all models is shown in Table 2. Based on the results, we have the following observations:

(1) **Our proposed model SFI consistently outperforms other baselines in terms of all metrics**. On the one hand, SFI captures fine-grained interactions to model user interests, gaining 2.71% up to 3.23% AUC improvements over all of the state-of-the-art representation-based methods. On the other hand, SFI(K) outperform Recent(K) baseline by 6.4% and 2.17% when $K = 5$ and 50 respectively. This result substantiates the power of the *learning-to-select* mechanism.

(2) The variant SFI(50) that keeps the entire browsing history outperforms SFI(5) that selects only 5 historical news. This is as expected because after removing noise, SFI(50) covers richer information to model the user. Interestingly, the improvement is tiny compared with the margin between Recent(50) i.e. FIM and Recent(5). We will study this phenomenon in detail in Section 5.4.1.

[3]The TF-IDF features are used.

Table 2: The performance of different methods for news recommendation. The number of news items involving in interactions is bolded in ( ) for interaction-based methods. The result with superscript * is referencing the one in [35] where MIND is presented. † indicates a significant improvement over all baselines with paired t-test ($p < 0.01$).

Type	Methods	AUC	MRR	NDCG@5	NDCG@10
General methods	LibFM*	59.93	0.2823	30.05	35.74
	DSSM*	64.31	0.3047	33.86	38.61
	Wide&Deep*	62.16	0.2931	31.38	37.12
	DeepFM*	60.30	0.2819	30.02	35.71
Represent. based	DFM*	62.28	0.2942	31.52	37.22
	GRU*	65.42	0.3124	33.76	39.42
	Hi-Fi Ark	65.87	0.3119	33.64	39.25
	NPA*	66.69	0.3224	34.98	40.68
	LSTUR*	67.73	0.3277	35.59	41.34
	NRMS*	67.76	0.3305	35.94	41.63
Interaction based	FIM	68.12	0.3354	36.45	42.11
	Recent (**5**)	65.39	0.3164	34.14	39.78
	SFI (**5**)	69.60†	0.3475†	37.86†	43.51†
	SFI (**50**)	**69.95**†	**0.3503**†	**38.31**†	**43.97**†

Table 3: The inference speed comparison of different methods for news recommendation. The improvement over FIM is given in the bracket.

Methods	Inference Speed	AUC	nDCG@5
NRMS	121.54	67.76	35.94
FIM	20.53	68.12	36.45
Recent (**5**)	125.96(↑ 517%)	65.39(↓ 4.01%)	34.14(↓ 6.34%)
Recent (**25**)	40.05(↑ 96%)	67.32(↓ 1.17%)	35.16(↓ 3.54%)
SFI (**5**)	**99.57**(↑ **385**%)	69.60(↑ 2.17%)	37.86(↑ 3.87%)
SFI (**25**)	33.11(↑ 66%)	69.75(↑ 2.39%)	38.01(↑ 4.28%)

(3) The interaction-based methods for news recommendation outperform all representation-based methods, which validates the benefit of capturing fine-grained matching signals. However, simply pruning the user's history to a smaller size to save speed is not feasible because it hurts the effectiveness seriously.

Without Bert [5], expanded SFI (with extra news abstract) ranks among the top 15 on the official testing leaderboard.

5.2 Results of Efficiency

Since the motivation of SFI is mainly concerned with efficiency, we further compare the inference speed between SFI and several baselines. Results in Table 3 substantiates the superiority of SFI: with the selection capacity of **5**, it can infer almost **four** times faster than the state-of-the-art interaction-based method, while significantly improving the recommendation effectiveness. SFI(5) also achieves comparable speed with the state-of-the-art representation-based method NRMS, and outperforms it by **2.71**% in AUC. The efficiency of SFI(K) and Recent(K) significantly drops from $K = 5$ to $K = 25$, which will be further studied in Section 5.4.1.

Table 4: The effectiveness of SFI with different *news encoders* and *news interactors*.

Encoders \ Interactors	2D-CNN	3D-CNN	MHAI	KNRM
PCNN	63.71	63.95	65.54	63.24
HDCNN	68.45	**69.56**	63.66	60.01
MHA	66.44	65.56	61.31	63.92
LSTM	68.58	67.91	68.39	64.12

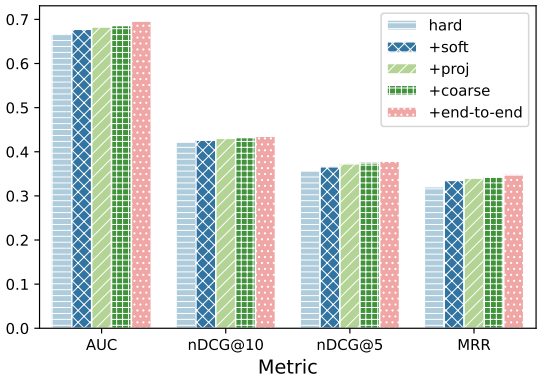

Figure 3: The effectiveness of history selector and coarse-grained information.

5.3 Ablation Study

Since SFI is essentially a flexible framework, we conduct extensive ablation studies to gain comprehensive insights into every module. In each subsection, we pose our claim first before explanations.

5.3.1 ***HDCNN and 3DCNN are the most effective encoder and interactor among a variety of state-of-the-art architectures.*** In the interaction-based workflow, the history selector can be easily inserted between any kind of news encoder and interactor. This flexibility motivates us to study how state-of-the-art encoders and interactors would perform. We compare among 1D-CNN with Personalized Attention [30] (denoted as PCNN), Hierarchical Dilated CNN [26] (denoted as HDCNN), Multi-head Self Attention [33] (denoted as MHA), LSTM for the *news encoder* and 2D-CNN, 3D-CNN [26], KNRM [37], Multi-Head Self Attention [33] (denoted as MHAI) for the *news interactor*. The AUC scores are reported in Table 4. We find **Hiearchical Dilated CNN** combined with **3D-CNN** is the best setting.

5.3.2 ***Every sub-module in history selector is critical to improving effectiveness.*** The *learning-to-select* mechanism comprises three parts: a selection projection, a hard selection, and a soft refinement. The hard selection is the cornerstone of our work so we no longer verify its impact. For the other two components, we compare SFI with the variant that applies only hard-selection, and that applies hard-selection followed by soft-selection without learning extra selection vectors. The result is reported in Figure 3. As we observe, the soft-selection network improves the effectiveness.

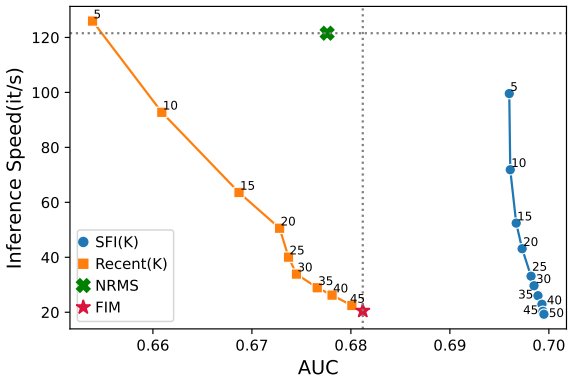

Figure 4: The efficiency and effectiveness of SFI with different numbers of selected news. The number next to the marker indicates the selected news count.

This is because it filters the authentically uninformative history news, and makes the gradient flow through to optimize the representation vectors used for selection. However, without selection projection, these news-level representations are optimized for two incompatible goals: selecting and matching, which may decrease the recommendation accuracy. Experiments validates our claim: the model benefits a lot from selection projection. Thanks to it, SFI can encode selective features into selection vectors, leaving the news-level representations to focus on the final matching.

5.3.3 ***The coarse matching signals of the unselected articles are also important.*** In Figure 3, SFI outperforms its variant that totally abandons the coarse-grained matching signals. This observation verifies that the coarse-grained matching signals are complementary. Note that doing so won't reduce efficiency because batched matrix multiplication is fast on GPU.

5.3.4 ***SFI benefits from end-to-end training.*** When deployed in production, the news encoding process could be done offline to speed up inference, known as a pipeline convention. It's natural to migrate it to the training phase, where we first pre-train SFI without the *history selector* to acquire coarse-and-fine representations of news. Then we replace the *news encoder* with a lookup table constructed from these representations and fine-tune them with *history selector* applied. End-to-end training is the counterpart, in which we jointly train the *news encoder*, *history filter*, *news-interactor* and *click predictor* by the final classification loss simultaneously. In Figure 3, the first four bars in every group are the performance of SFI trained in pipeline, and the last bar is that of trained end-to-end under the same setting. As expected, end-to-end training leads to better effectiveness. It's because optimizing the parameters rather than directly updating the vectors can make the *news encoder* learn more precise representations for both selection and interaction. Also, the modules can better learn the correspondence among them in end-to-end training.

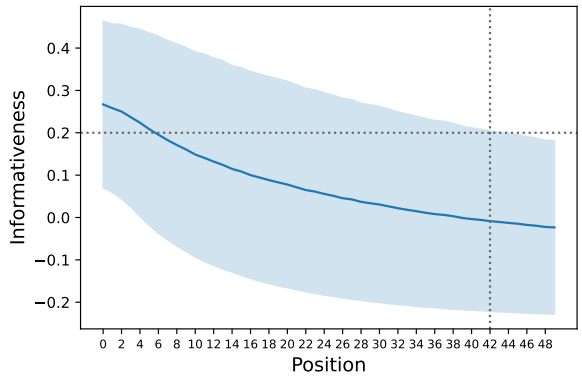

Figure 5: The informativeness score of history news at each position. Smaller x-axis represents more recent history.

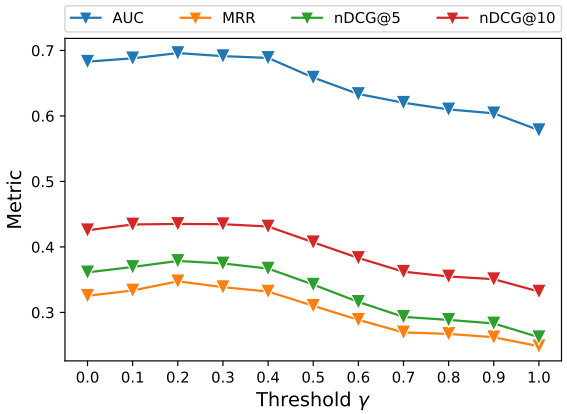

Figure 6: The effectiveness of SFI with different value of γ.

5.4 Hyper Parameter Analysis

5.4.1 Influence of the Interaction Capacity. The predefined interaction capacity K is the most important hyper parameter since the efficiency-effectiveness trade-off is up to it. We study its influence by drawing the inference speed curve against the AUC score of the model with different K in Figure 4. Motivated by several observations in Section 4, we also include Recent(K), which only interacts with the latest K historical news to save efficiency. We add two dashed lines to mark the best effectiveness and efficiency that baseline models ever achieve. The optimal result would be situated at the upper right corner.

According to the figure, we find: **First**, from $K = 5$ to $K = 50$, SFI(K) is far more effective than its naive counterpart Recent(K), this again validates the effectiveness of *history selector*. However, due to the time consumption of selection, SFI(K) is a bit slower. Overall, SFI(5) is the optimal setting because it greatly outperforms NRMS and all Recent(K) including FIM, while providing a much higher efficiency over FIM. **Second**, when K is growing, the effectiveness of both Recent(K) and SFI(K) is improving, which is because a bigger capacity keeps richer information to learn the user's interests. As a side effect, the model becomes slower. **Third**, as K increases, the effectiveness of SFI(K) grows slower than Recent(K) and is about saturated at $K = 40$. Intuitively, with the *learning-to-select* mechanism, SFI(K) can consistently select the most effective articles for interaction, so increasing the capacity only brings a little more valuable information. In contrast, Recent(K) cannot access the informative historical news in earlier history unless the capacity is big enough.

These observations motivate us to study what informativeness scores of the historical articles at different positions are learnt by the model itself. We report the informativeness score at each history position (averaged from $K = 5$ to $K = 50$) in Figure 5. The blue line is the mean value of informativeness, and the shade indicates standard deviation. The horizontal black line marks the threshold of the soft-selection. According to the figure, the increasing mean value of informativeness tells us that more recent reading history helps more in expectation. At the same time, the significantly high variance confirms that historical news at each position has the potential to interact with the candidate. Therefore, with a small K, Recent(K) cannot access earlier historical news that tends to be useful in interaction. Rather, SFI is quite able to inspect them and involve them in fine-grained interactions intelligently. Moreover, the informativeness of the history news whose position is farther than 40 hardly reaches the threshold. So they are considered authentically uninformative and masked even though they are among the top $K \geq 40$. This explains the saturation of SFI's effectiveness and justifies our intuition.

5.4.2 Influence of the Informativeness Threshold. Another crucial factor of SFI is the informativeness threshold in the soft-selection network. The effectiveness of SFI with different threshold settings is shown in Figure 6. In summary, the threshold shouldn't be too large or too small. When $\gamma < 0.1$, almost all history news articles are considered informative, so the selection fails. When *gamma* > 0.3, the *history selector* rules out too many history news articles, including the valuable ones. The gradient cannot be passed adequately, either. Hence the model's effectiveness declines. When it reaches 1, all fine-grained representations are masked as 0, completely disabling the *news interactor*. Recall that the coarse-grained matching signals persist, leading to better results than random recommendation. Overall, $\gamma = 0.2$ is the optimal configuration.

6 CONCLUSION AND FUTURE WORK

Capturing fine-grained interactions brings more accuracy and higher online costs for news recommenders. In this work, we proposed a selective fine-grained interaction framework to select a small number of valuable historical articles for interaction, drawing a good balance between efficiency and effectiveness. With the help of the **learning-to-select** mechanism, the selection can be performed efficiently, sparsely, and automatically. Experimental results show SFI can significantly improve the recommendation effectiveness by 2.17% over the state-of-the-art models with four times faster speed. We experimented a lot to provide comprehensive insights of SFI and studied the efficiency-effectiveness trade-off it achieves. In the future, we will dig deeper into representing users with terms to further improve the efficiency while keeping the effectiveness.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *ACL,2019*. Association for Computational Linguistics, 336–345.

[2] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 108–116. https://doi.org/10.1145/3159652.3159668

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS@RecSys, 2016*. ACM, 7–10.

[4] Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyamsundar Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW,2010*. ACM, 271–280.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT,2019*. Association for Computational Linguistics, 4171–4186.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI,2017*. ijcai.org, 1725–1731.

[7] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 55–64. https://doi.org/10.1145/2983323.2983769

[8] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1349–1358. https://doi.org/10.1145/3404835.3462889

[9] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *ACL,2020*. Association for Computational Linguistics, 4255–4264.

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

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR,2015*.

[12] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[13] Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. 2010. User attitudes towards news content personalization. *Int. J. Hum. Comput. Stud.* 68, 8 (2010), 483–495.

[14] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (Eds.). ACM, 661–670. https://doi.org/10.1145/1772690.1772758

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

[16] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *KDD,2020*. ACM, 2009–2019.

[17] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach. In *IJCAI,2018*. ijcai.org, 3805–3811.

[18] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 2636–2645. https://doi.org/10.1145/3394486.3403314

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

[20] Zheng Liu, Yu Xing, Fangzhao Wu, Mingxiao An, and Xing Xie. 2019. Hi-Fi Ark: Deep User Representation via High-Fidelity Archive Network. In *IJCAI,2019*. ijcai.org, 3059–3065.

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

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *ACL,2014*. ACL, 1532–1543.

[23] Steffen Rendle. 2010. Factorization Machines. In *ICDM,2010*. IEEE Computer Society, 995–1000.

[24] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3, 3 (2012), 57:1–57:22.

[25] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[26] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained Interest Matching for Neural News Recommendation. In *ACL,2020*. Association for Computational Linguistics, 836–845.

[27] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW,2018*. ACM, 1835–1844.

[28] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *WWW,2019*. ACM, 3307–3313.

[29] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *IJCAI,2019*. ijcai.org, 3863–3869.

[30] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural News Recommendation with Personalized Attention. In *SIGKDD,2019*. ACM, 2576–2584.

[31] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Topic-Aware News Representation. In *ACL,2019*. Association for Computational Linguistics, 1154–1159.

[32] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Heterogeneous User Behavior. In *EMNLP-IJCNLP,2019*. Association for Computational Linguistics, 4873–4882.

[33] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP-IJCNLP,2019*. Association for Computational Linguistics, 6388–6393.

[34] Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. 2019. Reviews Meet Graphs: Enhancing User and Item Representations for Recommendation with Hierarchical Attentive Graph Neural Network. In *EMNLP-IJCNLP,2019*. Association for Computational Linguistics, 4883–4892.

[35] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *ACL,2020*. Association for Computational Linguistics, 3597–3606.

[36] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI,2019*. AAAI Press, 346–353.

[37] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR,2017*. ACM, 55–64.

[38] Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR,2016*.

[39] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation. In *IJCAI,2019*. ijcai.org, 4213–4219.

[40] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *EMNLP-IJCNLP,2019*. Association for Computational Linguistics, 111–120.

[41] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *WWW,2018*. ACM, 167–176.