World Wide Web
https://doi.org/10.1007/s11280-022-01059-6

CupMar: A deep learning model for personalized news recommendation based on contextual user-profile and multi-aspect article representation

Dai Hoang Tran[1] · Quan Z. Sheng[1] · Wei Emma Zhang[2] · Nguyen H. Tran[3] · Nguyen Lu Dang Khoa[4]

Received: 10 January 2022 / Revised: 5 April 2022 / Accepted: 19 April 2022

Abstract
In modern days, making recommendation for news articles poses a great challenge due to vast amount of online information. However, providing personalized recommendations from news articles, which are the sources of condense textual information is not a trivial task. A recommendation system needs to understand both the textual information of a news article, and the user contexts in terms of long-term and temporary preferences via the user's historic records. Unfortunately, many existing methods do not possess the capability to meet such need. In this work, we propose a neural deep news recommendation model called CupMar, that not only is able to learn the user-profile representation in different contexts, but also is able to leverage the multi-aspects properties of a news article to provide accurate, personalized news recommendations to users. The main components of our CupMar approach include the News Encoder and the User-Profile Encoder. Specifically, the News Encoder uses multiple properties such as news category, knowledge entity, title and body content with advanced neural network layers to derive informative news representation, while the User-Profile Encoder looks through a user's browsed news, infers both of her long-term and recent preference contexts to encode a user representation, and finds the most relevant candidate news for her. We evaluate our CupMar model with extensive experiments on the popular Microsoft News Dataset (MIND), and demonstrate the strong performance of our approach.

Keywords Recommendation systems · News recommendation · Personalized recommendation · Neural networks · Attention mechanism · Contextual profile · Multi-aspects news

Quan Z. Sheng, Wei Emma Zhang, Nguyen H. Tran and Nguyen Lu Dang Khoa contributed equally to this work.

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2021*

Guest Editors: Hua Wang, Wenjie Zhang, Lei Zou, and Zakaria Maamar

Extended author information available on the last page of the article

Published online: 10 May 2022

1 Introduction

Reading news to stay up-to-date with the latest information has been an integral part of human life. In the modern days, the Word Wide Web provides us with abundant online news resources, enabling us to keep updated with the current occurrences. However, due to the large amount of news articles as well as the proliferation of news websites, users may get overwhelmed to decide what and where to read the news for their needs. As a result, online news websites such as Google News[1] or Bing News[2] try to solve the problem by aggregating many sources of news and generating a personalized reading list to the users based on their preferences. This strategy of recommending news that is tailored for each user has been an effective way to target user reading interests [1–5].

However, news recommendation poses several challenges in comparison to the traditional recommendation problem. First and foremost, unlike movies or shopping items recommendation, news articles are *time sensitive* items. The value and relevance of a news articles deteriorate quickly over a short period of time because fresh news items[3] are updated frequently. Due to this time sensitive property, traditional methods like collaborative filtering [6], which depends on the identity (ID) of users or items, would not work efficiently. Second, the content of a news article contains *dense textual data*, which also encodes the latent preferences of the users. For instance, certain individuals may only read sport news and discard the rest. Thus this is a strong signal indicating the "Sports" category encodes one of their long-term preferences. As another example, some users sporadically click on the latest news that has title or content related to the celebrities, and this behaviour shows a strong short-term interest signal, in which the article's content exhibits certain words or knowledge patterns. This second problem prompts a strong need to measure a user's reading history to infer her latent preferences, either long-term, short-term or a mixture of the both. Other aspects like *diversity*, also plays a significant role in news recommendation. Users should be able to find the types of news that they have high interest in, but also can explore other news that may pique their curiosity. This diversity can significantly improve users' satisfaction and retain their loyalty to these online news services. Essentially, the core tasks of solving news recommendation problems are i) capturing a user preferences from their reading history and ii) understanding all of the signals from a news article.

Figure 1 illustrates a scenario where understanding of different contexts between a user and a news item is very important. In this scenario, each news item has several features, such as the category, the knowledge entity inside the news, and the title. Each user can select different news item based on the number of news articles that is shown to them. It is very clear that User-1 only selects the items in the *Sports* category, which defines her long-term preferences. User-2 and User-3 demonstrate different behaviours. User-2 shows attention to the news items which contain the knowledge entities about countries, while User-3 only reads the latest news. Clearly, these behaviours display their recent preferences.

Given these challenges, researchers and industry partners resort to deep learning and also spend a significant amount of time to collect the right datasets to facilitate the development of news recommendation. In this paper, we address news recommendation problem

[1] news.google.com

[2] bing.com/news

[3] In this paper, we use news article and news item interchangeably.

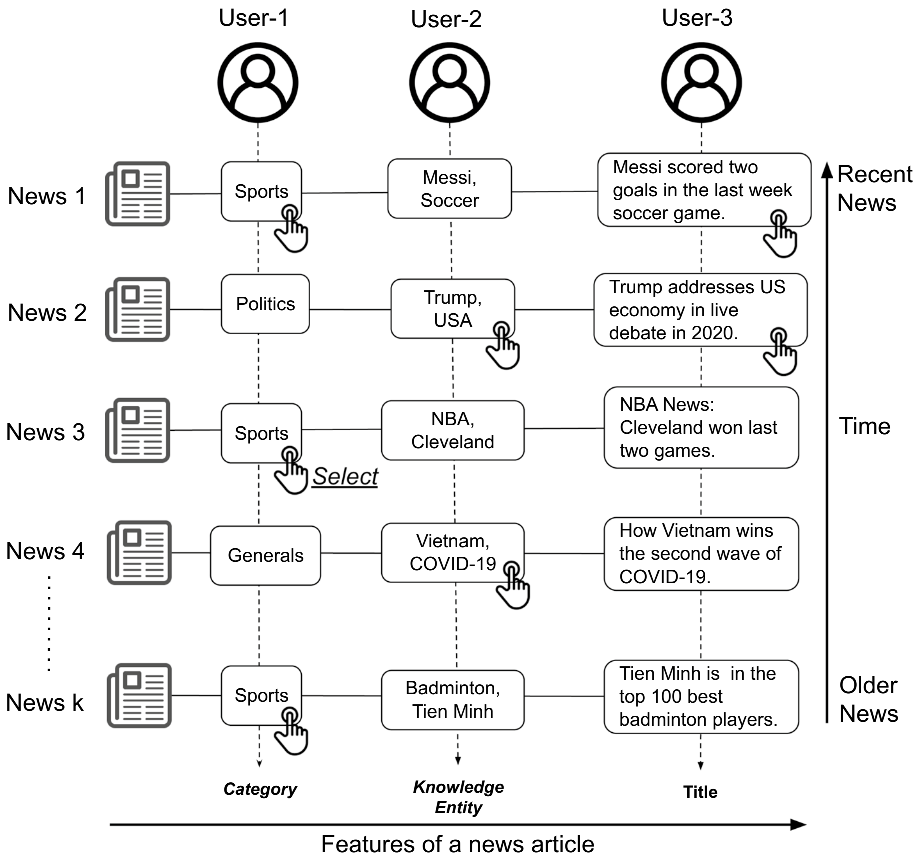
User-1
User-2
User-3
News 1
News 2
News 3
News 4
News k
Sports
Politics
Sports
Generals
Sports
Select
Messi, Soccer
Trump, USA
NBA, Cleveland
Vietnam, COVID-19
Badminton, Tien Minh
Messi scored two goals in the last week soccer game.
Trump addresses US economy in live debate in 2020.
NBA News: Cleveland won last two games.
How Vietnam wins the second wave of COVID-19.
Tien Minh is in the top 100 best badminton players.
Recent News
Time
Older News
Category
Knowledge Entity
Title
Features of a news article

Figure 1 Example of a news recommendation scenario. The horizontal axis shows multi-aspect properties of each news article. The vertical axis shows interactions of each user. Each of them has different choices depending on the properties of the news articles that they pay attention to

by proposing a novel deep learning model named CupMar, which is able to learn both the contextual user profile and the news article content representation. The main components of CupMar consist of i) the News Encoder (NE) and ii) the User-Profile Encoder (UE). The NE infers representation of a news article based on its important properties such as category, title, and abstract content. Self-attention and attention mechanism are used to learn news content effectively. In addition, due to recent successes of using knowledge entity for news recommendation task [7], we also enrich the learning of news representation by adding knowledge entities taken from WikiData knowledge graph[4] to the feature list of the NE. The UE contains two submodules. The first submodule is the *Long-term Latent Preferences Extraction* (LPE), and the second submodule is the *Recent Latent Preferences Extractor* (RPE). We are strongly motivated by the observation that the reading history of a user always encodes both of her long-term as well as her current interests. Thus by using both LPE and RPE submodules, we can learn the representations of the user's contextual

[4] wikidata.org

profile. The encoding news representation from the NE's output and the user-profile representation from the UE's output together are used to calculate an *interaction score*, which helps us to identify highly relevant candidate news items for each user. Hence, the online news services can recommend a ranked list of suitable news articles to their users, thereby improving their recommendation service quality and increasing users' satisfaction and experience.

We perform extensive experiments on the Microsoft News Dataset (MIND) [8], and the results show that our approach improves the performance of news recommendation task. Our source code is also available online for the reproducibility purpose[5]. In a nutshell, the main contributions in this paper are as follows:

- We introduce a novel deep learning model CupMar to solve news recommendation challenge. CupMar leverages its major components NE and UE to learn user and news article representation, and uses the Score Rating component to rank relevant articles and recommend to users.
- We propose two strategies to infer user and news article representation from the CupMar model's main components. The NE component uses multi-aspect properties of a news article and an ensemble of advanced neural network layers to accurately learn a news article representation. The UE component looks at a user's news reading history, and learns her contextual profile including long-term and temporary preferences to derive a user's representation.
- We conduct extensive experiments of the CupMar model on the popular MIND dataset. The CupMar model shows the state-of-the-art performance against all the baselines, thus demonstrating the efficiency of our approach.

This work is an extended version from our previous work accepted in the 22nd International Conference on Web Information Systems Engineering (WISE 2021) [9]. Compared to the previous work, we provide a more in-depth explanation of the enhanced CupMar approach, a thorough discussion of the literature, and additional experiments on new dataset and a detailed elaboration on the evaluation process. We also have made the source code of this research publicly available to the research community. The rest of this paper is organized as follows. In Section 2, we discuss the related works on news recommendation problem. We then first introduce the CupMar model design in Section 3, and describe the technical details of the News Encoder and the User-Profle Encoder in Sections 4 and 5 respectively. The experimentation and evaluation are described in Section 6. Finally, we provide some concluding remarks of our work in Section 7.

2 Related works

News recommendation is a popular and essential task in the field of natural language processing (NLP) and recommender systems [1, 10–12]. A number of online businesses rely heavily onto this task to tailor personalized experience for millions of users [5, 13–15]. The main approach for solving news recommendation problem is to accurately learn the news article and user representation [16]. Henceforth, several popular works rely on different

[5] https://github.com/heroddaji/cupmar

feature engineering strategies to build their own news article and user representation [4, 5, 17–22]. Particularly, Latent Dirichlet Allocation (LDA) is used to generate topic distribution features to infer news representation in each session, and user representation is inferred by all the news in her session [23]. Another noteworthy method is the Explicit Localized Semantic Analysis (ELSA) proposed by Son et al. [24] for location-based news recommendation, where location and topic signals are calculated from Wikipedia posts as news representation. Nevertheless, the downside of manual feature engineering is the dependence on expert domain knowledge, which is not always available for many approaches. Additionally, traditional NLP methods do not incorporate word context and word order well enough to derive semantic meaning and learn user and news article representation effectively [10].

Owning to the popularity of deep learning methods, there have been a lot of efforts during the recent years to address the aforementioned issues pertaining to the task of news recommendations [3, 4, 17]. For instance, the work of Wang et al. [7] tries to infer news item representation from the news title using a knowledge-entity-aware method with Convolutional Neural Network (CNN) layer, and learn user representation by her browsed history news. Another approach in the work of Okura et al. [5] takes advantages of the denoising autoencoder [25] to learn the news article representation. They use this technique with a Gated Recurrent Unit (GRU) neural network layer to learn the user representation from her history records. A recent deep learning approach is proposed by Wu et al. [26], where the attention mechanism is used to attend to word-level and news-level to learn a news article representation, and the embedding ID of the user is used as a vector for the attentions as well.

Additionally, as the recommendation engines are becoming more and more relevant in modern services, researchers also combine different neural approaches to make use several of information signals within the news articles. For instance, the authors of [27] propose to use the knowledge graph to enhance and distil signals from document representation and show improved performance. In another work, the authors of [28] leverage the implicit negative feedback from user interactions based on reading time and clicks, thus resulting in an improvement of the accuracy.

Our proposed CupMar model also takes advantage of deep neural network to solve the news recommendation problem. However, the most prominent features that make our model different with the aforementioned models are:

- The utilization of multi-aspect properties in each news article, where each property is first encoded differently and then merged together to derive the final news representations,
- The combination of both long-term and short-term interactions to infer the user representations. We rely on an ensemble of multiple advanced neural network mechanisms to automatically capture the similarity between a user and a candidate news article representation.

3 The CupMar model

In this section, we briefly introduce the CupMar (Contextual User-Profile and Multi-aspect Articles Representation) model as shown in Figure 2. The CupMar model comprises of two major components. The first component is NE (News Encoder) that uses multiple neural network mechanisms on its multi-aspect properties to learn a news article representation in the

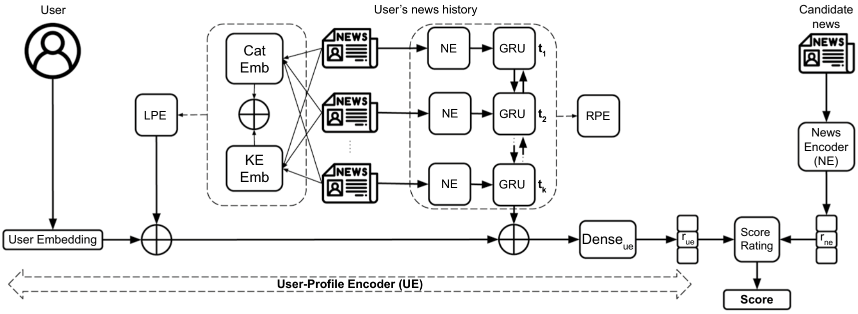

Figure 2 The overall design of the CupMar model. CupMar has two main components. The first component is NE (News Encoder), which is able to learn a news article representation. The second component is UE (User-Profile Encoder), which is able to derive a contextual user representation thanks to its submodules: LPE and RPE. The final interaction score is calculated by the *Score Rating* component

form of a news vector. The second component is UE (User-Profile Encoder) that is further sub-divided into two submodules, which are LPE (Long-term Preferences latent Extractor) and RPE (Recent Preferences latent Extractor). The LPE is responsible for understanding a user's long-term latent preferences, while the RPE is responsible for extracting temporary preferences from a user's reading history. The latent vectors of the LPE and RPE are concatenated to form a contextual user-profile vector. Finally, a *Score Rating* component uses inputs of both the candidate news vector and the contextual user profile vector to predict the interaction score between these two entities.

To accurately train the CupMar for the news recommendation task, there are several things we need to address. First, we need to have a scoring function to measure the interactive score between a user and news article representation. One of the fast and effective methods to gain this requirement is the dot product operation, as applied in the famous work of Okura et al. [5]. Hence, we use the dot product operation to compute the interaction probability inside the final component *Score Rating* of the CupMar model, as illustrated in Figure 2. If we have a user-profile u with its representation vector $\boldsymbol{r_u}$ and a candidate news article n with its representation vector $\boldsymbol{r_n}$, then we can calculate the interaction score between them as $\boldsymbol{s}(u, n) = \boldsymbol{r}_{\boldsymbol{u}}^{\mathsf{T}}\boldsymbol{r}_{\boldsymbol{n}}$.

Secondly, we address our news recommendation problem as a classification task, and use the negative sampling technique during model training [29]. Therefore, when a user is presented with multiple news articles, the articles that are clicked by the user are the positive samples, whereas the other N random sampled articles that are not clicked by the user are the negative samples. Then, the CupMar model can learn to infer the interaction probability between the positive and N negative news articles, thus formulating this as a $N+1$ classes prediction for the classification task. The loss function is the negative log-likelihood of the positive samples. As such, the training total loss of all positive samples is calculated as follows:

$$\text{loss} = -\sum_{i=1}^{P} \log \frac{\exp(\boldsymbol{s}(u, n_i^{pos}))}{\exp(\boldsymbol{s}(u, n_i^{pos}) + \sum_{k=1}^{K} \exp(\boldsymbol{s}(u, n_{i,k}^{neg}))} \tag{1}$$

where P is the amount of positive training samples, n_i^{pos} is the i^{th} positive sample in one news session, and $n_{i,k}^{neg}$ is the k^{th} negative sample for this i^{th} positive sample.

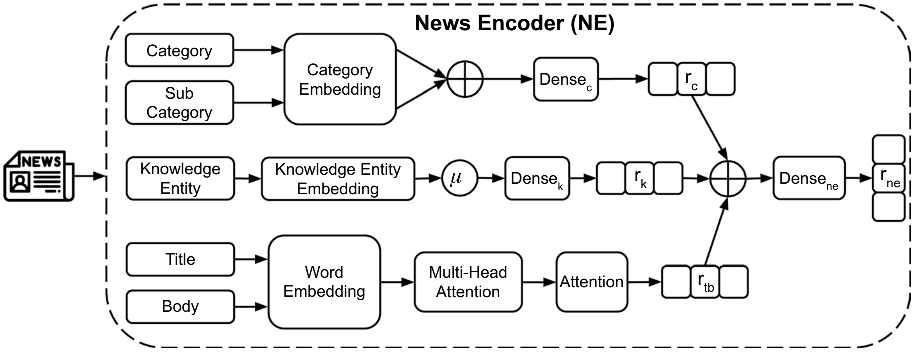

Figure 3 The News Encoder (NE) component design. In this NE component, multi-aspect properties such as the category, knowledge entity, content of the news article are used and processed in different ways via multiple neural network layers. All of these property vectors are concatenated and go through a Dense layer to derive the final representation vector

In the sequel, the technical details of NE and UE will be described in Sections 4 and 5, respectively.

4 Learning the news representation

The task of the CupMar News Encoder (NE) is to learn the representation of a news article. A news article contains several pieces of useful information such as the news category, news title, news body content, and the knowledge entities, as depicted in Figure 3. It is essential to leverage all of these pieces of information to derive a meaningful representation for downstream machine learning tasks. As such, for each news item, we use five main features to encode its representation vector. We denote a news item as $n = \{c,sc,k,t,b\}$, where $c \in C$ is the category feature in the set C of all categories in the dataset, $sc \in C$ is the subcategory feature. We have $k \in K$ as the knowledge entity feature in the set K of all knowledge entities in the dataset. We have t as the news title feature with T words, hence $t = [w^t_1, w^t_2, \dots, w^t_T]$, where $w^t \in W$ is a word in the title t in the set of all distinct words W in the dataset. Similarly, b is the news body content feature with B words, hence $b = [w^b_1, w^b_2, \dots, w^b_B]$ where $w^b \in W$ is a word in the body b.

First, we derive the vector r_c from both the category c and subcategory sc of the news article. The category and subcategory features give us clear information about the topic of the news article, and they also serve as strong signals for a user's long-term preferences. The vector $\boldsymbol{r_c}$ is formulated as follows:

$$\boldsymbol{r_c} = \mathrm{ReLU}(\mathbf{W_c} \times [e_c \parallel e_{sc}] + \boldsymbol{b_c}), \tag{2}$$

where $\mathbf{W_c}$ and $\boldsymbol{b_c}$ are the weight and bias parameters of the $Dense_c$ (feed-forward) layer in Figure 3, the $[e_c \parallel e_{sc}]$ is the concatenation of the category embedding e_c of category c, and subcategory embedding e_{sc} of subcategory sc, and $ReLU$ is the non-linear activation function [30].

Likewise, we perform a similar procedure to learn the vector $\boldsymbol{r_k}$ of the knowledge entity k of the news article. Since one article can contain multiple knowledge entities, we perform the mean operation on their embedding before feeding them into the $Dense_k$ layer as illustrated in Figure 3. The formulation is as follows:

$$\boldsymbol{r_k} = \mathrm{ReLU}(\mathbf{W_k} \times \boldsymbol{\mu}(e_{k_1}, e_{k_2}, \dots, e_{k_n}) + \boldsymbol{b_k}), \tag{3}$$

where $\mathbf{W_k}$ and $\boldsymbol{b_k}$ are the weight and bias parameters of the $Dense_k$ layer, $\boldsymbol{\mu}(e_{k_1}, e_{k_2}, \dots, e_{k_n})$ is the mean operation of n knowledge entity embeddings e_k in the article.

The most important feature of a news article is actually the content itself. We want to learn the representation from both the news article's title and body content. Primarily, we want to know how each word interacts with its surrounded nearby words. Therefore, we choose to apply both the attention and multi-head self-attention mechanisms that is popularized by the work of Vaswani et al. [31]. The formulation to learn the representation $\boldsymbol{r}_{\mathbf{tb}}$ of the news article content in the title and the body is as follows:

$$\boldsymbol{r}_{\mathbf{tb}} = \mathbf{Att}(\mathbf{Heads}(e_{w^t_1}, e_{w^t_2}, \dots, e_{w^t_T}, e_{w^b_1}, e_{w^b_2}, \dots, e_{w^b_B})), \tag{4}$$

where $\mathbf{Heads}(e_{w_1}, \dots, e_{w_i})$ is a word-level multi-head self-attention layer [31] on each word embedding e_{w_i}. This layer contains k heads, which is a hyperparameter. The head $\boldsymbol{h_k}$ learns the representation of word w_i as follows:

$$\boldsymbol{h^w_{i,k}} = \mathbf{V}^w_k \left(\sum_{j=1}^{T+B} \boldsymbol{a^k_{i,j}} e_j \right), \tag{5}$$

$$\boldsymbol{a}^{\boldsymbol{k}_{i,j}} = \frac{\exp(e^{\mathrm{T}}_{w_i} \mathbf{Q}^w_k e_j)}{\sum_{m=1}^{T+B} \exp(e^{\mathrm{T}}_i \mathbf{Q}^w_k e_m)}, \tag{6}$$

where $\mathbf{Q}^w_k$ and $\mathbf{V}^w_k$ are the weight parameters in the $\boldsymbol{h_k}$ head, $(\cdot)^T$ is the transpose operation, $T + B$ is the total amount of words in the title and body, and $\boldsymbol{a^k_{i,j}}$ is the interaction weight between i and j words. The final representation for each word w_i is the concatenation of all the self-attention heads, that is $\boldsymbol{h^w_i} = [\boldsymbol{h^w_{i,1}} \parallel \boldsymbol{h^w_{i,2}} \parallel \dots \parallel \boldsymbol{h^w_{i,h}}]$, hence we have $\mathbf{Heads}(e_{w_1}, \dots, e_{w_i}) = \{\boldsymbol{h^w_1}, \dots, \boldsymbol{h^w_i}\}$. Subsequently, the $\mathbf{Att}$(Heads) function of the attention layer then attends to each word after the self-attention representation $\boldsymbol{h^w_i}$. The formula for deriving the attention weight of each word $\boldsymbol{\alpha^w_i}$ is:

$$\boldsymbol{b^w_i} = \boldsymbol{q}^{\mathrm{T}}_w tanh(\mathbf{V}_w \times \boldsymbol{h^w_i} + \boldsymbol{v}_w), \tag{7}$$

$$\boldsymbol{\alpha^w_i} = \frac{\exp(\boldsymbol{b^w_i})}{\sum_{j=1}^{T+B} \exp(\boldsymbol{b^w_j})}, \tag{8}$$

where $\mathbf{V}_w$ and $\boldsymbol{v}_w$ are the attention weight and bias parameters, $\boldsymbol{q}_w$ is the query vector. After all of these attention weights are calculated, the content vector $\boldsymbol{r}_{\mathbf{tb}}$ of a news article is computed as:

$$\boldsymbol{r}_{\mathbf{tb}} = \sum_{i=1}^{T+B} \boldsymbol{\alpha^w_i} \boldsymbol{h^w_i}. \tag{9}$$

Finally, we combine all of these multi-aspect vectors $\boldsymbol{r}_c$, $\boldsymbol{r}_k$ and $\boldsymbol{r}_{\mathbf{tb}}$ to learn the *multi-aspect news representation* vector $\boldsymbol{r}_{\mathbf{ne}}$ by combining them and let the final $Dense_{ne}$ layer to extract the most prominent patterns of a news article as illustrated in Figure 3, using the following formula:

$$\boldsymbol{r}_{\mathbf{ne}} = \mathrm{ReLU}(\mathbf{W}_{\mathbf{ne}} \times [\boldsymbol{r}_c \parallel \boldsymbol{r}_k \parallel \boldsymbol{r}_{\mathbf{tb}}] + \boldsymbol{b}_{\mathbf{ne}}), \tag{10}$$

where $\mathbf{W}_{\mathbf{ne}}$ and $\boldsymbol{b}_{\mathbf{ne}}$ are the weight and bias parameters of the $Dense_{ne}$ layer, and $[\boldsymbol{r}_c \parallel \boldsymbol{r}_k \parallel \boldsymbol{r}_{\mathbf{tb}}]$ is the concatenation of the multi-aspects vectors from a news article.

5 Learning the user representation

The CupMar User-Profile Encoder (UE) is responsible for learning user representation from their news reading history. Figure 2 shows the complete architecture of the CupMar model, where the left-side portion visualizes the UE component and its submodules. A user's reading habit can exhibit both long-term and recent preferences. To extract both of these signals, the UE uses two of its submodules, Long-term Preferences latent Extractor (LPE) and Recent Preferences latent Extractor (RPE), to handle them. One might think we need the user's complete reading history records for the UE's submodules to do their job. However, we do not do so due to the high computation complexity and low extraction performance. Instead, by sampling the reading history of the last several days, one can infer the long-term preferences of a user by paying attention to her most frequent reading topics. Likewise, it is also feasible to extract her recent interests by paying attention to the news article title, the body, as well as the embedded knowledge entities. That is advantage of this sampling strategy. The following sections dive into the details of each submodule.

5.1 Long-term preferences latent extractor

The sole purpose of LPE is to learn the long-term preferences of a user throughout her history in news reading records. It looks for frequent signals that signify repetitive behaviours of a user. For example, a user keeps reading entertainment news over multiple sessions, which clearly is an indication about her strong preference for enjoying entertainment content.

We argue that in real life, the news genres or topics from a user's history records serve as a strong indication for a user's general and long-term preferences. Additionally, the unique characteristic of a user also refines her choice. For instance, a fan of basketball is more likely to check sports news about the "The National Basketball Association" (NBA) rather than checking for badminton news. Therefore, to mimic those long-term preferences scenarios, we decide to assign each user a unique embedding vector based on her Identification (ID), and calculate the accumulation of the most frequent categories of a user's history news records via their categorical embedding and knowledge entity embedding. The algorithm for extracting a user's long-term latent preferences vector $\boldsymbol{r}_{\mathbf{lpe}}$ is detailed in Algorithm 1.

Algorithm 1 LPE Algorithm.

```
   Input: The history of news reading records of $k$ news $N = \{n_1, n_2, \ldots, n_k\}$, threshold
          $L$ for $L$ most frequently chosen categories, threshold $K$ for $K$ most frequently
          chosen knowledge entities
   Result: The long-term latent preferences vector $r_{\mathbf{lpe}}$
 1 begin
 2     $e_{u_i} \leftarrow$ UserEmbedding$(u_i)$
 3     for $i \leftarrow 1$ to $k$ do
 4         count category of news $n_i$
 5         select top $L$ categories (1)
 6     end
 7     for $i \leftarrow 1$ to $k$ do
 8         count knowledge entity of news $n_i$
 9         select top $K$ knowledge entities (2)
10     end
11     $C = \{c_1, c_2, \ldots, c_L\} \leftarrow$ Set $C$ of $L$ categories from (1)
12     $D = \{d_1, d_2, \ldots, d_K\} \leftarrow$ Set $D$ of $K$ categories from (2)
13     count $\leftarrow 0$
14     $r_{\mathbf{lpe}} \leftarrow \{\vec{0}\}$
15     for $i \leftarrow 1$ to $k$ do
16         if $n_i$ has category in Cand knowledge entity in $D$ then
17             $r_{\mathbf{lpe}} \leftarrow r_{\mathbf{lpe}} +$ CateEmbedding$(n_i) +$ KnowledgeEntityEmbedding$(n_i)$
18             count $\leftarrow$ count $+ 1$
19         end
20     end
21     $r_{\mathbf{lpe}} \leftarrow r_{\mathbf{lpe}}/$count
22     $r_{\mathbf{lpe}} \leftarrow [r_{\mathbf{lpe}} \parallel e_{u_i}]$
23     return $r_{\mathbf{lpe}}$
24 end
```

First, we initialize each user with a unique user embedding vector $\boldsymbol{e}_{u_i}$ using a *UserEmbedding* layer (line 2). Second, we learn the most frequent L categories inside a user history records and store them in the set C (lines 3 to 7). Third, we initially set the long-term latent preferences vector $\boldsymbol{r}_{\mathbf{lpe}}$ as a zero-vector, then accumulate $\boldsymbol{r}_{\mathbf{lpe}}$ with all of the summation of category embedding and knowledge entity embedding vectors that belong in the set C using the category embedding layer *CateEmbedding* and set D using the knowledge entity embedding layer, respectively (lines 8 to 15). Then, we average the $\boldsymbol{r}_{\mathbf{lpe}}$ based on the *count* value, where it counts the total news articles that has the category in set C (line 16). Finally, we concatenate $\boldsymbol{r}_{\mathbf{lpe}}$ with the user embedding $\boldsymbol{e}_{u_i}$ (line 17). Using this algorithm, we can extract both user's long-term preferences and her unique characteristic into the representative vector $\boldsymbol{r}_{\mathbf{lpe}}$.

5.2 Recent preferences latent extractor

RPE learns recent preferences of a user via the Gated Recurrent Unit (GRU) neural network layer. We denote Z as a variable for the amount of news articles a user has recently read, then for Z news articles in chronological order, the set of news records is denoted as $N = \{n_1, n_2, \dots, n_Z\}$. The RPE derives the recent preference latent vector $\boldsymbol{r}_{\mathbf{rpe}}$ from a user using the GRU layer as follows:

$$z_t = \sigma(W_z[h_{t-1}, \boldsymbol{NE}(n_t)]), \tag{11}$$

$$\boldsymbol{r_t} = \sigma(W_r[h_{t-1}, \boldsymbol{NE}(n_t)]), \tag{12}$$

$$\widetilde{\boldsymbol{h}}_{\boldsymbol{t}} = tanh(W_{\widetilde{h}}[r_t \odot h_{t-1}, \boldsymbol{NE}(n_t)]), \tag{13}$$

$$\boldsymbol{h_t} = z_t \odot \boldsymbol{h_t} + (1 - z_t) \odot \widetilde{\boldsymbol{h}}_{\boldsymbol{t}}, \tag{14}$$

where σ is the sigmoid function, $\odot$ is the item-wise product, W_r, W_z, and $W_{\widetilde{h}}$ are the GRU's network weights, $\boldsymbol{NE}(.)$ is the News Encoder function described in Section 4. With the initial hidden vector state $\boldsymbol{h_0}$ initialized as a zero-vector, we repeat the process with the GRU network until we reach the last hidden state vector $\boldsymbol{h_Z}$. Thus, the RPE vector is $\boldsymbol{r}_{\mathbf{rpe}} = \boldsymbol{h_Z}$.

5.3 The representation of contextual user-profile

Given the two contextual vectors of a user, which are the long-term latent preferences vector $\boldsymbol{r}_{\mathbf{lpe}}$ and the recent latent preferences vector $\boldsymbol{r}_{\mathbf{rpe}}$, the final contextual user-profile vector $\boldsymbol{r}_{\mathbf{ue}}$, is calculated as follows:

$$\boldsymbol{r}_{\mathbf{ue}} = \text{ReLU}(\mathbf{W}_{\mathbf{ue}} \times [\boldsymbol{r}_{\mathbf{lpe}} \parallel \boldsymbol{r}_{\mathbf{rpe}}] + \boldsymbol{b}_{\mathbf{ue}}), \tag{15}$$

where $\mathbf{W}_{\mathbf{ue}}$ and $\boldsymbol{b}_{\mathbf{ue}}$ are the weight and bias parameters of the $Dense_{ue}$ layer (Illustrated in Figure 2), and $[\boldsymbol{r}_{\mathbf{lpe}} \parallel \boldsymbol{r}_{\mathbf{rpe}}]$ is the concatenation of both contextual vectors that we learn from the aforementioned sections. The usage of both contextual vectors is the key ingredient to help the CupMar model achieving better scores as we discuss in the later sections.

6 Evaluations of CupMar

In this section, we describe the evaluation processes and the detailed performance analysis about internal components of the proposed CupMar model against several baselines.

6.1 Experimental dataset

There has been a shortage of quality datasets for news recommendation research. Fortunately, the recent work of Wu et al. [8] introduces a large-scale MIND dataset, which can serve as a benchmark dataset for news recommendation. We conduct all our experiments on this high-quality dataset. MIND is collected from the user's behaviour logs of Microsoft News website[6]. It contains more than 150,000 news articles, and more than 15 million behaviour logs that are generated by one million users. Each news item comes with rich textual attributes such as the category, subcategory, title, body and knowledge entities

[6] microsoftnews.msn.com

Table 1 MIND dataset statistics [8]

MIND dataset information	
Number of news articles	161,013
Number of category	20
Number of knowledge entity	3,299,687
Number of users	1,000,000
Total news sessions	15,777,377
Total click behaviours	24,155,470
Average title length	11.52
Average abstract length	43.00

embedded inside. Additionally, the MIND dataset also comes with a smaller version called MIND-small, which is suitable for quick prototyping and validation. MIND-small accounts for 5% data of the total dataset. Henceforth, the research community has quickly adapted to use MIND as a robust benchmarking dataset for news recommendation, as shown in [3, 4, 17, 32]. We run our evaluation on the both sizes of the MIND dataset. Table 1 summarizes the statistics of the MIND dataset.

Before the training, we perform preprocessing steps to align the MIND dataset into an appropriate format for CupMar to train. We first convert all words, categories and knowledge entities into integer numbers for embedding purpose. Then for each news session, we choose one positive sample and four random negative samples, and repeat this process five times. Hence, for every log session in the dataset, we generate five training log sessions, resulting in an even larger training dataset for the CupMar model. This helps us to have a balance ratio of correct positive and negative pairs of input signals, and improve the model accuracy.

6.2 Experimental environment

For evaluation, we apply the same settings for different variations of our CupMar model. The categorical embedding dimension is 100 for the category, subcategory and knowledge entity features. We also use the popular pre-trained word embedding FastText [33] (it should be noted that we used Glove word embedding [34] in our conference paper), with the embedding dimension of 300. We use dropout with a drop rate of 30% to prevent overfitting the model. The Adam optimizer [35] is used to optimize the network. The batch size is 32 and the learning rate is 0.001. We also select four negative samples for each positive sample to emulate a classification task of 5 classes as mentioned in previous sections to be compatible for comparison with the training methods outlined in the works of [3, 4]. We choose the amount of self-attention head to be 10, and each head has a dimension of 10; thus the total dimension of all heads for each word vector is 100.

For the evaluation metrics, we use ranking metrics to benchmark the performance of the validation models. The ones that we choose are the Area Under ROC curve (AUC), Mean Reciprocal Rank (MRR), and the Normalized Discounted Cumulative Gain (nDCG). Each model is evaluated five times and the average scores are reported. We also report our scores on both the MIND and MIND-small datasets.

6.3 Baseline models

Our CupMar model is evaluated against the following baseline models:

- Factorization Machines (FM) [36]: FM is a state-of-the-art model for many recommendation problems based on matrix factorization approach. In our evaluation, we define the user representation as the combination of all TF_IDF signals extracted from the title and the body of a user's history news. The news article representation includes the TF_IDF features from its title and body, the one-hot encoding of its category and subcategory. Finally, the input into the FM model is the concatenation of both user and candidate news article representation.
- CNN [37]: We adopt the CNN model proposed by Kim as one of the baselines. It uses max pooling on the text to learn news article representation from the title and body.
- DKN [7]: Deep Knowledge-Aware Network for News Recommendation is a deep learning model that leverages CNN and knowledge entity awareness attention on the news article to derive the user and news representation.
- HiFiArk [4]: High-Fidelity Archive Network is another robust deep learning model for news recommendation task. It treats user's news reading history as a compact vector and store them into archives during offline stage. Then during the online stage, these compact vectors are used to infer user interest upon candidate news.
- NRMS [3]: Neural News recommendation with Multi-head Self-attention is a recent deep news recommendation model, which uses its news encoder with multi-head self-attention to learn words interaction, and attention mechanism on the user encoder to extract user preferences.
- CUPCate: CUPCate is a simple variant of our CupMar model. In this model, we only consider the category and subcategory features in the news encoder. The user-profile then is encoded by averaging all the history news records representation. We develop this as a simple baseline during our experimentation.
- CUPShort: CUPShort is another variant of the CupMar model. It is identical to CupMar, but without the LPE submodule. CUPShort only learns to extract the recent preferences of a user via its GRU layer. By using CUPShort, we can compare the effectiveness of the CupMar model when we employ the LPE submodule.
- CUPLong: CUPLong is another variant of the CupMar model. It is identical to CupMar, but without the RPE submodule. CUPLong only learns to extract the long-term preferences of a user.

6.4 Evaluation results

By training and evaluating the CupMar model on the MIND and MIND-small datasets, we obtain the performance results shown in Table 2. It is interesting to see that we achieve the state-of-the-art scores on the MIND dataset, but fail to have that position on the MIND-small dataset, and that will be explained together with other observations in detail.

First, our CupMar model achieves the state-of-the-art scores and outperforms all the baseline methods on the MIND dataset. CupMar's performance is followed closely behind by NRMS and HiFiArk, which are the two strongly performing models for news recommendation. This result has proven that using multi-aspect properties for news encoding and leveraging contextual user profile signals can significantly boost the learning capability

Table 2 Performance comparison of the CupMar model with other methods over the MIND and MIND-small datasets

Methods	NDCG@5	NDCG@10	AUC	MRR
MIND				
FM	0.2701	0.3415	0.5661	0.2416
CNN	0.3224	0.3552	0.5907	0.3071
DKN	0.2833	0.3475	0.5937	0.2614
HiFiArk	0.3105	0.3662	0.6378	0.2749
NRMS	0.3298	0.3954	0.6491	0.2999
CUPCate	0.2586	0.3005	0.5601	0.2312
CUPShort	0.3290	0.3913	0.6428	0.2954
CUPLong	0.2965	0.3124	0.6041	0.2678
CupMar	**0.3435**	**0.4084**	**0.6582**	**0.3123**
MIND-small				
FM	0.2854	0.3546	0.5845	0.2785
CNN	0.3025	0.3725	0.6178	0.2987
DKN	0.3262	0.3895	0.6354	0.3142
HiFiArk	0.3539	**0.4212**	0.6660	0.3287
NRMS	**0.3587**	0.4160	**0.6664**	**0.3321**
CUPCate	0.3276	0.3867	0.6239	0.2964
CUPShort	0.3272	0.3894	0.6396	0.2965
CUPLong	0.3172	0.3694	0.6137	0.2844
CupMar	0.3289	0.3902	0.6415	0.2961

of the deep learning model for news recommendation task. Moreover, our CupMar model also achieves slightly better performance from having the LPE submodule compared to the CUPShort model which only employs the RPE submodule, as can be seen by the small gap in the scores between them. We will give detailed analysis of this point in the later sections.

Second, the deep neural network models clearly show superior performance in comparison to the matrix factorization FM approach. The better performance of neural network models can be explained by their high learning capacity. Due to the high amount of weight parameters, neural network models have the ability to tackle the complicated task of news recommendation. Another evident supporting this statement is the ranking scores of our simple CUPCate model, which has the lowest scores across all metrics in the MIND dataset. The most likely reason is the low number of parameters it has due to the crude design of only using two categorical features and one dense layer.

Third, we observe an interesting phenomenon. Our CupMar model does not perform well when it is trained and evaluated on the MIND-small dataset, which contains about 5% total samples of the MIND dataset. The CupMar model ranks in the third place, while the top spots belong to NRMS and HiFiArk models. After careful examination, we believe that due to the usage of multi-aspect properties and multiple advanced neural network layers such as self-attention heads, attention layer, GRU layer and several Dense layers in the whole model, the number of the weight parameters in the CupMar model increases significantly. We have 40% more weight parameters in comparison to our implemented NRMS model. Although the high number of parameters helps CupMar to make better generalization over large datasets, it is underfit when being trained on smaller datasets. This is a little setback we want to improve in the future work, we want the lower bound for total samples should be 10% total samples of the MIND dataset for a good performance model training.

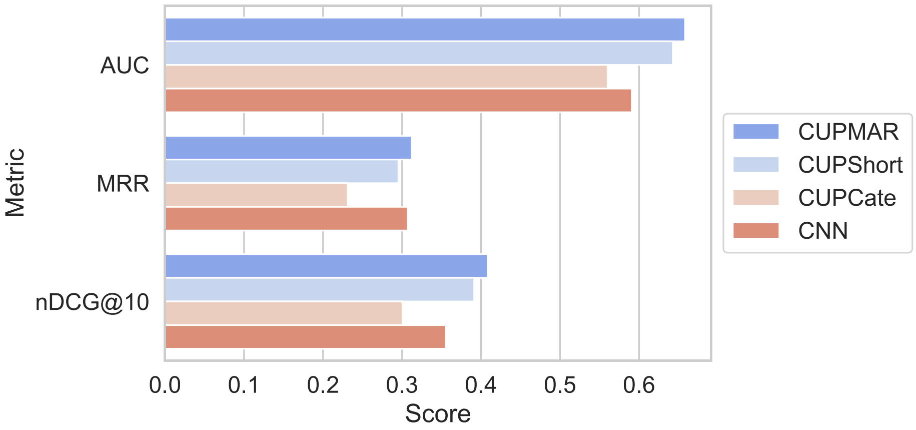

Figure 4 Evaluation results to show the analysis of LPE submodule in comparison to other methods

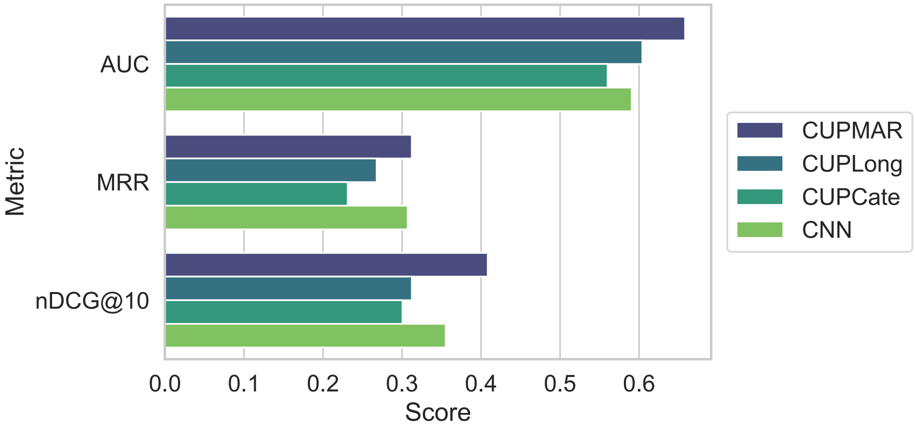

Figure 5 Evaluation results to show the analysis of RPE submodule in comparison to other methods

6.5 Detailed analysis on contextual user-profile

In this section, we analyze our CupMar model's performance concerning the use of contextual information, which is handled by the LPE and RPE submodules. We create two variant models, called CUPShort and CUPLong, respectively. The CUPShort model only uses the RPE submodule inside the UE component to tackle the task, while the CUPLong model only uses the LPE submodule. Then we compare the inference scores of each of them to other models to see the changes in the performance. In particular, we compare with CupMar as the full model, CUPCate as the simple baseline, and CNN as a neural network model with high learning capacity for text representation. According to the results shown in Figures 4 and 5, we can see that leveraging both the long-term and recent-term contexts can strongly boost the performance of the CupMar model. CupMar always has higher scores than both CUPShort and CUPLong across all three different

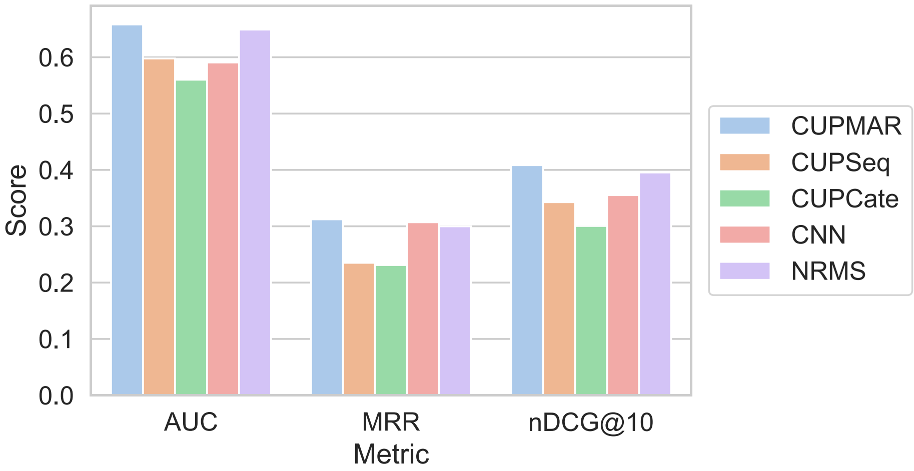

Figure 6 Evaluation results to show the analysis of using multi-aspect properties to encode a news representation

metrics. This clearly shows the effectiveness of the contextual information of a user in the news recommendation task.

We also want to answer a further question: *which user contextual aspect contributes more to the CupMar model*. Hence, by looking at the percentage gap in their respective scores to the CupMar model, we can confirm that the RPE submodule contributes more to the performance of the CupMar model. CUPShort scores (sum of all metrics) lower than CupMar by only 3.6%, while the CUPLong's scores witness a gap of 14%. This result shows that the recent-term preferences contribute more to the user representation than the long-term preferences, which does make sense since a user's recent preferences usually also include her long-term preferences as well.

6.6 Detailed analysis on multi-aspect properties

In this section, we further run evaluations to compare the effectiveness of using multi-aspects properties in the News Encoder (NE) with other approaches. Similar to the analysis of the user contexts, we deploy a model variant called CUPSeq, where instead of using self-attention mechanism and categorical features, we use Seq2Seq [38] architecture with recurrent neural network (RNN) on the news title and body to infer the r_{ne} vector, as explained in Section 4. We then compare the evaluation scores of CUPSeq with other approaches, including the full CupMar model, the NRMS model with self-attention layer, the CNN model for its convolution operation on text data, and the baseline CUPCate with only categorical features. The experimental results are depicted in Figure 6.

At first glance, we can see that using advanced neural network mechanisms such as Seq2Seq or self-attention layer outperforms the simple baseline using categorical features, since the CupMar, CUPSeq, CNN and NRMS advanced models all score significantly higher than the CUPCate model. Especially, this also signifies that the body of text of a news article contributes more information to the neural models than other signals since both CUPSeq and CNN only employ textual data of the title and body of a news article. Additionally, we also understand that more sophisticated architecture such as self-attention layer can learn more effectively than older approaches such as RNN and CNN, because NRMS model achieves better scores than CUPSeq and CNN. Nevertheless, we do see the benefits of using multi-aspect properties in the CupMar model, as CupMar outscores all other models, albeit just a little better than the NRMS model. This demonstrates the strong performance of the proposed CupMar model.

7 Conclusion

In this paper, we propose a novel deep neural network called CupMar for the challenging task of news recommendation. Making personalized recommendations from news articles requires the understanding of both the textual information of a news item, and the user contexts in terms of long-term and recent preferences via the user's history records. To resolve those issues, at the heart of our proposed CupMar model are the News Encoder and User-Profile Encoder. More specifically, the News Encoder learns news article representation from various features such as the category, subcategory, knowledge entities inside the article, the article title and news body. It uses self-attention, attention and dense layers to effectively combine all the necessary signals to represent a news article. On the other hand, the User-Profile Encoder uses the user's recent historical news data with dense textual information to infer both long-term and recent-term signals for the user representation, thanks to the two submodules, the Long-term Preferences latent Extractor and the Recent Preferences latent Extractor with GRU network layer. We perform extensive evaluation of the CupMar model on the popular MIND dataset, and CupMar shows a better performance against all the baselines.

For the future work, we plan to enhance the CupMar model in the recommendation serendipity. We plan to develop a new interaction score based on both the click-probability and the diversification of the candidate news items when compared to that user's historical news reading data. This can help the model to suggest a more diversified news list to its users and increase exploration as well as satisfaction.

Acknowledgements Quan Z. Sheng's work has been partially supported by Australian Research Council Discovery Grants DP200102298 and DP210101723.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. ACM Computing Surveys 54(7) (2022)
2. Tran, D.H., Aljubairy, A., Zaib, M., Sheng, Q.Z., Zhang, W.E., Tran, N.H., Nguyen, K.L.D.: HeteGraph: a convolutional framework for graph learning in recommender systems. In: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020, pp. 1–8 (2020)
3. Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 6388–6393 (2019)
4. Liu, Z., Xing, Y., Wu, F., An, M., Xie, X.: Hi-fi ark: Deep user representation via high-fidelity archive network. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 3059–3065 (2019)
5. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, vol. 0, pp. 1933–1942 (2017)
6. Chen, R., Hua, Q., Chang, Y., Wang, B., Zhang, L., Kong, X.: A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE Access **6**, 64301–64320 (2018)
7. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pp. 1835–1844 (2018)
8. Wu, F., Qiao, Y., Chen, J., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., Zhou, M.: MIND: A large-scale dataset for news recommendation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pp. 3597–3606 (2020)
9. Tran, D.H., Hamad, S.A., Zaib, M., Aljubairy, A., Sheng, Q.Z., Zhang, W.E., Tran, N.H., Khoa, N.L.D.: Deep news recommendation with contextual user profiling and multifaceted article representation. In: Zhang, W., Zou, L., Maamar, Z., Chen, L. (eds.) Web Information Systems Engineering - WISE 2021 - 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26-29, 2021, Proceedings, Part II, pp. 237–251 (2021)
10. Tran, D.H., Sheng, Q.Z., Zhang, W.E., Hamad, S.A., Khoa, N.L.D., Tran, N.H.: Challenges and opportunities on deep conversational recommender systems. IEEE Computer (2022, in press) (2022)
11. Xiao, Y., Yao, L., Pei, Q., Wang, X., Yang, J., Sheng, Q.Z.: MGNN: Mutualistic graph neural network for joint friend and item recommendation. IEEE Intell. Syst. **35**(5), 7–17 (2020)
12. Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q.Z., Orgun, M.A., Cao, L., Ricci, F., Yu, P.S.: Graph learning based recommender systems: a review. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pp. 4644–4652 (2021)
13. Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pp. 271–280 (2007)
14. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q.Z., Orgun, M.A.: Sequential recommender systems: challenges, progress and prospects. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 6332–6338 (2019)
15. Tran, D.H., Sheng, Q.Z., Zhang, W.E., Aljubairy, A., Zaib, M., Hamad, S.A., Tran, N.H., Khoa, N.L.D.: HeteGraph: Graph Learning in Recommender Systems via Graph Convolutional Networks. Neural Computing and Applications (2022, in press) (2022)
16. Shah, D., Koneru, P., Shah, P., Parimi, R.: News recommendations at scale at bloomberg media: Challenges and approaches. In: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, p. 369 (2016)
17. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: Neural news recommendation with attentive multi-view learning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 3863–3869 (2019)
18. Phelan, O., McCarthy, K., Smyth, B.: Using twitter to recommend real-time topical news. In: Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009, pp. 385–388 (2009)

19. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI 2010, Hong Kong, China, February 7-10, 2010, pp. 31–40 (2010)
20. Li, L., Chu, W., Langford, J., Schapire, R.E.: 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, pp. 661–670 (2010)
21. Bansal, T., Das, M.K., Bhattacharyya, C.: Content driven user profiling for comment-worthy recommendations of news and blog articles. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015, pp. 195–202 (2015)
22. Lian, J., Zhang, F., Xie, X., Sun, G.: Towards better representation learning for personalized news recommendation: a multi-channel deep fusion approach. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pp. 3805–3811 (2018)
23. Li, L., Zheng, L., Yang, F., Li, T.: Modeling and broadening temporal user interest in personalized news recommendation, vol. 41, pp 3168–3177 (2014)
24. Son, J.W., Kim, A.-, Park, S.: A location-based news article recommendation with explicit localized semantic analysis. In: The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’13, Dublin, Ireland - July 28 - August 01, 2013, pp. 293–302 (2013)
25. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, vol. 11, pp 3371–3408 (2010)
26. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, pp 2576–2584 (2019)
27. Liu, D., Lian, J., Wang, S., Qiao, Y., Chen, J.-H., Sun, G., Xie, X.: KRED: Knowledge-aware document representation for news recommendations, New York, NY, USA. pp 200–209 (2020)
28. Wu, C., Wu, F., Huang, Y., Xie, X.: Neural news recommendation with negative feedback. CCF Trans. Pervasive Comput. Interaction **2**(3), 178–188 (2020)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp 3111–3119 (2013)
30. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, pp 807–814 (2010)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp 5998–6008 (2017)
32. An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., Xie, X.: Neural news recommendation with long- and short-term user representations. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp 336–345 (2019)
33. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017)
34. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL, pp 1532–1543 (2014)
35. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
36. Rendle, S.: Factorization Machines with libFM. ACM Transactions on Intelligent Systems and Technology 3(3) (2012)
37. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL, pp 1746–1751 (2014)

38. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp 3104–3112 (2014)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Dai Hoang Tran[1] · **Quan Z. Sheng**[1] · **Wei Emma Zhang**[2] · **Nguyen H. Tran**[3] · **Nguyen Lu Dang Khoa**[4]

✉ Dai Hoang Tran
dai.tran@mq.edu.au

✉ Quan Z. Sheng
michael.sheng@mq.edu.au

Wei Emma Zhang
wei.e.zhang@adelaide.edu.au

Nguyen H. Tran
nguyen.tran@sydney.edu.au

Nguyen Lu Dang Khoa
khoa.nguyen@data61.csiro.au

[1] School of Computing, Macquarie University, Sydney, NSW 2109, Australia

[2] School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia

[3] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia

[4] Data61, CSIRO, Sydney, Australia