DAN: Deep Attention Neural Network for News Recommendation

Qiannan Zhu,[1,2] Xiaofei Zhou,*[1,2] Zeliang Song,[1,2] Jianlong Tan,[1,2] Li Guo[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
zhuqiannan@iie.ac.cn, zhouxiaofei@iie.ac.cn

Abstract

With the rapid information explosion of news, making personalized news recommendation for users becomes an increasingly challenging problem. Many existing recommendation methods that regard the recommendation procedure as the static process, have achieved better recommendation performance. However, they usually fail with the dynamic diversity of news and user's interests, or ignore the importance of sequential information of user's clicking selection. In this paper, taking full advantages of convolution neural network (CNN), recurrent neural network (RNN) and attention mechanism, we propose a deep attention neural network DAN for news recommendation. Our DAN model presents to use attention-based parallel CNN for aggregating user's interest features and attention-based RNN for capturing richer hidden sequential features of user's clicks, and combines these features for new recommendation. We conduct experiment on real-world news data sets, and the experimental results demonstrate the superiority and effectiveness of our proposed DAN model.

Introduction

As the amount of online content and services increases, users are inundated with tremendous choices and options. Therefore recommender systems that automatically recommend a small set of items for satisfying user's preferences, have growingly attracted attentions in both industry and academic. Different from other items (movies, music and books), news provided by online news web sites, such as Google News and Bing News can be overwhelming to users. Facing the overloading information, targeting users' reading interests and recommending the most interesting news to users are necessary (Bansal, Das, and Bhattacharyya 2015; Li et al. 2010).

There is a wide variety of typical methods to make personalized news recommendations, including collaborative filtering (CF) based methods (Das et al. 2007; Xue et al. 2017), content based methods (IJntema et al. 2010; Huang et al. 2013), hybrid methods (Morales, Gionis, and Lucchese 2012; Li et al. 2011). These methods usually have the cold start problem when being exposed to the sparsity of user-item interactions. With the high time-sensitivity and dynamic diversity of news, they also have difficulties in reflecting a user's interests in real time. Recently deep learning based methods (Zheng, Noroozi, and Yu 2017; Wang et al. 2017) have better capability of modeling complex user-item (i.e., news) interactions, and capturing the dynamic properties of news and users. Among such methods, DKN (Hongwei Wang 2018) has achieved state-of-the-art recommendation performance by introducing an attention mechanism to model the diversity of news and user's interests. However, deep learning based models still have two major disadvantages. First, they usually consider news title as features for recommendation, and ignore the news profile which has more influence on user's potential interests than the news title does. Second, many models such as DKN only depend on user' current potential interests on the current candidate news for making recommendation, and can not consider the influence of sequential information of a user's clicked news. The sequential information of user's history reading can better reflect the changes and diversity of user' interests within a period of time.

For addressing above issues, this paper proposes a deep attention neural network DAN for news recommendation. For learning news feature representations, DAN devises a PCNN component composed of two parallel convolutional neural networks to fuse the title-level and profile-level information of news. For learning user feature representation, DAN not only introduces an ANN component to model a user's current interest, but also designs an ARNN component to capture the potential sequential features of a user's history readings. ARNN component is an attention-based recurrent neural network that enforces the attention mechanism on all states of RNN for getting richer sequential feature at different clicking time. Based on the outputs of PCNN, ANN component inspired by DKN, uses an attention mechanism to capture different impacts of the user's clicked news on the candidate news for modeling user's current interest. Finally the user's history sequential features and user's current interest together determine the user feature representation. DAN considers that whether the user clicks on the candidate news according to the similarity between the user feature representation and the candidate news representation.

Our contributions. We propose a deep attention neural network DAN for personalized news recommendation,

*Corresponding Author

which can better capture the user features for targeting user's reading preferences and making news recommendation. Three components PCNN, ANN and ARNN are included in our DAN model.

- PCNN as news representation extractor can learn news feature representation from title-level and profile-level of news.
- ARNN as sequential informatin extractor can automatically get a dynamic history sequential feature representation of a user' clicked news.
- ANN as user interest extractor can extract user's current interest with respect to the candidate news.

To evaluate the effectiveness of our DAN model, we conduct experiments on real-world news data sets. The experimental results show that DAN can significantly outperform state-of-the-art typical models.

Problem Formulation

The recommendation problem in this paper is to use the given users' click history to predict whether user will click a candidate news that he has not seen before. Assume that the history click news of a user are $\{x_1, x_2, ..., x_{t-1}\}$, where x_j is the j-th news that the user clicked. Given the candidate news as x_t, the recommendation system takes $\{x_1, x_2, ..., x_{t-1}\}$ as input and needs to calculate the probability of the user clicking the candidate news x_t.

Notations: This paper considers the title and profile of news as features, and describes the news x as $x = \{T, E, G\}$, where T is the title of news, E is the entity set extracted from the news article, G is the entity-types set of entities in E. Each news title T consists of a sequence of words(entity), i.e., $T = \{w_1, w_2, ..., w_n\}$. Each entity set E and entity-type set G respectively contain a large number of entities and corresponding entity-types, which are defined as $E = \{e_1, e_2, ..., e_m\}$ and $G = \{g_1, g_2, ..., g_m\}$. We denote title embeddings of news as $\mathbf{T} = [\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_n]^T \in R^{n\times d}$, entity set embeddings as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_m]^T \in R^{m\times d}$ and entity-type set embeddings as $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, ...\mathbf{g}_m]^T \in R^{m\times k}$. $\mathbf{w}$, $\mathbf{e}$ and $\mathbf{g}$ are respectively the embedding vectors of word w, entity e and entity-type g. n and m are separately the number of words(entities) in title and the entities extracted from news article. d and k are the dimension of word(entity) and entity-type embedding. These embeddings can be pre-learned from a large corpus or randomly initialized.

Preliminaries

Recently a deep learning based model DKN (Hongwei Wang 2018) can be seen for news recommendation, which takes one candidate news and a user's click history as input, and outputs the probability of user clicking on the candidate news.

As Figure 1 shown, given the user's click history $\{x_1, x_2, ..., x_{t-1}\}$, DKN first learns feature representation $e(x_j)$ for each input news x_j, then aggregates the feature representations by different weights $w_{j,t}$ provided by an attention mechanism, i.e., $\mathbf{e} = \sum_j^{t-1} w_{j,t} \cdot e(x_j)$, where $w_{j,t}$

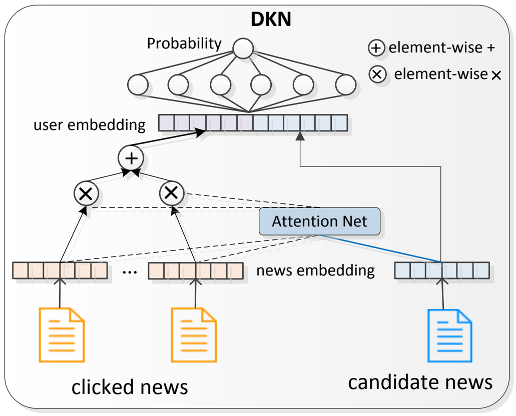

Figure 1: Simple visualization of DKN model.

are from the match of candidate news x_t to clicked news x_j, and $w_{j,t} = softmax(\mathtt{G}[e(x_j), e(x_t)])$, $\mathtt{G}$ is a deep neural network. Finally the similarity of user's embedding $\mathbf{e}$ and candidate news embedding $e(x_t)$ are measured by another DNN $\mathtt{H}$, which determines the clicking probability $P = \mathtt{H}(\mathbf{e}, e(x_t))$.

DAN Framework

The framework of DAN is illustrated in Figure 2. As shown in Figure 2, (1) firstly the embeddings of a user's clicked history news are all input into news representation extractor PCNN. PCNN is a combination of two parallel CNNs that fuse the title-level and profile-level representation of news. (2) Based on the news feature representations by PCNN, user interest extractor ANN uses an attention mechanism to match the candidate news to each clicked news, and aggregate the feature representations of user's current interest. (3) Still for the output news feature representations of PCNN, sequential information extractor ARNN, an attention-based RNN, enforces the attention mechanism on each repeated module of RNN, in order to capture sequential features at different clicking time. (4) We concatenate entire sequential features and users current interest into a fully-connected neural network to get the final user feature representation. For a given candidate news, we also extract its feature representation by PCNN, and then calculate the similarity of it to the user feature representation.

PCNN:News Representation Extractor

PCNN as news representation extractor is composed of two parallel convolutional neural networks, which respectively take the title and profile of news as inputs and learn the title-level and profile-level representation of news. The concatenation of such two representations is regarded as the final feature representation of news.

Following above notations, we align and stack the entity embeddings $\{\mathbf{e}_j\}$ and their corresponding entity-type embeddings $\{\mathbf{g}_j\}$ to define the *profile* embedding of news as follows:

$$\mathbf{C} = [\mathbf{e}_1, f(\mathbf{g}_1), \mathbf{e}_2, f(\mathbf{g}_2)...\mathbf{e}_m, f(\mathbf{g}_m)]^T$$

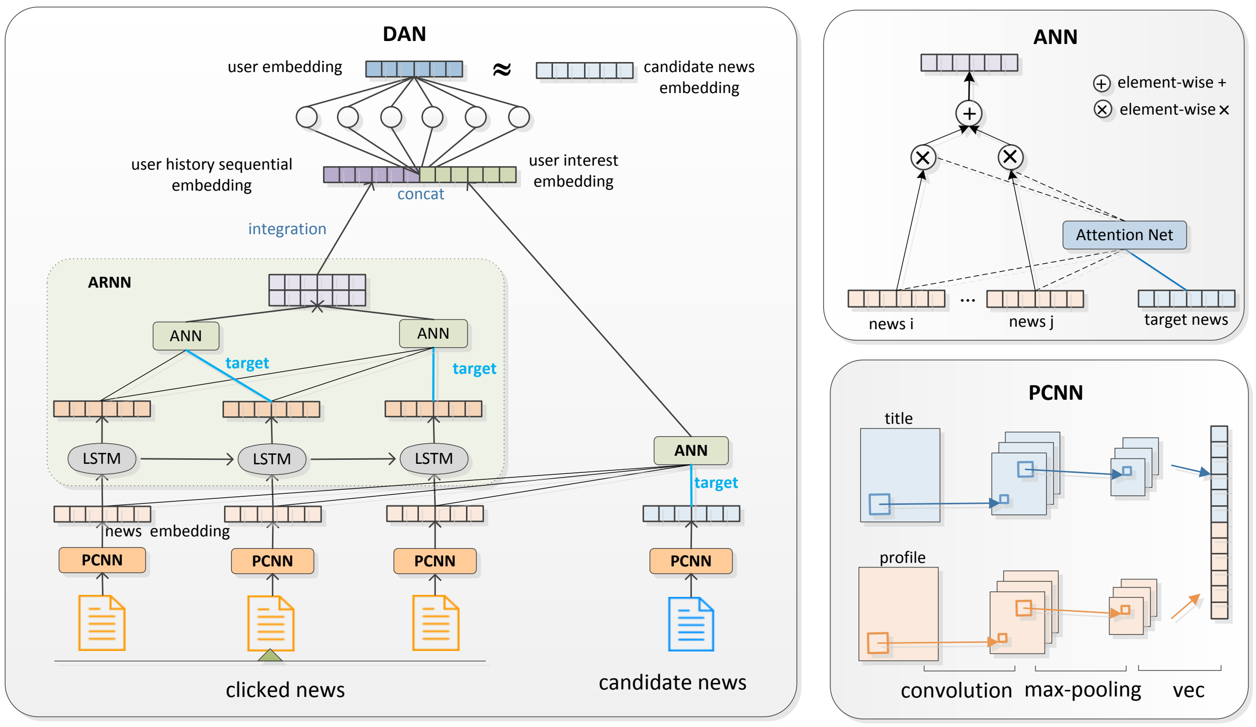

Figure 2: Simple visualization of DAN model.

where $\mathbf{C} \in R^{2m \times d}$, $f(\mathbf{g})$ is the transformation function, which is either $f(\mathbf{g}) = \mathbf{M}_t\, \mathbf{g}$ or $f(\mathbf{g}) = \sigma(\mathbf{M}_t\, \mathbf{g}{+}\mathbf{b})$, where $\mathbf{M}_t \in R^{d \times k}$ is the trainable transformation matrix, $\mathbf{b} \in R^d$ is the trainable bias, and $\sigma = sigmoid$ or $\sigma = tanh$.

PCNN processes title embedding matrix $\mathbf{T}$ and profile embedding matrix $\mathbf{C}$ in parallel using respective CNN that has separate weight parameters. Each CNN in PCNN consists of a convolutional layer and a max-pooling layer. We denote input embedding matrix for each CNN as $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ...\mathbf{z}_o]$. In convolutional layer, we apply a filter $\mathbf{c} \in R^{c_1 \times c_2}$ with $[s_1, s_2]$-stride[1] to input embedding matrix and produce a new feature map $\mathbf{m}$, which is given by:

$$\mathbf{m} = f(\mathbf{Z} \odot \mathbf{c} + b)$$

where $\odot$ is the convolution operator, $b \in R$ is a bias term and f is $ReLu$. In pooling layer, we use a max-pooling operation on feature map $\mathbf{m}$, and identify the most significant local feature matrix as $\mathbf{p}$=$maxpooling(\mathbf{m})$. Meanwhile we utilize multiple filters $\mathbf{c}^i, i = 1, ..., v$ to generate multiple local feature matrix $\mathbf{p}^i$, and take the concatenation of these local features as the representation of input embedding $\mathbf{Z}$, i.e.,

$$r(\mathbf{Z}) = [vec(\mathbf{p}^1); vec(\mathbf{p}^2); ...vec(\mathbf{p}^v)]$$

where $vec(\mathbf{p})$ is the vector operation that reshapes the matrix $\mathbf{p}$ to a column vector, and $r(\mathbf{Z})$ is a column vector.

Hence for the input title embedding $\mathbf{T}$ and profile embedding $\mathbf{C}$, we separately obtain their features representations as $r(\mathbf{T})$ and $r(\mathbf{C})$ through two parallel CNN. Finally we concatenate $r(\mathbf{T})$ and $r(\mathbf{C})$ as the final news feature representation, i.e.,

$$\mathbf{I} = [r(\mathbf{T}); r(\mathbf{C})]$$

ANN:User Interest Extractor

ANN as user interest extractor obtains user's current interest with respect to the candidate news based on his clicked history news. ANN considers the user's clicked news have different impacts on candidate news. Therefore it devises an attention-based neural network, which automatically matches each clicked news to the candidate news, and aggregates the user's current interest with different weights.

Given an user's clicked history news $\{x_1, x_2, ..., x_{t-1}\}$ and the candidate(target) news x_t, their embedding representation are $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_{t-1}\}$ and $\mathbf{I}_t$, ANN employs following attention mechanism to calculate the user's current interest feature representations:

$$\begin{aligned} \mathbf{u}_j &= tanh(\mathbf{W}_w \mathbf{I}_j + \mathbf{b}_w) \\ \mathbf{u}_t &= tanh(\mathbf{W}_t \mathbf{I}_t + \mathbf{b}_t) \\ \alpha_{j,t} &= \frac{exp(\mathbf{v}^T(\mathbf{u}_t + \mathbf{u}_j))}{\sum_j exp(\mathbf{v}^T(\mathbf{u}_t + \mathbf{u}_j))} \\ \mathbf{s}_t &= \sum_j \alpha_{j,t} \mathbf{I}_j \end{aligned} \tag{1}$$

where $\alpha_{j,t}$ is weight used to measure the impact of clicked news x_j on candidate news x_t, $\mathbf{s}_t$ is the user's current interest representation. $j = 1, ..., t-1, \mathbf{W}_w, \mathbf{W}_t \in$

[1]$[s_1, s_2]$-stride means that convolution strides along two dimensions of matrix are respectively s_1 and s_2.

$R^{s \times s}, \mathbf{b}_w, \mathbf{b}_t, \mathbf{v}, \mathbf{I}_j \in R^s$, s is the dimension of news embedding.

ARNN:Sequential Information Extractor

ARNN as sequential information extractor is an attention-based RNN, whose aim is to capture the user's history sequential features from the user's history readings. ARNN assumes that the user's click at each time is influenced by previous news selection. Therefore, ARNN adds the attention mechanism on each state of RNN for getting richer sequential feature at different clicking time. These features from distinct time are integrated as the sequential feature representation of user's clicking history. Here Long Short-Term Memory(LSTM) (Graves and Schmidhuber 2005) is utilized in our ARNN component.

As Figure 2 shown, ARNN takes user's clicked news embedings $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_{t-1}\}$ as input, and output user's history sequential feature representation. In j-step of LSTM, both the j-th news embeding and the output of $j-1$-step of LSTM are taken as input, and it outputs an intermediate representation $\mathbf{h}_j$, which is given by:

$$\mathbf{h}_j = LSTM(\mathbf{h}_{j-1}, \mathbf{I}_j)$$

Further we feed the first j-step outputs of LSTM into ANN component and get the j-step sequential feature as $\mathbf{s}_j (j \in [2, t-1])$. Note that ANN contained in ARNN just has the same network as ANN described in above subsection, they have their respective parameters. Thus for the user's click history, we can get the sequential feature matrix as $\mathbf{S} = [\mathbf{s}_2\ \mathbf{s}_3\ ...\mathbf{s}_{t-1}]$, the final user's history sequential feature representation is defined as $\tilde{\mathbf{h}} = f(\mathbf{S})$, where f is integration function that *integrates* a matrix to a vector. Here we define four types of integration function f as:

$$\begin{aligned} f(\mathbf{S}) = cnn(\mathbf{S}) \quad or \quad f(\mathbf{S}) = sum(\mathbf{S}) = \sum_{j=2}^{n_i} \mathbf{s}_j \\ f(\mathbf{S}) = vec(\mathbf{S}) \quad or \quad f(\mathbf{S}) = mul(\mathbf{S}) = \prod_{j=2}^{n_i} \mathbf{s}_j \end{aligned} \tag{2}$$

where cnn() means that we model the user's sequential feature representation as the output of convolutional neural network with $\mathbf{S}$ as input.

Similarity

Finally we feed the concatenation of feature representations $\tilde{\mathbf{h}}$ and $\mathbf{s}_t$ into a fully connected network and get final user's embedding $\tilde{\mathbf{I}}_t = \mathbf{M}_g[\tilde{\mathbf{h}}; \mathbf{s}_t], \mathbf{M}_g \in R^{s \times 2s}$, and use the similarity between user's embedding and candidate news embedding to define the probability that user clicks the candidate news x_t:

$$P = cosine(\tilde{\mathbf{I}}_t \cdot \mathbf{I}_t)$$

Training

We use the existing observed clicked history reading as positive samples and unobserved history reading as negative samples for training our DAN model. We denote a training sample as $X = (\{x_1, x_2, ..., x_{t-1}\}, x_t, y)$, x_j is the j-th news clicked by users, x_t is the candidate news. For each positive input sample, its label is $y = 1$, also $y = 0$ is the label for the negative sample. After our model, each input sample has the respective estimated probabilities $P \in [0, 1]$ of the user clicking the news x_t. Then we minimize the following negative log-likelihood function to train our model:

$$L_r = -\{\sum_{X \in \Delta^+} y\ log\ P + \sum_{X \in \Delta^-} (1-y)\ log(1-P)\} \tag{3}$$

where Δ^+ and Δ^- are the positive sample set and negative sample set.

As the fixed-length input is required in convolutional layer of our PCNN component, we pad all title inputs whose length are less than *lenT* with special symbols at the end of original inputs. Similarly, we also pad the profile inputs with length not more than *lenP*. Usually we set *lenT*=10, and *lenP*=80. For regularization and avoiding over-fitting, two metrics are employed in our DAN model: dropout and L2 weight regularization. We apply dropout and L2 regularization to the weight parameters of three components and the last fully-connected layer for preventing co-adaptation.

Experiment

Experiment Setup

Datasets. We conduct experiments on a real-world online news dataset *Adressa*[2] (Gulla et al. 2017) to evaluate the performance of our model. *Adressa* is published with the collaboration of Norwegian University of Science and Technology (NTNU) and Adressavisen (local newspaper in Trondheim, Norway) as a part of RecTech project on recommendation technology. It is an event-based news dataset that includes anonymized users with their clicked news articles. Each reading event corresponds to a user reading a particular news article that contains 18 attributes. These attributions are not all very useful in our recommender system, thus we just select the (sessionStart, sessionStop)[3], user id, news id, time-stamp, the title and profile of news as our selection for generating our datasets. Specifically we first string the news article sequence into a session according to timestamp for a particular user and split longer news sequence into shorter sequence. For example, for the longer sequence $\{x_1, x_2, ..., x_T\}$, we can get $T - l + 1$ shorter positive samples $\{x_t, x_{t+1}, ..., x_{t+l-1}\}, t = 1, ..., T - l + 1$, and we set $l = 10$ to generate our positive samples on all the data sets. Then we treat the shorter article sequence in a session as a training sample. The selected sequences are positive samples that have true candidate news, for producing the negative samples, we just randomly select a news from all news set as the negative candidate news.

Two versions of *Adressa* data set named *Adressa-1week* and *Adressa-10week*, are released, which separately collects as long as 1 week(from 1 January to 7 January 2017) and 10 weeks (from 1 January to 31 March 2017) datasets. We reorganize the original dataset according to our strategies, and illustrate the details of such two data sets in Table 1.

[2]http://reclab.idi.ntnu.no/dataset/

[3]sessionStart and sessionStop reveal if the event is at the beginning or end of a user session are given

Table 1: Statistics of the dataset.

Number	*Adressa-1week*	*Adressa-10week*
#users	640,503	3,614,911
#news	20,428	81,018
#events	3,101,991	35,244,078
#entity	160,559	417,572
#entity-type	19	19
#average words per title	6.57	6.64
#average entities per news	27.7	26.5
#average entity-types per news	12.6	12.5

Parameter Settings. We implement our model based on Tensorflow. In our model, we select the dimension of entity embeddings d and entity type k from $\{20, 50, 100, 200, 300\}$. For each CNN in PCNN component, we select the size of filter $[c_1, c_2]$ from all combinations in set $\{o, \frac{o}{2}, \frac{o}{3}, \frac{o}{4}, \frac{o}{5}\}$ and $\{d, \frac{d}{2}, \frac{d}{3}, \frac{d}{4}, \frac{d}{5}\}$, the filter stride $[s_1, s_2]$ from all combinations in set $\{2, 3, 4\}$, and the number of filter v from $\{4, 8, 16, 32, 64\}$, batch size B from $\{64, 128, 256, 512, 1024\}$. The best parameters are determined by validation set. To compare our model with baselines, we use F1 and AUC value as the evaluation metrics.

For our model, the best optimal parameter configurations are $d = 100$, $k = 100$, $B = 256$, $v = 8$. In PCNN, for CNN with title embedding as input , $o = 10, [c_1, c_2] = [2, 50], [s_1, s_2] = [2, 2]$, for CNN with profile embedding as input, $o = 80, [c_1, c_2] = [40, 50], [s_1, s_2] = [2, 2]$. The key parameter settings for baselines are same as configurations reported in DKN (Hongwei Wang 2018).

Baselines

We use the following state-of-the-art methods as baselines in our experiments.

- LibFM (Rendle 2012) is an excellent feature-based factorization model and widely used in click-through rate prediction scenarios. In this paper, we use the concatenation of news title and profile embeddings as input features.
- DSSM (Huang et al. 2013) is a typical deep structured semantic model for document ranking given a query. It projects the given query and the set of documents into low-dimensional vectors using the word hashing and fully connected layers, and calculates the cosine similarity between their vectors. In this paper, we model the user's clicked news as the query and candidate news as the documents.
- DeepWide (Cheng et al. 2016) is a deep learning based model that combines the feed-forward neural network(Deep) and the linear model (Wide) for recommendation. In this paper, we also use the concatenation of news title and profile embeddings as features.
- DeepFM (Guo et al. 2017) is an end-to-end factorization machine based neural network model for recommendation system, which combines the power of factorization machines for recommendation and deep learning for feature learning. We use the same input as in DeepWide for DeepFM.

Table 2: Comparison of different models.

Model	**Adressa-1week**		**Adressa-10week**	
	F1	AUC	F1	AUC
LibFM(-)	63.93	61.79	55.75	53.83
LibFM	70.69	69.53	64.44	61.41
DSSM(-)	69.36	68.25	62.24	60.74
DSSM	74.78	72.71	69.11	67.57
DeepWide(-)	67.39	64.83	59.98	57.98
DeepWide	73.94	71.07	67.87	66.80
DeepFM(-)	66.09	64.83	58.47	57.03
DeepFM	72.47	70.33	64.71	63.60
DMF	63.43	61.49	55.43	53.47
DKN(-)	75.38	73.45	68.57	60.57
DKN	79.97	77.24	70.39	67.53
DAN	**82.32**	**80.18**	**73.58**	**70.17**

- DMF (Xue et al. 2017) is a deep matrix factorization model for recommender systems, which uses multiple non-linear layers to process raw rating vectors of users and items. We ignore news embedding and take the implicit feedback[4] as input for DMF.
- DKN (Hongwei Wang 2018) is a deep recommendation framework, which devises a multi-channel CNN to fuse semantic-level and knowledge-level representations of news, and introduces an attention mechanism for click-through rate prediction. In this paper, we model news title and profile as semantic-level and knowledge-level representations.

Note that DMF is a collaborative filtering based model. Moreover, except for LibFM, all those models are based on deep neural networks. In this paper, we aim to compare our model with deep learning models.

Results and analysis

In this experiment, we compare our DAN with several baseline models, and we rerun these compared models on the dataset. Table 2 shows the results of comparison of different models. For the compared models, we also remove the profile embeddings from the input matrix, and such results are denoted by "(-)" in Table 2.

Our model DAN consistently outperforms all baselines on both datasets, which is at least 3.19% on F1, 2.64% on AUC higher than other models. We attribute the superiority of our model to its three advantages: (1)Our model uses PCNN to process title and profile of news and can better extract the specific local patterns in news. (2) Our model uses ARNN to better capture the sequential correlation of user's clicked history reading. (3) Our model uses ANN to treat users' click history discriminatively, which better captures users' diverse reading interests. More detailed analysis is shown as follows:

The performance of most models can be increased by using profile embedding. For example, DKN with considering

[4]The interests in terms of click counts and time spent reading the articles in Adressa are regarded as implicit feedback.

Table 3: Comparison among DAN variants.

Model	**Adressa-1week**		**Adressa-10week**	
	F1	AUC	F1	AUC
DAN without entity and entity-type	76.01	72.51	69.37	58.19
DAN with entity-type	77.21	74.49	70.76	65.42
DAN with entity	80.46	78.64	71.93	67.63
DAN with entity and entity-type	**82.32**	**80.18**	**73.58**	**70.17**
DAN without mapping	78.06	75.36	67.13	65.49
DAN with linear mapping	80.17	77.24	70.57	67.81
DAN with no-linear mapping	**82.32**	**80.18**	**73.58**	**70.17**
DAN without attention	78.13	75.44	70.23	68.76
DAN with attention	**82.32**	**80.18**	**73.58**	**70.17**
DAN with mul	72.46	68.54	60.39	58.73
DAN with sum	80.17	78.29	69.98	67.04
DAN with vec	79.91	77.74	69.46	66.29
DAN with cnn	**82.32**	**80.18**	**73.58**	**70.17**
DAN without ARNN	81.59	77.27	71.25	69.61
DAN with ARNN	**82.32**	**80.18**	**73.58**	**70.17**

profile embedding as input has 4.59%(1.82%) higher than respective models with "(-)" on F1 and 3.79%(6.96%) on AUC. It proves that profile embedding can provide complementary information of news for recommendation.

We also find that all models achieve better performance than DMF model, which is CF-based model. This is because CF-based methods cannot work well in recommending news due to the time-sensitive attribution of news.

Except for DMF, LibFM has the lowest performance among deep learning based baselines, which is at least 1.78%(0.27%) to 11.63%(9.14%) on F1 and 0.80%(2.19%) to 10.65%(8.76%) on AUC lower than other deep learning based models for both data sets. It suggests that non-linear relations and dependencies in news data can well be captured by deep models.

Discussion on different DAN variants

Further for demonstrating the efficacy of the design of our model, we compare among the variants of our model with respect to the following several aspects: the usage of entity and entity types, the choice of transformation function, the usage of an attention mechanism, the choice of integration function and the usage of ARNN componnet. The results are shown in Table 3, from which we can conclude that:

Model with extra either entities or entity types is more competitive than model that only considers the news title. Additionally model with considering both entity and entity type achieves best performance. It indicates that entities and entity types can provide indispensable and effective characteristic information of news.

Incorporating mapping function into our model excellently improves performance compared with model without mapping function, suggesting that the heterogeneity between entity and entity-type space can be alleviated by self-learning of transformation function. Moreover non-linear mapping function has a stronger ability of alleviating the heterogeneity.

Introducing attention mechanism into our model can bring a 4.19%(3.35%) gain on F1 and 4.74%(1.41%) gain on AUC for both datasets. The mainly reason is that the attention mechanism can dynamically get the user's history sequential information representation and the user's current interest representation.

The integration function using *cnn* mode has the best performance than other three modes. The *sum* and *vec* modes have similar results. This is mainly because that *cnn* mode may be more effective to extract local correlation information and weak redundant information of multiple vectors than simple linear operation *sum*, *mul* and *vec*.

DAN with ARNN component has higher performance than DAN without ARNN component, which suggests that the extracted history sequential information of a user's clicked news using ARNN is beneficial to improve the recommendation performance.

Different dimension of entity embedding and entity-type embedding.

In this section, we mainly explore the effect of different dimension of entity embedding d and entity type embedding k on the performance. In this experiment, expect for the parameter being tested, all other parameters are set as optimal configuration. We select the dimensions from all combinations of d and k in set $\{20, 50, 100, 150, 200\}$. Figure 3 gives the convinced results, which are (1) Our model achieves best performance at $d = k = 100$ setting, indicating than such dimension setting best express semantic information of entity space and entity type space. (2) Given the dimension of entity embedding d, the performance of our model first is increasing with the growth of k and then drops with k further increases. That is, too small or too large dimension can reduce the performance. This is because that too low dimension has insufficient capability of capturing the necessary information and properties, too large dimension introduces unnecessary noise and reduces generalization ability. The case is similar for d when k is given.

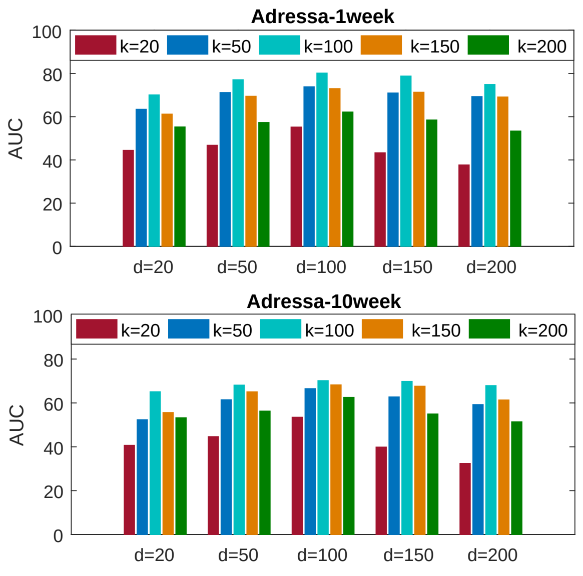

Figure 3: Dimension sensitivity of entity and entity-type embedding.

Related Work

With the growth of a gigantic number news articles, how to make personalized news recommendation based on the user's preference has been extensively investigated in research community and industry. For tackling personalized news recommendation, a variety of techniques have been proposed including collaborative filtering (CF) based methods (Das et al. 2007; Marlin and Zemel 2004; Rendle 2010; Sheng, Kawale, and Fu 2015; Wu et al. 2016; Xue et al. 2017), content based methods (IJntema et al. 2010; Kompan and Bieliková 2010; Huang et al. 2013), hybrid methods (Morales, Gionis, and Lucchese 2012; Li et al. 2011; Liu, Dolan, and Pedersen 2010) and deep learning based methods (Zheng, Noroozi, and Yu 2017; Wang et al. 2017).

CF-based methods aim to predicts a personalized ranking over a set of items (music, book, movie or news) for each individual user with the similarities among the users and items. It is a standard for recommendation due to their scalability, simplicity and flexibility. For example, Autoencoders (Sheng, Kawale, and Fu 2015) reconstructs the users' ratings through learning hidden structures with the explicit historical ratings. Another method CDAE (Wu et al. 2016) makes recommendation by modeling user's preference with implicit historical feedback. Different from other related methods using either only explicit ratings or only implicit ratings, DMF (Xue et al. 2017) as a typical CF-based model makes full use of both explicit ratings and implicit feedback for news recommendation. CF-based methods often suffer from the cold-start problem since news items are substituted frequently.

For addressing the issues of CF-based methods, a large amount of content-based or hybrid methods have been proposed. Content based methods consider the actual content or attributes of the items for making recommendations. Hybrid methods recommend items through a hybrid recommender system that usually combines several different recommender algorithms. For example, DSSM (Huang et al. 2013) is a content-based deep neural network to rank a set of documents for a given query, where the ranking is related to the relevance of query to each document calculated by cosine similarity between the low dimensional vectors of query and document. SCENE (Li et al. 2011) is a hybrid method based on two-stage personalized news recommendation, in which the first stage learns users' representation for exclusive characteristics(news content, named entities), and the second stage learns users' representation from the intrinsic property of user interest.

Recently deep learning based methods can works well in modeling complex user-item interactions. For instance, DeepWide (Cheng et al. 2016) jointly combines a linear model component(wide) and a neural network component(deep) to achieve both memorization and generalization in one model. DeepFM (Guo et al. 2017) integrates the architectures of FM for learning low-order feature interactions and deep neural networks (DNN) for learning high-order feature interactions. Compared with WideDeep model, DeepFM has a shared input to its "wide" and "deep" parts. DKN (Hongwei Wang 2018) is state-of-the-art deep learning based models, which fuses semantic-level and knowledge-level representations of news through a multi-channel CNN, and introduces an attention mechanism to model dynamic user's embedding for click-through rate prediction.

DeepJoNN (Zhang, Liu, and Gulla 2018) uses a tensor-based CNN to extract session-based news representations and a RNN to compute the freshness of news within a time slide, which is a session-based model but not an event-based model. Different from DeepJoNN, our model DAN utilizes parallel CNN to learn event-based news representation and attention-based RNN to capture the sequential information of clicked news, and also adopt an attention-based neural network for getting user's current interests.

Conclusion

This paper proposes a deep attention neural network DAN for news recommendation, which considers the user's history sequential information and user's current interest together to determine whether the user clicks on the candidate news. Therefore three components are devised, including news representation extractor PCNN, sequential information extractor ARNN and user interest extractor ANN. PCNN learns the news representation through two parallel CNNs by fusing the title-level and profile-level information of news. Based on such news representations, ARNN captures the user's history sequential information through an attention-based RNN, which enforces the attention mechanism on each state of RNN. Meanwhile ANN also utilizes the same attention mechanism in ARNN to learn the user's current interest with respect to candidate news. We conduct extensive experiments on a real world data set *Adressa*. The experimental results demonstrate the significant superiority of our DAN model compared with strong baselines.

Acknowledgment

This work is supported by National Key R&D Program No.2017YFB0803003, and the National Natural Science Foundation of China (No.61202226), We thank all anonymous reviewers for their constructive comments.

References

Bansal, T.; Das, M.; and Bhattacharyya, C. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. 195–202.

Cheng, H. T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; and Ispir, M. 2016. Wide & deep learning for recommender systems. 7–10.

Das, A.; Datar, M.; Garg, A.; and Rajaram, S. 2007. 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*, 271–280.

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18(5-6):602–610.

Gulla, J. A.; Zhang, L.; Liu, P.; ?zlem ?zg?bek; and Su, X. 2017. The adressa dataset for news recommendation. In *the International Conference*, 1042–1048.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. Deepfm: A factorization-machine based neural network for ctr prediction. 1725–1731.

Hongwei Wang, Fuzheng Zhang, X. X. M. G. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *In Proceedings of the 27th international world wide web conference*, 191–198.

Huang, P. S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *ACM International Conference on Conference on Information & Knowledge Management*, 2333–2338.

IJntema, W.; Goossen, F.; Frasincar, F.; and Hogenboom, F. 2010. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops, Lausanne, Switzerland, March 22-26, 2010.*

Kompan, M., and Bieliková, M. 2010. Content-based news recommendation. In *E-Commerce and Web Technologies, 11th International Conference, EC-Web 2010, Bilbao, Spain, September 1-3, 2010. Proceedings*, 61–72.

Li; Lihong; Chu; Wei; Langford; John; Schapire; and Robert, E. 2010. A contextual-bandit approach to personalized news article recommendation. 661–670.

Li, L.; Wang, D.; Li, T.; Knox, D.; and Padmanabhan, B. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, 125–134.

Liu, J.; Dolan, P.; and Pedersen, E. R. 2010. 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*, 31–40.

Marlin, B. M., and Zemel, R. S. 2004. The multiple multiplicative factor model for collaborative filtering. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004.*

Morales, G. D. F.; Gionis, A.; and Lucchese, C. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*, 153–162.

Rendle, S. 2010. Factorization machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, 995–1000.

Rendle, S. 2012. Factorization machines with libfm. *Acm Transactions on Intelligent Systems & Technology* 3(3):1–22.

Sheng, L.; Kawale, J.; and Fu, Y. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. 811–820.

Wang, X.; Yu, L.; Ren, K.; Tao, G.; Zhang, W.; Yu, Y.; and Wang, J. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 2051–2059.

Wu, Y.; Dubois, C.; Zheng, A. X.; and Ester, M. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *ACM International Conference on Web Search and Data Mining*, 153–162.

Xue, H. J.; Dai, X. Y.; Zhang, J.; Huang, S.; and Chen, J. 2017. Deep matrix factorization models for recommender systems. In *International Joint Conference on Artificial Intelligence*, 3203–3209.

Zhang, L.; Liu, P.; and Gulla, J. A. 2018. A deep joint network for session-based news recommendations with contextual augmentation. In *Proceedings of the 29th on Hypertext and Social Media, HT 2018, Baltimore, MD, USA, July 09-12, 2018*, 201–209.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, 425–434.