Dynamic News Recommendation with Hierarchical Attention Network

Hui Zhang*†, Xu Chen‡, Shuai Ma*†
*SKLSDE Lab, Beihang University, China
†Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China
‡School of Software, Tsinghua University, China
{zhangh17, mashuai}@buaa.edu.cn, xu-ch14@mails.tsinghua.edu.cn

***Abstract*—News recommendation is an effective information dissemination solution in modern society. In general, news articles can be modeled from multiple granularities: sentence-, element- and news-level. However, the first two levels have been largely ignored in existing methods and it is also unclear how such multi-granularity modeling can enhance news recommendation. In this paper, we propose a novel dynamic model for news recommendation. A unique perspective of our model is to discriminate the contributions of previously interacted contents for triggering the next news-reading, in sentence-, element- and news-level simultaneously. To this end, we design a hierarchical attention network of which the lower layer learns the impacts of sentences and elements, while the upper layer captures disparity of news. Moreover, we incorporate a time-decaying factor to reflect the dynamism, as well as convolution neural networks for learning sequential influence. Using three real-world datasets, we conduct extensive experiments to verify the superiority of our model, compared with several state-of-the-art approaches.**

***Index Terms*—news recommendation, attention model, dynamic model, convolutional neural networks**

I. Introduction

The ever-prospering World Wide Web (WWW) has gradually shifted the ways people seek news, i.e., from traditional printed media to on-line portals. To alleviate information overloading problem, news recommendation systems have been widely deployed. They can provide people with tailored news contents and improve their reading experience.

In this area, a key observation is that previously interacted news have a strong impact on the next reading choice [1]. Along this line, a number of news recommendation models have been built [2], [3], which capture people's sequential reading patterns. In spite of the results achieved, these models still face challenges. The previously interacted contents usually have different impacts on choosing the next one to recommend in practice. Given the multi-granularity modeling of news articles, it is desired to discriminate such various contributions in sentence-, element- and news-level.

First, news articles are composed of sentences with inconsistent contents. Sentences of people's previously interacted news have different impacts on their next actions. For example, a user has read a news article about an NBA match, after that he will read another news about a new match largely due to the sentences forecasting the match in the previous news. Second, news articles have basic components called news elements, which are known as 5W1H, i.e., *who*, *when*, *where*, *what*, *why* and *how* [4]. The 5W1H elements clearly describe the key information of news in an explicit manner. Only parts of elements of people's previously read news influence their choices. For example, for a news article that a user read before, the first 4W elements are "Warriors, Cavaliers", "June 1-9, 2018", "Cleveland, Oakland" and "NBA finals", respectively. The user will read another news about "Warriors" mostly due to the *who* element in the previous news. Third, think of each news as a whole, news articles in people's reading logs have different influences on their decisions. For example, a user will read a sports news mostly because of the sports news rather than the economic news in his reading logs. Besides, the difference also comes from each user's unequal preferences for historical news. What's more, due to the timeliness in news scenario, it is necessary to incorporate dynamism. Specifically, in a short time interval, a user tends to read similar news contents for maximum information.

Motivated by the above observations, in this paper, we propose **DNA**, a Dynamic News recommender based on a hierarchical Attention network. The main block of our model is a two-layer attention network that learns different impacts of previously interacted news contents on the next choice. Specifically, the lower layer determines attention weight for each sentence and element with respect to the candidate news. The upper layer determines attention weight for each news in relation to the candidate news. Besides, we further incorporate a time-decaying factor and news embedding in the upper layer, which learn dynamism and structural information respectively. Then, the aboves are convolutional layers for learning users' sequential news-reading information and fully-connected layers for computing the click rate. By such a model, we improve the performance of news recommendation, compared with several state-of-the-art methods.

The contributions of this paper can be concluded as follows:

- We highlight the concept of discriminating various influences of the previously interacted news articles on the target one in the domain of news recommendation.
- We propose a dynamic model for news recommendation, which learns to discriminate impacts by incorporating sentence-, element- and news-level attention mechanisms.
- We conduct extensive experiments on three real-world datasets to demonstrate the superiority of our model, from both qualitative and quantitative perspectives.

II. Overview

A. Problem Definition

Assuming there is an on-line news platform offering services to a set of users. Once the platform receives a new piece of news, it estimates the click rate for each user, based on the contents the user had read earlier. Due to the timeliness in news scenario, it suffices to consider the influence of the most recent pieces of news. Formally, let $\mathcal{C}_i = [c_1, c_2, ..., c_L]$ denotes the sequence of the most recent L pieces of news read by user i. Each piece of news c_j consists of a sequence of K sentences, i.e., $[s_1^j, s_2^j, ..., s_K^j]$, where s_k^j is the k-th sentence of c_j. Each piece of news c_j is also represented by a set of news elements. Given the news sequence $\mathcal{C}_i$ and candidate news c^*, we aim to predict the click rate of c^* by user i.

B. News Elements

We can not employ existing methods to extract 5W1H elements due to the language difference [5] and the lack of specific news corpus [6]. We define news elements by ourselves that can be easily extracted by NLP tools. Specifically, they are *person*, *organization*, *time*, *location* and *keywords*, corresponding to *who*, *who*, *when*, *where* and *what* elements of 5W1H respectively. Formally, each piece of news c_j is represented by a set of elements, i.e., $\{e_p^j, e_o^j, e_t^j, e_l^j, e_{ke}^j\}$.

C. DNA Framework

Fig. 1 illustrates the architecture of DNA model. It has three main components: the core hierarchical attention layers, the convolutional layers and the fully-connected layers. Given news sequence $\mathcal{C}_i$ of user i and candidate news c^* as input, it first computes weights of sentences in the sentence-level attention and gets the content vector $\mathbf{v}(c_j)$ of news c_j by assembling the content vectors of sentences with their attention weights. Simultaneously, it computes weights of elements in the element-level attention and the element vector $\mathbf{l}(c_j)$ of news c_j is derived as a weighted sum of the element vectors of elements in it. Besides, it also learns an embedding $\mathbf{n}_{c_j}$ for news c_j such that the concatenation $[\mathbf{v}(c_j)\,\mathbf{l}(c_j)\,\mathbf{n}_{c_j}]$ determines a temporary representation of news c_j. Then in the upper news-level attention, it further computes weights of news, which also incorporate the time-decaying factor, and obtains the candidate-dependent news representation $\mathbf{x}_j$ based on $[\mathbf{v}(c_j)\,\mathbf{l}(c_j)\,\mathbf{n}_{c_j}]$ and its attention weight. By stacking these news representations into a matrix in temporal order, the convolutional layers learn the convolutional sequence vector $\mathbf{p}_i$ for user i, which captures sequential reading patterns. Finally, the convolutional sequence vector $\mathbf{p}_i$, the candidate news representation $\mathbf{x}^*$ and the user embedding $\mathbf{u}_i$ are concatenated and fed into the fully-connected layers to compute the click rate of news c^* by user i.

III. Our DNA Model

We first show the designs of sentence-, element- and news-level attention mechanisms. Then we introduce the details of the convolutional layers and the fully-connected layers.

A. Sentence-level Attention

Intuitively, sentences content-relevant to the candidate news have large impacts on reading. The sentence-level attention aims to discriminate various influences of sentences.

We first obtain content vectors of sentences of news c_j and content vector of news c^*. We utilize Paragraph Vector [7] due to its consideration of the ordering and semantics of the words. The content vector $\mathbf{v}(s_k^j)$ of sentence s_k^j is embedded in a d-dimensional space. Furthermore, the content vector $\mathbf{v}(c^*) \in \mathbb{R}^d$ of news c^* is calculated by averaging the content vectors of sentences in it.

We adopt a two-layer feed-forward neural network to determine the *un-normalized* attention weight b_k^j of s_k^j:

$$b_k^j = \mathbf{W}_2\, \phi(\, \mathbf{W}_1\, [\, \mathbf{v}(s_k^j)\, \mathbf{v}(c^*)\,] + \mathbf{b}_1), \tag{1}$$

where $[\, \mathbf{v}(s_k^j)\, \mathbf{v}(c^*)\,]$ is the concatenation of $\mathbf{v}(s_k^j)$ and $\mathbf{v}(c^*)$, $\phi(x)$ is the ReLU function and $\mathbf{W}_1 \in \mathbb{R}^{d \times 2d}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times d}$ and $\mathbf{b}_1 \in \mathbb{R}^d$ are the parameters of the neural network. We further obtain the *normalized* attention weight β_k^j of s_k^j by softmax function. Here β_k^j can be interpreted as the content relevance between sentence s_k^j and candidate news c^*. Based on these weights, the content vector $\mathbf{v}(c_j) \in \mathbb{R}^d$ of news c_j with respect to news c^* is calculated as a weighted sum of the content vectors of sentences.

B. Element-level Attention

Elements, which are also included in the candidate news or similar to elements in the candidate news, play an important role in clicking. The goal of the element-level attention is to discriminate various impacts of elements.

With named entity recognition and keywords extraction modules of NLP tools, we can extract elements, i.e., *person*, *organization*, *time*, *location* and *keywords*, for news c_j and c^*. Each element is extracted in the form of one or more words. Distributed word embeddings are learned by Word2vec [8] that is successful in capturing semantics relatedness. Assuming that each word is embedded in a d-dimensional space. For instance, the element vector $\mathbf{l}(e_p^j) \in \mathbb{R}^d$ of element e_p^j is obtained by averaging the vectors of words that represent e_p^j. Furthermore, the element vector $\mathbf{l}(c^*) \in \mathbb{R}^d$ of the news c^* is the average of all element vectors of it.

We calculate the dot product between the corresponding element vectors of news c_j and c^* to determine the *un-normalized* attention weight. Still take *person* element for example, the *normalized* attention weight γ_p^j of element e_p^j is also obtained by softmax function. Here γ_p^j can be interpreted as the *person* element relevance between news c_j and candidate news c^*. At the end, we compute the element vector $\mathbf{l}(c_j) \in \mathbb{R}^d$ of news c_j with respect to candidate news c^* as a weighted sum of all element vectors.

C. News-level Attention

News articles that are content-relevant to the candidate news have large impacts on the click rate. Besides, user preferences and dynamism are also important for making predictions.

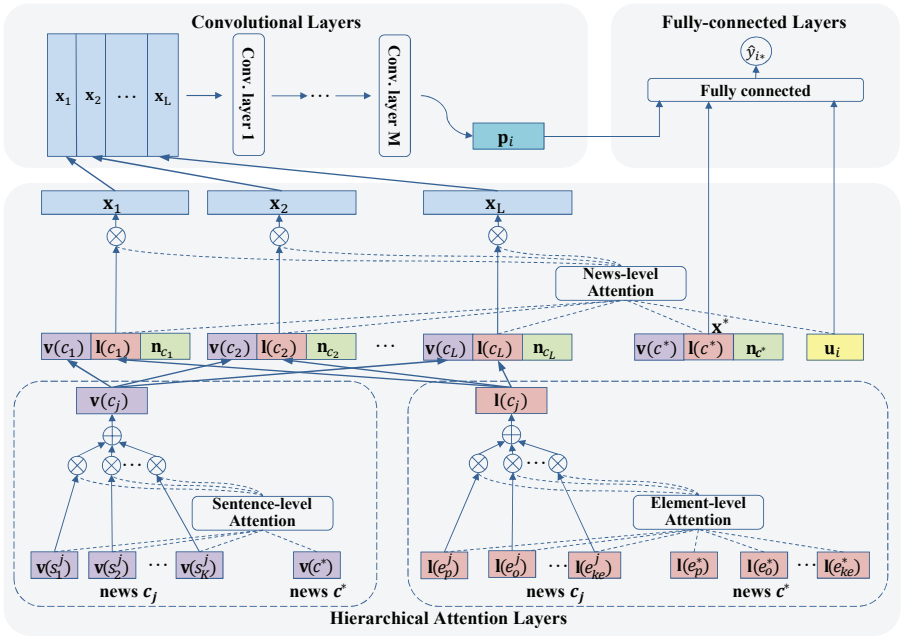


Fig. 1. The architecture of DNA model.

We utilize the news-level attention to discriminate various contributions of news articles.

Note that so far we have obtained the content vector and element vector for each news, which only depend on news contents. However, structural information also provides a way to measure news similarity. For instance, two pieces of news which are frequently co-clicked by people tend to be similar. To extract the structural information, we learn an embedding in a d-dimensional space for each news, i.e., $\mathbf{n}_{c_j} \in \mathbb{R}^d$ for news c_j. For news c_j, the temporary representation is $[\mathbf{v}(c_j)\,\mathbf{l}(c_j)\,\mathbf{n}_{c_j}]$. For candidate news c^*, this operation leads to its representation, i.e., $\mathbf{x}^* = [\mathbf{v}(c^*)\,\mathbf{l}(c^*)\,\mathbf{n}_{c^*}] \in \mathbb{R}^{3d}$. Moreover, users' structural information partly reflects user preferences [9]. Therefore, we learn an embedding in a d-dimensional space for each user, i.e., $\mathbf{u}_i \in \mathbb{R}^d$ for user i. News embeddings and user embeddings both are randomly initialized and automatically learned in training phase.

To include content relevance and user preferences, we adopt another two-layer feed-forward neural network to calculate the *un-normalized* news-level attention weight a_j of c_j:

$$a_j = \mathbf{W}_4\,\phi(\,\mathbf{W}_3\,[\,\mathbf{v}(c_j)\,\mathbf{l}(c_j)\,\mathbf{n}_{c_j}\,\mathbf{x}^*\,\mathbf{u}_i\,]\,+\,\mathbf{b}_2)\,+\,\mathbf{b}_3, \quad (2)$$

where $\mathbf{W}_3 \in \mathbb{R}^{3d\times 7d}$, $\mathbf{W}_4 \in \mathbb{R}^{1\times 3d}$, $\mathbf{b}_2 \in \mathbb{R}^{3d}$ and $\mathbf{b}_3 \in \mathbb{R}$ are the parameters of the neural network.

We further incorporate the time-decaying factor into news-level attention. This idea comes from that people tend to read similar news contents in a short time. For example, a user has just finished reading a news article, and he intended to learn more about related news in a short time, e.g., 1 minute. However, the impact of this news on the user became weaker when time passed for a long time, e.g., 6 hours. We exploit our observations to model the temporal dynamic of user news-reading behaviors. Specifically, the time-decaying factor of news c_j is defined as an exponentially decaying formula [10]:

$$f_t(j) = \exp(-\eta(t^* - t_j)/3600), \quad (3)$$

where $\eta \geq 0$ represents the time-decaying rate, t_j is the timestamp of user i reading news c_j and t^* is the timestamp of making the recommendation.

We then calculate $a_j f_t(j)$ for news c_j and further apply softmax function to obtain the *normalized* weight α_j of c_j. Here α_j can be interpreted as the content relevance and time proximity between news c_j and candidate news c^*, and also user i's preference for c_j. We compute the representation of news c_j with respect to news c^* as follows:

$$\mathbf{x}_j = \alpha_j\,[\,\mathbf{v}(c_j)\,\mathbf{l}(c_j)\,\mathbf{n}_{c_j}\,] \in \mathbb{R}^{3d}. \quad (4)$$

D. Convolutional Layers

Due to the sequential characteristic of news-reading [1], we exploit convolutional neural network (CNN) to capture sequential information. CNN models sequential patterns as local features using convolutional filters [11]. To be specific, we first stack the representations of L news into a feature map $E \in \mathbb{R}^{L\times 3d}$. A convolutional layer has m convolution filters $F^q \in \mathbb{R}^{h\times 3d}, q = 1, ..., m$, where h and $3d$ are the height and width of filters respectively. These filters capture sequential patterns by sliding over the rows of E. The result of F^q is $[f_1^q, f_2^q, ..., f_{L-h+1}^q]$, where $f_s^q \in \mathbb{R}$ is carried out by convolution operation and ReLU function. The results of m filters lead to a new feature map $E^{'} \in \mathbb{R}^{(L-h+1)\times m}$.

To model long-range sequential patterns, we adopt M convolutional layers, and all of them have the same number and height of filters. The resulting feature map of the previous

layer is the input of the next layer. The output feature map of the last convolutional layer is of size $(L - M(h-1)) \times m$. To preserve the sequential information of the user's reading, we concatenate the vectors of the output and obtain the convolutional sequence vector $\mathbf{p}_i$ for user i.

E. Fully-connected Layers

We feed the convolutional sequence vector $\mathbf{p}_i$, candidate news representation $\mathbf{x}^*$ and user embedding $\mathbf{u}_i$ into the fully-connected layers to estimate the click rate:

$$\hat{y}_{i*} = \mathbf{W}_{3f}\phi(\mathbf{W}_{2f}\phi(\mathbf{W}_f[\mathbf{p}_i\, \mathbf{x}^*\, \mathbf{u}_i] + \mathbf{b}_f) + \mathbf{b}_{2f}) + \mathbf{b}_{3f}, \quad (5)$$

where $\mathbf{W}_f \in \mathbb{R}^{5d \times (m(L-M(h-1))+4d)}$, $\mathbf{W}_{2f} \in \mathbb{R}^{2d \times 5d}$, $\mathbf{W}_{3f} \in \mathbb{R}^{1 \times 2d}$, $\mathbf{b}_f \in \mathbb{R}^{5d}$, $\mathbf{b}_{2f} \in \mathbb{R}^{2d}$ and $\mathbf{b}_{3f} \in \mathbb{R}$ are the parameters of the fully-connected layers and the binary cross-entropy loss is adopted as the objective function.

With a trained model, the time complexity of DNA model to predict the click rate of a news article for a user is $O(L \cdot K \cdot d^2)$, since it computes attention weights for $L \cdot K$ sentences, each of which costs $O(d^2)$ with a feed-forward neural network, and the complexity of other components is under $O(L \cdot K \cdot d^2)$.

IV. Experimental Study

A. Experimental Setup

1) Datasets: We conduct experiments on three datasets: Adressa, Cert and Caing. Each log of these datasets contains user ID, news ID, reading timestamp and news contents. **Adressa** is constructed by [12] and contains reading logs of 10 weeks from Adressavisen, a Norwegian news portal. **Cert** is provided by Computer Emergency Response Technical Team of China and contains user logs from various news portals, from March 2016 to April 2017. **Caing**[1] is from a news portal called Caing and contains reading logs of 10,000 users in March 2016. The statistics are shown in Table I.

2) Evaluation Protocols: Following [3], [9], we adopt the leave-one-out strategy. For each user, we use a sliding window of $L + 1$ (L historical news and 1 candidate news) length to slide over his interactions and each window generates one instance. We hold out the latest instance for testing and utilize the remaining instances for training. We randomly sample 99 news articles that are not interacted by the user and rank the 100 news articles. The performance of the ranked list is evaluated by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). We calculate both metrics for each user and report the average here.

3) Baselines: We compare our DNA model with several state-of-the-art methods. The first group only includes collaborative filtering methods, i.e., **BPR** [13] that utilizes pairwise ranking loss on implicit feedback data, **GRU4Rec** [14] that applies RNN for session-based recommendation, and **Caser** [11] that adopts 1D CNN to capture sequential patterns. The second group contains methods which utilize content information, i.e., **GRU4Rec+** that further considers news contents based on GRU4Rec, and **WE3CN** [3] that applies 3D CNN for news recommendation.

[1]http://www.dcjingsai.com/

TABLE I
STATISTICS OF EVALUATION DATASETS.

Datasets	#Interaction	#User	#News	Sparsity
Adressa	1,604,879	66,649	12,034	99.79%
Cert	1,573,959	199	588,907	98.65%
Caing	61,615	1,947	5,275	99.40%

4) Implementation: We implement deep learning models with Pytorch. We utilize the named entity recognition and keywords extraction modules of the following NLP tools: Polyglot [2] for Adressa and NLPIR [3] for Cert and Caing. For DNA model, the dimensions of content/element vector and news/user embedding are all set to 64. The number of convolutional layers is set to 2, and each layer has 64 filters of height 3. The number of sentences for each news is set to 20. Adam optimizer is applied, and the learning rate, batch size and weight decay are set to 10^{-3}, 256 and 10^{-4}, respectively. For baselines, for the sake of fairness, some parameters are reset, i.e., the dimensions of content vector, news/user embedding. Others are fixed to the default. Each experiment is repeated three times and the average is reported.

B. Performance Comparison

In this subsection, we evaluate DNA model compared with baselines. From Table II, we have the following findings.

Our DNA model achieves the best performance on all datasets, significantly outperforming the best baseline. We attribute the performance gains to the effectiveness of the hierarchical attention layers we design as well as multi-granularity news modeling.

Sequential information improves the performance of news recommendation. BPR only utilizes users' feedback information and performs the worst among all the models. In addition to users' feedback information, GRU4Rec and Caser both utilize sequential information of users' reading behaviors and perform better than BPR.

Content information improves the performance of news recommendation. In most cases, DNA model and the second group models are superior to the first group models. Indeed, GRU4Rec+ outperforms GRU4Rec on all datasets. However, it is surprising that Caser is better than GRU4Rec+ on Cert and better than WE3CN on Adressa and Caing. This may be because many adjacent actions do not have apparent dependency relationships in Cert [3] and WE3CN represents each news article with the first 50 words, which are not enough to well express news contents in Adressa and Caing.

We also examine the impacts of two hyper-parameters and report the results on Caing with HR@5 and NDCG@5, shown in Fig. 2. Scores are small when $L = 5$ because a short news-reading sequence provides insufficient sequential information. Scores first increase and then drop with the increase of η because small η makes the time-decaying factor close to 1 and large η makes only the news a user just read work.

[2]https://polyglot.readthedocs.io/en/latest/index.html
[3]http://ictclas.nlpir.org/

TABLE II
PERFORMANCE COMPARISON ON THREE DATASETS FOR ALL METHODS.

Adressa	HR@1	HR@5	HR@10	ND@1	ND@5	ND@10
BPR	0.0777	0.2676	0.4278	0.0777	0.1724	0.2239
GRU4Rec	0.2969	0.6926	0.8535	0.2969	0.5026	0.5551
Caser	0.3327	0.7277	0.8684	0.3327	0.5397	0.5856
GRU4Rec+	0.3423*	0.7348*	0.8773*	0.3423*	0.5474*	0.5939*
WE3CN	0.3070	0.6790	0.8340	0.3070	0.4965	0.5466
DNA	**0.4528**	**0.8627**	**0.9505**	**0.4528**	**0.6726**	**0.7015**
Imp.	*32.28%*	*17.41%*	*8.34%*	*32.28%*	*22.87%*	*18.12%*
Cert	HR@1	HR@5	HR@10	ND@1	ND@5	ND@10
BPR	0.2463	0.4539	0.5444	0.2463	0.3511	0.3801
GRU4Rec	0.3367	0.4623	0.5193	0.3367	0.4038	0.4222
Caser	0.4556*	0.5812	0.6281	0.4556*	0.5251	0.5403
GRU4Rec+	0.3920	0.5343	0.5662	0.3920	0.4677	0.4781
WE3CN	0.3744	0.6884*	0.8015*	0.3744	0.5447*	0.5818*
DNA	**0.5239**	**0.8116**	**0.8920**	**0.5239**	**0.6746**	**0.7007**
Imp.	*14.99%*	*17.90%*	*11.29%*	*14.99%*	*23.85%*	*20.44%*
Caing	HR@1	HR@5	HR@10	ND@1	ND@5	ND@10
BPR	0.3546	0.5728	0.6774	0.3546	0.4699	0.5038
GRU4Rec	0.4633	0.7713	0.8541	0.4633	0.6314	0.6586
Caser	0.5999	0.7964	0.8514	0.5999	0.7057	0.7235
GRU4Rec+	0.6150*	0.8139*	0.8615*	0.6150*	0.7237*	0.7393*
WE3CN	0.4992	0.6622	0.7329	0.4992	0.5861	0.6089
DNA	**0.6220**	**0.8391**	**0.9033**	**0.6220**	**0.7401**	**0.7609**
Imp.	*1.14%*	*3.10%*	*4.85%*	*1.14%*	*2.26%*	*2.92%*

ND represents NDCG. The bolded and starred are the best results and best baseline results. Imp is the improvements of DNA to the starred.

TABLE III
THE RESULTS OF ABLATION STUDY.

Model	DNA_1	DNA_2	DNA_3	DNA_4	DNA_5	DNA_6	DNA
Module	None	S	E	SE	SEN-T	N	SEN
HR@5	0.8057	0.8194	0.8216	0.8223	0.8254	0.8348	**0.8391**
ND@5	0.7077	0.7152	0.7177	0.7180	0.7214	0.7317	**0.7401**

ND represents NDCG. S, E, N and T represent sentence-, element-, news-level attention and time factor respectively. The bolded are the best result.

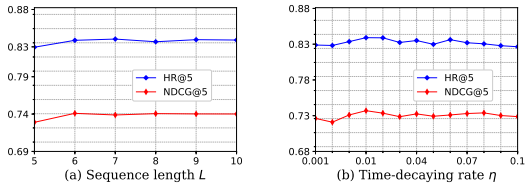


Fig. 2. Impacts of hyper-parameters L and η on performance.

C. Impacts of Attention

In this subsection, we discuss the impacts of the hierarchical attention layers. We present results on Caing with HR@5 and NDCG@5, shown in Table III. We can observe that: (1) Without any attention modules, DNA_1 performs the worst. (2) With sentence-, element- and news-level attention respectively, DNA_2, DNA_3 and DNA_6 improve performance by (1.70%, 1.06%), (1.97%, 1.41%) and (3.61%, 3.39%) on two metrics compared with DNA_1. (3) With both sentence- and element-level attention, DNA_4 has a better performance. (4) With sentence-, element- and news-level attention simultaneously, DNA reaches the best performance. The performance of DNA_6 is close to DNA and this may be because users' discriminations of news are more evident than sentences and elements. (5) Without time-decaying factor based on DNA, DNA_5 reduces performance by (1.63%, 2.53%) on two metrics, slightly weaker than GRU4Rec+ on NDCG@5.

We come to the conclusion that the sentence-, element- and news-level attention modules all improve performance significantly, and the time-decaying factor is also effective.

D. Case Study

In this subsection, we present case studies for better understanding our model. We manually summarize and translate the contents in Table IV and V. The results show that the hierarchical attention layers ensure certain explainability [15].

1) Sentence- and Element-level Attention: We randomly sample a pair of historical news and candidate news and obtain weights by a trained model. Table IV shows the sentences s_1 to s_4 of the historical news, with weights, and the text of the candidate news c^*. From the results of sentence-level, sentence s_4 has the largest weight because s_4 and c^* are both about gold medals of speed skating.

From the results of element-level, the *location* element is assigned the maximum weight 0.276 due to the same location Russia. The *keywords* element has the second largest weight 0.199 because of their almost identical keywords, i.e., Winter Olympics, speed skating and gold medal. The weight of *person* element 0.180 ranks the third because Zhang Hong and Li Jianrou are close in the word space.

This shows that the sentence- and element-level attention mechanisms assign large weights to sentences and elements that are consistent with the candidate news.

2) News-level Attention: As shown in Table V, we randomly sample a user who has a preference for economic news. We extract sequential news c_1 to c_7 from his logs and randomly sample candidate news c_1^* to c_3^* from the categories he has read. "Cat." means news category and "Int./h" means the time interval in hour between the reading time of each news and actual next one. We feed each pair of the user's reading sequence and candidate news into four trained models and get the news-level weights, shown in Fig. 3.

Fig. 3 (a) shows the results of the first model that utilizes content relevance, time-decaying factor and user preferences simultaneously. Fig. 3 (b) shows the results without time factor. The first model assigns larger weights to recently reading news than this model. Fig. 3 (c) shows the results without news representation $\mathbf{x}^*$ in (2). The first model obtains larger weights between news articles of the same category than this model and captures the content relevance. Fig. 3 (d) shows the results without user embedding $\mathbf{u}_i$ in (2). The first model assigns larger weights to economic news articles, which are aligned with the user preference, than this model.

This indicates that news articles which are temporally close and content-relevant to the candidate news and consistent with user preferences are assigned large weights.

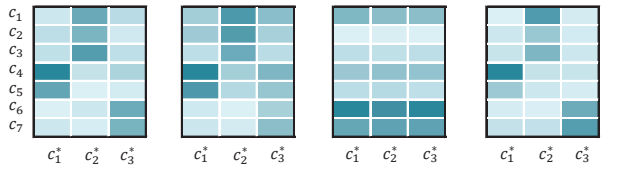
TABLE IV
CASE STUDY OF THE SENTENCE-LEVEL ATTENTION.

No.	Sentences and News Summary	Weight
s_1	The 22nd Winter Olympics have closed in Russia.	0.063
s_2	The speed skating produced 11 records.	0.085
s_3	China won 3 golds, 4 silvers and 2 bronzes.	0.048
s_4	Zhang Hong won the gold medal in the speed skating.	0.100
c^*	At the Winter Olympics, in the short track speed skating finals, Li Jianrou won China's first gold medal.	-

TABLE V
CASE STUDY OF THE NEWS-LEVEL ATTENTION.

No.	News Summary	Cat.	Int./h
c_1	The CSRS approved the issuance of preferred stock.	Eco	24.09
c_2	The Federal Reserve scaled back quantitative easing.	Eco	24.07
c_3	The RMB depreciation will pierce the estate bubble.	Eco	24.06
c_4	The inspection team asked to review a corruption case.	Pol	24.03
c_5	The discipline inspection committee set up an office.	Pol	24.02
c_6	A Malaysia airlines plane with 239 people lost contact.	Soc	0.04
c_7	The Malaysia airlines plane lost contact for 24 hours.	Soc	0.02
c_1^*	Nine officials in Hainan committed violations of law.	Pol	
c_2^*	The release of deposit rates is likely to be realized.	Eco	-
c_3^*	The sea search for MH370 continued on day 17.	Soc	

Eco, Pol and Soc represent economics, politics and society respectively.



(a)w/ all (b)w/o time factor (c)w/o content relevance (d)w/o user preference

Fig. 3. Attention visualization of news-level. The darker color represents the larger weight.

V. RELATED WORK

Traditional news recommendation methods can be divided into three categories: Content-based methods recommend news based on content similarity [16], collaborative filtering methods utilize users' feedback [17] and hybrid methods combine the two strategies [18]. Recently, deep learning has shown its superior performance, such as Recurrent neural network (RNN) is exploited for modeling sequential news-reading behaviors [2], [19], [20] and attention network is applied to get users' dynamic representations [21]. Moreover, [22] proposes a reinforcement learning framework for on-line news scenario. Sequential recommendation aims to predict the next item by exploiting historical information as a sequence [23]. Markov chain [24] and RNN [14] are widely used for the task. Recently, CNN is also adopted, e.g., convolution filters are utilized to capture sequential patterns [11], [25].

VI. CONCLUSIONS

We propose DNA, a dynamic news recommender based on hierarchical attention network. We design a two-layer attention network that learns different impacts of previously interacted contents on users' actions. The time-decaying factor and CNN are incorporated to model dynamism and sequential information of news-reading behaviors. The results of extensive experiments show that DNA achieves a superior performance. In the future, we are to explore more time functions and utilize more information, *e.g.*, knowledge graph, into our model.

ACKNOWLEDGMENTS

This work is supported in part by National Key R&D Program of China 2018YFB1700403 and NSFC U1636210&61421003. For any correspondence, please refer to Shuai Ma.

REFERENCES

[1] F. Garcin, C. Dimitrakakis, and B. Faltings, "Personalized news recommendation with context trees," in *RecSys*, 2013.
[2] K. Park, J. Lee, and J. Choi, "Deep neural networks for news recommendations," in *CIKM*, 2017.
[3] D. Khattar, V. Kumar, V. Varma, and M. Gupta, "Weave&rec: A word embedding based 3-d convolutional network for news recommendation," in *CIKM*, 2018.
[4] J. Li, J. Li, and J. Tang, "A flexible topic-driven framework for news exploration," in *Proceedings of KDD*, 2007.
[5] F. Hamborg, C. Breitinger, M. Schubotz, S. Lachnit, and B. Gipp, "Extraction of main event descriptors from news articles by answering the journalistic five w and one h questions," in *JCDL*, 2018.
[6] W. Wang, "Chinese news event 5w1h semantic elements extraction for event ontology population," in *WWW(Companion Volume)*, 2012.
[7] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014, pp. 1188–1196.
[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.
[9] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *SIGIR*, 2017.
[10] L. Li, L. Zheng, F. Yang, and T. Li, "Modeling and broadening temporal user interest in personalized news recommendation," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3168–3177, 2014.
[11] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018.
[12] J. A. Gulla, L. Zhang, P. Liu, Ö. Özgöbek, and X. Su, "The adressa dataset for news recommendation," in *Proceedings of the International Conference on Web Intelligence*, 2017, pp. 1042–1048.
[13] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009.
[14] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.
[15] X. Chen, Y. Zhang, and Z. Qin, "Dynamic explainable recommendation based on neural attentive models," in *AAAI*, 2019, pp. 53–60.
[16] Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang, "Learning to model relatedness for news recommendation," in *WWW*, 2011.
[17] A. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *WWW*, 2007.
[18] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, "Scene: a scalable two-stage personalized news recommendation system," in *SIGIR*, 2011.
[19] S. Okura, Y. Tagami, S. Ono, and A. Tajima, "Embedding-based news recommendation for millions of users," in *SIGKDD*, 2017.
[20] G. de Souza P. Moreira, F. Ferreira, and A. M. da Cunha, "News session-based recommendations using deep neural networks," in *DLRS*, 2018.
[21] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *WWW*, 2018.
[22] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *WWW*, 2018.
[23] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in *WSDM*, 2018.
[24] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010.
[25] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *WSDM*, 2019.