Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration

Xuejian Wang[†], Lantao Yu[†], Kan Ren[†], Guanyu Tao[‡], Weinan Zhang[†], Yong Yu[†], Jun Wang[♯*]

[†]Shanghai Jiao Tong University, [‡]ULU Technologies Inc., [♯]University College London
{xjwang,yulantao,kren,wnzhang,yyu}@apex.sjtu.edu.cn, guanyu.tao@ulu.ai, j.wang@cs.ucl.ac.uk

ABSTRACT

As aggregators, online news portals face great challenges in continuously selecting a pool of candidate articles to be shown to their users. Typically, those candidate articles are recommended manually by platform editors from a much larger pool of articles aggregated from multiple sources. Such a hand-pick process is labor intensive and time-consuming. In this paper, we study the editor article selection behavior and propose a learning by demonstration system to automatically select a subset of articles from the large pool. Our data analysis shows that (i) editors' selection criteria are *non-explicit*, which are less based only on the keywords or topics, but more depend on the quality and attractiveness of the writing from the candidate article, which is hard to capture based on traditional bag-of-words article representation. And (ii) editors' article selection behaviors are *dynamic*: articles with different data distribution come into the pool everyday and the editors' preference varies, which are driven by some underlying periodic or occasional patterns. To address such problems, we propose a meta-attention model across multiple deep neural nets to (i) automatically catch the editors' underlying selection criteria via the automatic representation learning of each article and its interaction with the meta data and (ii) adaptively capture the change of such criteria via a hybrid attention model. The attention model strategically incorporates multiple prediction models, which are trained in previous days. The system has been deployed in a commercial article feed platform. A 9-day A/B testing has demonstrated the consistent superiority of our proposed model over several strong baselines.

KEYWORDS

Recommendation; Learning by Demonstration; Attention Models; Convolutional Neural Network

1 INTRODUCTION

As the wide adoption of high bandwidth mobile networks such as 4G, mobile news portals or news feed services including social media posts [17] and news articles [6, 34] have gained significant attention. Such textual content feed services or news portals are commonly presented in a cascade-form of user interface (UI) and interactively learn each user's interest and further provide personalized content for them [54]. Notable examples include BuzzFeeds in United States, which serves more than 200 million unique users monthly in 2016 [36], and Toutiao in China, which has 600 million users in total and 66 million daily active users in 2016 [39].

Typically, there are two stages of news article filtering in those systems. In the first stage, professionally trained editors select articles manually, that they think are of high quality, from a huge amount of crawled or submitted articles and in the second stage, user personalized recommender systems deliver relevant articles to each end user with machine learning models based on user data collection [26, 28]. So far, extensive researches have been conducted in the second user-oriented stage [11, 27, 40]. However, little attention has been made on how these articles are gathered as candidates within the platform first. For example, in the article feed platform that we studied in this paper, each of the editors needs to read more than 1,000 articles per day, which is a labor intensive and time-consuming work. In this paper, we aim to alleviate the platform editors' working load by automating the manual article selection process and recommending a subset of articles that fits the human editors' taste and interest as illustrated in Figure 1.

We do this by learning through the limited demonstration from the human editors[1]. Specifically, each editor tries to be objective to perform a judgement on whether an article should be passed to today's candidate set, which will be further picked by the personalized recommender system to deliver to different end users. Thus it is feasible to regard the editor team as a whole article filter and learn a model to select articles to fit their hidden criteria. Such a judgement process seems easy to be automated by training a binary classifier based on the text content. However, the underlying criteria for the editors' selection is *non-explicit* and is hardly just based on the keywords or topics of the article. Instead, it highly depends on the writing styles, such as attractiveness and stringency, which is hidden and hard to capture from the traditional bag-of-words article representation [2, 30] or unsupervised topic models [15, 43]. For example, in our test, a well-tuned naive Bayes text classifier

[*] X. Wang and L. Yu contribute equally and share the co-first authorship. W. Zhang is the corresponding author.

[1] In commercial production, the proposed automation of article selection will not completely replace the editors' work but alleviate their working load. The editors' demonstration will constantly guide the learning system.

KDD '17, August 13-17, 2017, Halifax, NS, Canada
© 2017 ACM. 978-1-4503-4887-4/17/08…$15.00
DOI: 10.1145/3097983.3098096

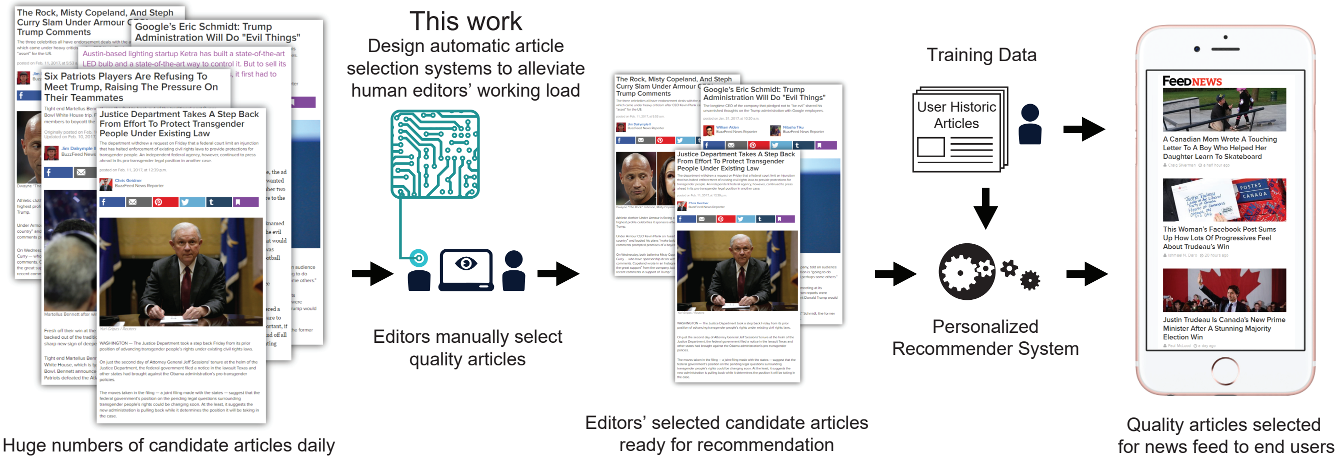
This work
Design automatic article selection systems to alleviate human editors' working load
Huge numbers of candidate articles daily
Editors manually select quality articles
Editors' selected candidate articles ready for recommendation
Training Data
User Historic Articles
Personalized Recommender System
Quality articles selected for news feed to end users
FeedNEWS
A Canadian Mom Wrote A Touching Letter To A Boy Who Helped Her Daughter Learn To Skateboard
This Woman's Facebook Post Sums Up How Lots Of Progressives Feel About Trudeau's Win
Justin Trudeau Is Canada's New Prime Minister After A Stunning Majority Election Win

Figure 1: An illustration of the process that human editors select quality articles from a huge pool and then the recommender systems will do the personalized recommendation for the end users on the selected candidate pool. Our work aims to automate the quality article selection process to alleviate the editors' working load.

[29] can only attain a fair 60% AUC performance in our deployed commercial platform for the binary prediction of whether an editor will select an article or not.

The second challenge lies in that the crawled or submitted article data distribution and the editors' article selection behavior on the data are *non-stationary*. Articles with different data distributions come into the pool everyday and the editors' preference also varies significantly, which could be driven by some underlying periodic or occasional patterns. As we will see in Figures 4 and 5, the incoming article volume, editors' overall selection ratio, the distribution of the source organization of the total and selected articles all vary over time.

In this paper, we propose a Dynamic Attention Deep Model (DADM) to address above two problems for the editor article recommendation task. Specifically, DADM uses character-level text modeling [21] and convolutional neural networks (CNNs) [25] to effectively learn the representation of each article, which captures words interaction patterns that are helpful to predict the editors' selection behavior. More importantly, to handle diverse variance of editors' selection behavior, we introduce attention-based network architecture that would dynamically assign influence factors on recent models based on the current article and these models' timeliness, aiming to improve the performance of the current recommender system. Specifically, we propose to incorporate two kinds of factors into the attention model:

- **Model profile** is a latent factor associated with each day's individual model, depicting which types of articles the model is capable of predicting the editors' selection behavior. For example, the Monday model would be more helpful on predicting a financial article, while the Friday model would be more leveraged to predict an entertainment article.
- **Time profile** is a latent factor associated with each relative day, which tells how fast the model prediction on some type of articles would get out-of-date. For example, the editors' selection criteria on financial articles varies dramatically while that on academic articles varies slightly across time.

With such two types of factors, the hybrid attention model is capable of adaptively allocating attention to previously trained individual models and make a prediction with a lower variance.

Our proposed model has been deployed in ULU Technologies article filtering service rendered by application program interfaces (APIs). A 9-day online A/B testing has been conducted, where at the end of each day the editors return their article selection to demonstrate the learning of the system, and the system learns to predict the editors' article selection during the next day. DADM significantly reduces training efforts by dynamically reusing recent learned models. The results demonstrate higher than 90% AUC performance of our proposed DADM and its consistent performance gains over several strong baselines.

In the rest of the paper, we will discuss related work in Section 2 and present our model in Section 3. The experiments and the corresponding results will be given in Section 4. Finally, we will conclude this paper and discuss the future work in Section 5.

2 RELATED WORK

2.1 Recommender Systems

There are two main solutions to recommender systems, namely, collaborative filtering (CF) based and content based (CB) recommendation [9, 44]. CF methods [22, 37, 51] focus on revealing the patterns in the historic user-item interaction data [47] and make recommendations based on the assumption that users with similar behavior like similar items, without considering any attributes or content of the items. However, these methods face the cold-start problem and may perform poor on sparse history logs [54]. On the other hand, content-based recommendation [4, 24, 32] fully depends on the item attributes or content thus has no cold-start problem but may not provide personalized recommendation.

By contrast, the goal in this paper is to recommend articles for professional editors by learning from textual contents and their side information, e.g., author, original media, author city etc., which needs to combine these two types of data to derive general models for capturing non-trivial useful patterns. In addition, most articles are news, which have very short time span, with very sparse user interaction data. As such, content based recommendation techniques are more suitable for our work than CF ones.

In terms of tasks, our recommendation problem is different compared to the conventional content-based recommendation or collaborative filtering based recommender systems. In our case, the 'users' are a group of professional editors facing abundant articles to manually classify or select. Thus, it is an aggregated interest rather than individual, personalized interest studied by the most mainstream recommender systems.

2.2 Deep Learning for Text Representation

Due to its adequate model capability and the support of big data, deep neural network (DNN) has achieved a huge success in computer vision [23], speech recognition [16] and natural language processing [8, 38] during the recent five years. Neural network language models (NNLM) [7] provide a general solution for text distributed representation learning [31]. Ranging from character level [21] to word level [31, 33], the embedding models trained by back-propagation from the higher-layer neural work makes the text representation of high flexibility and effectiveness. Further techniques are adopted for constructing high level representations of sentences [20] and documents [8]. These NNLM methods have shown very promising performance for capturing semantic and morphological relatedness [21]. In this paper, we adopt character-level NNLM for low-level textual feature learning.

For textual classification and recommendation tasks, deep learning also delivers convincing performance [5, 14, 20, 44, 53]. In [44], the authors proposed a hierarchical Bayesian model to jointly perform deep representation learning for the textual information and collaborative filtering for the rating matrix. Deep recurrent neural networks are utilized in [5] to encode the text sequence into a latent vector and trained end-to-end on the collaborative filtering task. However, these methods require a large number of user-item samples which are not guaranteed in our task and the variance of editors' selection criteria is an issue within these models. To our knowledge, there has little work in leveraging CNNs [21] as the representation learner for articles and based on that performing article recommendation.

2.3 Attention-based Models

Attention is a mechanism to flexibly selecting the reference part of context information, which can facilitate global learning [3, 45]. Attention model was originally proposed in machine translation tasks to deal with the issue for encoder-decoder approaches that all the necessary information should be compressed into the fix-length encoding vector [3]. Soon after the use on language, attention model is leveraged on image caption task [45] where the salient part of an image is automatically detect and based on that the model could generate high-quality description of the image. Then, the attention model is leveraged in various tasks. The authors in [49] utilized attention to capture hierarchical patterns of documents from word to sentence and finally to the whole document. The authors in [48] took attention on question text and extracted the semantically related parts between question-answer pairs. Other attention-based work includes natural language parsing [42] and text classification [50]. In these work, attention has been used as a textual level mechanism for modeling interactions within different parts of the contents. In order to capture the dynamics of editors' selection behavior, we build our new attention-based architectures, which will dynamically take the effects from recent knowledge to the current model. To our knowledge, it is the first work utilizing attention to model varying user preferences.

3 METHODOLOGY

3.1 Problem Definition

We formally define the problem as below. We have a set of articles gathered from multiple sources at a given time t as $\mathbb{D}_t = \{d_1^t, ..., d_i^t, ...d_{N_t}^t\}$, where $t \in \{1, ..., T\}$ and $N_t = |\mathbb{D}_t|$ is the number of articles gathered at time t. In the past, we have observed that the human editors have selected a subset of articles as relevant ones: $\mathbb{S}_t \subset \mathbb{D}_t$ at each timestamp from $t-1$, $t-2$, ...1, where $M_t = |\mathbb{S}_t|$ and $M_t \ll N_t$. Now the task is, at the current time t, to automatically obtain a new subset $\mathbb{S}_t$ from the new pool $\mathbb{D}_t$ gathered. The objective is to make the predicted set $\mathbb{S}_t$ as close as possible to the one if we had asked the human editors to select.

To make the problem manageable, we assume the decision whether a document is recommended or not is independent on the other documents decisions in the same day although the documents could be correlated. The rationale behind the idea is that like text classification, the model is trained to capture the underlying patterns among documents and then is used to predict the label of each document independently. Thus we could simplify the problem by predicting the selection probability $\Pr(y_i|d_i^t; \mathbb{D}_t)$ of the professional editor taking the specific *action* y_t of selecting for public readers the given article d_i^t; and we then choose top-K documents as the chosen set. In the dataset, each document's features $d_i^t = \{\boldsymbol{x}, \boldsymbol{i}\}$ consist of textual content $\boldsymbol{x}$ and categorical meta-data $\boldsymbol{i}$, e.g. source website, article categories, authors, etc.

The above problem is a unique one. On one hand, at a specific time t, it is an unsupervised problem as the document pool $\mathbb{D}_t$ is disjoint from the previous and we don't have any label at that time timestamp. But, on the other hand, it is also a supervised binary prediction task as we have human editors' labels for previous time $t-1, ..., 1$. Thus, we need to transfer or combine the knowledge of recently learned models from the previous timestamps and improve the overall performance of the current model.

There are three challenges for the modeling and learning of the above problem. (i) The editors' selection criteria for rich textual content is non-explicit, such as the attractiveness or stringency of the writing, which could be hidden within some deep interaction among word sequences. One needs a general model to capture such underlying patterns, which is an open research topic in natural language processing [19] and content-based recommendation [6, 30]. (ii) The meta-data is another issue since its sparse property and categorical data type [35, 52]. Learning this hybrid style of data remains unsolved in the community of data mining [12, 44]. Moreover, the editors' preference may vary over time, which implies some trends and periodic patterns among daily dataset. So that the third challenge is (iii) to dynamically capture varying preferences for better model generalization. Specifically, the final objective is to combine the knowledge of recent learned models and perform a robust prediction.

To solve these problems, we propose our DADM model constituted by three main parts: (i) We take CNN-based method to extract a general representation of the document; (ii) Motivated by wide & deep model [12], we combine the linear model and our CNN part together to jointly model the sparse categorical meta-information and the sequential categorical data (textual content of the document). (iii) To adaptively capture the editors' dynamic behavior, we propose an attention model over multiple deep networks which jointly considers the speciality and the timeliness of each model trained in previous days.

3.2 Text Representation Learning

To model the textual content of the document, traditional methods including bag-of-words features [2, 30], e.g. TF-IDF feature or naive Bayes and unsupervised learning objective [15, 43], e.g. topic models, are based on counting statistics which ignore word orders and suffer from sparsity and poor generalization performance. Considering the semantic relatedness between different words, we implement a convolutional neural network architecture to model the text. Moreover, in order to generalize for different languages, we construct the CNN based on character level since not all the languages have explicit "word" specification. For example, Chinese needs word segmentation which requires much specific domain knowledge [46].

Figure 2 illustrates the architecture of our CNN component which is motivated by [21]. We will introduce the architecture with a bottom-up approach.

The raw textual input of the document is represented as a sequence of characters $\boldsymbol{x} = \{c_1, c_2, \dots, c_l\}$ of length l, where the i-th entry c_i is one of the elements in the character set C. We define $\mathcal{E} \in \mathbb{R}^{d \times |C|}$ as the set of character embeddings where d is the predefined dimension of each embedding vector.

First of all, by concatenating the corresponding character embedding $\mathbf{e}_i \in \mathbb{R}^d$, provided by the embedding function $\Pi : c \longrightarrow \mathbf{e} \in \mathcal{E}$, we build the document matrix $\mathbf{D} \in \mathbb{R}^{d \times l}$.

Secondly, we apply a convolution operation on $\mathbf{D}$ with a kernel $\mathbf{K}_j \in \mathbb{R}^{d \times w}$, $j \in [1, J]$ among the total J kernels of width w [25], to obtain the feature map $\mathbf{m}_j$ as

$$\mathbf{m}_j[i] = f(\mathbf{D}[*, i : i + w - 1] \odot \mathbf{K}_j + \mathbf{b}_j)\,, \tag{1}$$

where $i \in [1, l - w + 1]$ is the iteration index and $\mathbf{m}_j \in \mathbb{R}^{l-w+1}$, while f is the non-linear transformation function such as the hyperbolic tangent (tanh) $f(z) = (\exp(z) - \exp(-z))/(\exp(z) + \exp(-z))$ and operation $\odot$ is the Frobenius inner product between two matrices. Now we obtain a feature map matrix $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k] \in \mathbb{R}^{(l-w+1) \times k}$.

Thirdly, the max-over-time pooling [13] is used on the column of the feature map matrix such that

$$\mathbf{x} = \max \mathbf{M}[*, p]\,,\ p \in [1, k]\,, \tag{2}$$

where the pooling output $\mathbf{x} \in \mathbb{R}^k$ is the learned representation of the textual content $\mathbf{D}$.

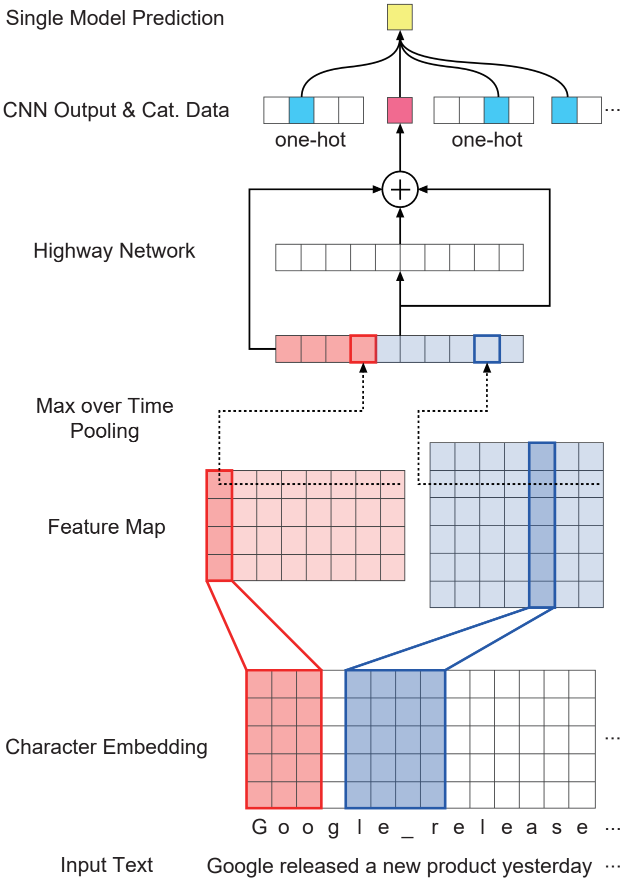

Figure 2: CNN architecture for text prediction.

The final part of the CNN model is a highway network defined as:

$$\begin{aligned}
\eta &= \sigma(\mathbf{W}_q^H \cdot \mathbf{x}_q + \mathbf{b}_q^H)\,, q \in [1, n]\,, \\
\mathbf{x}_{q+1} &= \eta \cdot g(\mathbf{W}_q \cdot \mathbf{x}_q + \mathbf{b}_q) + (1 - \eta)\mathbf{x}_q\,, \\
o_{\mathbf{D}} &= \sigma(\mathbf{w}_{\mathbf{D}} \cdot \mathbf{x}_n + \mathbf{b}_{\mathbf{D}})\,,
\end{aligned} \tag{3}$$

where $\sigma(z) = 1/(1 + \exp(-z))$ is the sigmoid function and the final output $o_{\mathbf{D}}$, as the extracted features of the textual content, will be fed into later prediction. $g(\cdot)$ represents the operation of each highway net layer in the CNN model. Here η and $(1 - \eta)$ plays the role of "transform gate" and "carry gate" respectively, which controls the information carried by the highway layer from input to the output [21, 41].

The reason for the adoption of CNN model with one convolutional layer and a total of 1050 kernels on the textual content is that convolutional operation and max-pooling technique can be leveraged to capture the underlying semantic patterns within the word sequence, which are helpful for the prediction but may not be explicit to be specified. Recent literatures [21, 21, 38] have shown that CNN-based model can achieve promising performance comparative with or even more competitive than other deep models, e.g. Long Short Term Memory (LSTM) [18] in many NLP tasks.

3.3 Multi-view Categorical Data Modeling

As previously described, each document is combined with two parts of information: textual content and categorical meta-information. We apply a wide & deep infrastructure [12] to jointly model these two types of data as illustrated in the top layer of the architecture in Figure 2. The difference is that we adopt a base model with CNN architecture to represent the textual content, which has been presented above, while [12] only reuses the categorical meta-information to both the deep part and the linear part.

We use one-hot representation for the categorical meta-information of each article. Specifically, the field set $\mathcal{S}$ contains three fields: authors, source organizations and original websites. For the s-th field in $\mathcal{S}$, we take binary vector $\mathbf{o}_s \in \mathbb{R}^{l_s}$, where only the value of the column corresponding to the presented category is 1 and the value of other columns is 0. l_s is the total number of the possible category values taken in the s-th field.

Thus we obtain a hybrid feature representation $\mathbf{o}$, which combines the one-hot categorical vectors and numerical CNN output as

$$\mathbf{o} = [\mathbf{o}_1^\top \oplus \mathbf{o}_2^\top \oplus, \ldots \oplus \mathbf{o}_{|\mathcal{S}|}^\top \oplus o_{\mathbf{D}}^\top]^\top \in \mathbb{R}^{l_1+l_2+\ldots+l_{|\mathcal{S}|}+1} , \tag{4}$$

where $\oplus$ is the concatenation operator between vectors.

After all these operations, we utilize a logistic regression to predict the final probability over the model as

$$\hat{y} = \Pr(\mathrm{y}_i | d_i^t; \mathbb{D}_t) = \sigma(\mathbf{w} \cdot \mathbf{o} + b) . \tag{5}$$

3.4 Dynamic Attention Deep Model

Here we present our dynamic attention deep model (DADM) over multiple networks, which plays a key role for capturing editors' dynamic behavior patterns to make the final prediction. The basic idea is from twofold considerations:

- **Model Speciality**: as the data distribution of the incoming candidate articles is different across days, the correspondent trained model has different speciality. For example, there would be a higher portion of news articles on finance on Monday than that on Saturday, thus for an incoming article about finance, it is likely that the Monday model is more helpful to predict the editors' preference than the Saturday model.
- **Timeliness**: the editors' behavior may vary over time since overabundance may cause disturbing and it might also repeatedly present preference over daily fed corpus from recent experiences. More importantly, for different types of articles, their timeliness could be highly different. For example, the news articles on the latest event would be easily out-of-date while a research article on a scientific finding would be attractive for a longer time.

In order to incorporate these two aspects, we consider a twofold attention solution. Specifically, we deal with speciality of each recent model i using a factor vector $\mathbf{w}_i^{\mathcal{M}} \in \mathbb{R}^{l_1+l_2+\ldots+l_{|\mathcal{S}|}+1}$ and the article timeliness for the model trained on day t using a factor vector $\mathbf{w}_t^{\mathcal{T}} \in \mathbb{R}^{l_1+l_2+\ldots+l_{|\mathcal{S}|}+1}$. Based on such two factors, we build the model of allocating attention over recently trained prediction models.

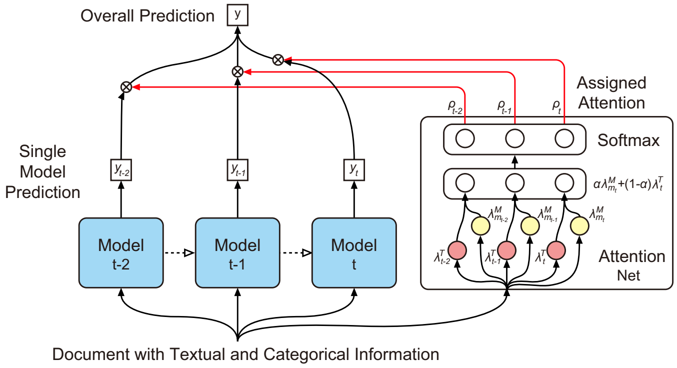

Figure 3: Dynamic attention over multiple deep nets.

The attention model (DADM) architecture is shown in Figure 3. We formulate the DADM in a *softmax* form as below. Specifically, denote the model trained on day t as m_t, we have the assigned attention to the model m_t as

$$\begin{aligned} \lambda_{m_t}^{\mathcal{M}} &= \mathbf{w}_{m_t}^{\mathcal{M}} \cdot \mathbf{o} + b_{m_t}^{\mathcal{M}} , \\ \lambda_t^{\mathcal{T}} &= \mathbf{w}_t^{\mathcal{T}} \cdot \mathbf{o} + b_t^{\mathcal{T}} , \\ \rho_t &= \mathrm{softmax}(\alpha\lambda_{m_t}^{\mathcal{M}} + (1-\alpha)\lambda_t^{\mathcal{T}}) \\ &= \frac{\exp(\alpha \cdot \lambda_{m_t}^{\mathcal{M}} + (1-\alpha) \cdot \lambda_t^{\mathcal{T}})}{\sum_{\tau \in [0,K]} \exp(\alpha \cdot \lambda_{m_\tau}^{\mathcal{M}} + (1-\alpha) \cdot \lambda_\tau^{\mathcal{T}})} \end{aligned} \tag{6}$$

In the above equations, the model speciality is formulated as $\lambda_{m_t}^{\mathcal{M}}$, where $b_{m_t}^{\mathcal{M}}$ is the overall effectiveness term and the inner product term $\mathbf{w}_{m_t}^{\mathcal{M}} \cdot \mathbf{o}$ further captures the model's capability on predicting the specific article representation $\mathbf{o}$. Similarly, the article timeliness is formulated as $\lambda_t^{\mathcal{T}}$, where $b_t^{\mathcal{T}}$ is the overall timeliness of the patterns in day t to the current prediction day and the inner product term $\mathbf{w}_t^{\mathcal{T}} \cdot \mathbf{o}$ further models the relative timeliness of the specific article representation $\mathbf{o}$.

Overall, $\{\mathbf{w}_{m_t}^{\mathcal{M}}, \mathbf{w}_t^{\mathcal{T}}, b_{m_t}^{\mathcal{M}}, b_t^{\mathcal{T}}\}_{t=t_0-K+1,\ldots,t_0}$ is the set of parameters to train for our attention model. α is the hyperparameter to control the impact of the two factors, and K is the attention day parameter representing maximal distance to the current date.

Finally, we obtain the attention-based probability estimation as

$$\hat{y} = \Pr(y_i | d_i^{t_0}; \mathbb{D}_{t_0}) = \sum_{\tau=t_0-K+1}^{t_0} \rho_\tau \cdot \hat{y}_\tau . \tag{7}$$

The additional advantage of our attention model lies in saving training efforts since DADM leverages the learned knowledge from previous models while traditional method needs to train on the full past history to attain competitive performance [12].

4 EXPERIMENTS

In this section, we present our experiments and the corresponding results, including data analysis about the dynamics of the article distribution and the editors' selection behavior. We also make some discussions about the hyperparameter tuning in the ablation study.

4.1 Experimental Setup

We have conducted experiments based on ULU Technologies article filtering API platform. ULU Technologies is a startup team based

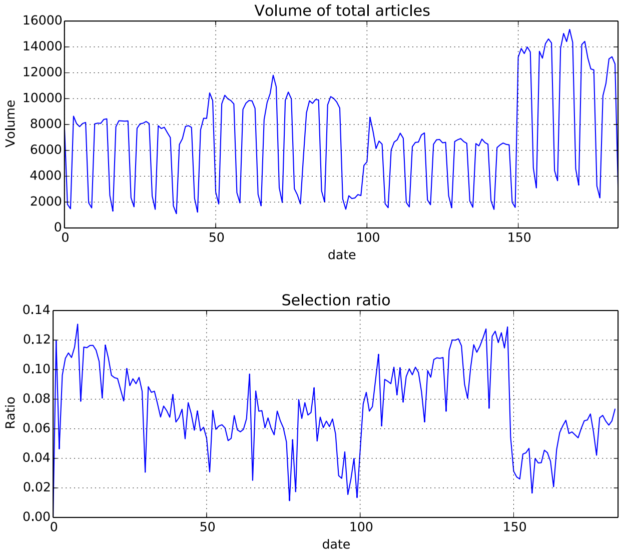

Figure 4: Dynamic characteristics of the dataset. Above: The change of the number of total submitted articles over time. Below: The change of the editors' selection ratio over time.

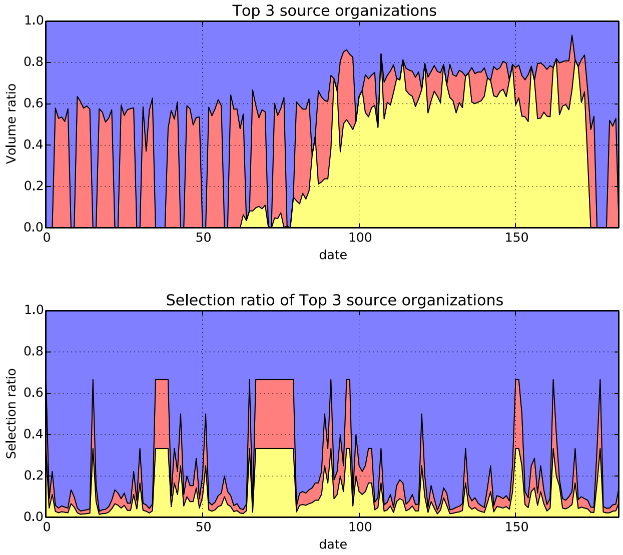

Figure 5: Dynamic characteristics of 3 main source organizations. Each of the 3 areas in the figures represents the proportion of their total volume (above) and selected articles (below).

in Beijing, working on artificial intelligence based Platform-as-a-Service (PaaS) for media partners, including the API services of article recommendation in web pages and mobile news feeds, text mining, sentiment analysis, article filtering and native advertising. By Jun. 2017, ULU platform serves more than 30 medias with 37 million daily page views and 20 million daily active users. The platform API links to the article data and editors' selection interfaces of an anonymized large Chinese finance article feeds platform. The model was deployed in a 3-node cluster with Tensorflow (TF) [1] based on CUDA 7.5 using single GPU, GeForce GTX 1080 with 8GB VRAM.

The dataset is a large collection of quality article selection demonstration with average length of 900 characters over six months, manually created by professional editors. As is shown in Figure 4, both of the total number of given articles and selection ratio on each day vary over time, which indicates that the data volume and editors' selection board line vary significantly. Furthermore, as is shown in Figure 5, we also keep track of the total volume and the selection ratio of the top three organizations consuming the largest proportion of the dataset, which underlines the drastic drift in the distribution of the given articles and the variability of editors' selection criteria every day.

Each data instance of the dataset can be represented as a tuple $(\boldsymbol{d}, y)$, where y is the binary feedback of editors' selection and $\boldsymbol{d}$ is the combination of the article with textual content $\boldsymbol{x}$ and categorical meta-information $\boldsymbol{i}$ including author, source organization and original website. For exploiting meta-information effectively, we discard the authors with low frequency(less than 3 times). For preprocessing the textual content in the tensor form using GPU acceleration, we pad (clip) the shorter(longer) articles to the same length of 100.

4.2 Compared Settings

To investigate the selection performance of quality candidate articles, we compare our proposed DADM with three models in our experiments.

LR-meta is the logistic regression model, which is based on the categorical meta-information formulated with one-hot encoding.

CNN-text is the powerful convolutional neural network for text representation and classification, which focuses on the textual information.

W&D is the widely-used wide&deep model (discussed in Sections 3.2 and 3.3) which leverages both of the two aspects of information, where the deep neural network learns the text representation and a logistic regression component takes the learned text representation and categorical meta-information as inputs and performs the final prediction task. Note that, our W&D model takes one more step than traditional concept [12] since our model consumes different structure of data into different components rather than feeding the same data source into both the wide part and the deep part.

DADM is our proposed dynamic attention deep model as discussed in Section 3.4.

To evaluate the candidate pool recommendation quality of the models, we use the widely-used measurements of Precision, Recall, F1 score and Area under the Curve of ROC (AUC) as the evaluation metrics. For the threshold selection of Precision and Recall measurements, we choose the one to maximize the F1 score, which can be interpreted as a weighted average of the precision and recall, since it is more reasonable to take both into consideration. Thus

Table 1: The data statistics over 9 tested days.

Date	Articles	Selected	Authors	Websites	Orgs.
10-01	2,233	126	499	84	211
10-02	1,449	41	299	62	178
10-03	2,494	66	200	65	200
10-04	2,275	101	365	65	190
10-05	2,319	36	407	68	194
10-06	2,582	67	412	75	186
10-07	2,504	100	488	69	193
10-08	4,837	65	974	122	560
10-09	5,109	228	1,088	137	594
Overall	25,802	830	4,732	747	2,506

Table 2: Quality articles recommendation performance comparison.

Model	AUC	F1	Precision	Recall
CNN-text	0.777±0.052	0.186±0.079	0.170±0.077	0.253±0.143
LR-meta	0.807±0.055	0.255±0.107	0.221±0.118	0.376±0.148
W&D	0.833±0.049	0.284±0.091	0.220±0.094	0.484±0.187
DADM	**0.853±0.036**	**0.317±0.079**	0.258±0.059	0.451±0.202

we mainly compare the models' performance on AUC and F1 while the precision and recall serve as the auxiliary evaluation metrics.

4.3 Results and Discussions

First, Table 1 shows the data statistics over the 9 tested days, i.e., Oct. 1-9, 2016. Since Oct. 1-7 is Chinese national holiday, the data volume and distribution in such a period is different with the later two days, which makes the data even more dynamic than other period. In addition, the number of article source organization (Orgs.) each day is more than 50% of the number of authors, which means the authors are distributed in a variety of organizations and thus the text writing style could be much diverse. Due to business constraints, we could only perform a 9-day A/B testing, which is considered as a sufficiently long period for a full-volume A/B testing for the compared algorithms on a commercial platform.

In Table 2, we report the overall performance of recommending quality articles over a time period of 9 days. We can observe that the proposed wide&deep model successfully utilizes both the textual content and categorical meta-information and achieves better classification performance over both AUC and F1 metrics than LR-meta and CNN-text, which only utilize one aspect of the information. Such a result also indicates the effectiveness of using categorical meta-information which contains fields of authors, organizations, etc. Furthermore, we can see the obvious impact of DADM over the strong W&D due to the dynamic attention mechanism which adaptively and smartly takes previous knowledge into consideration to capture the dynamics of the editors' preference. Moreover, as the F1 score is a weighted average of both precision and recall and thus provides more comprehensive evaluation of the model, which the selection threshold tries to maximize, DADM tends to emphasize the precision more and W&D is just the opposite.

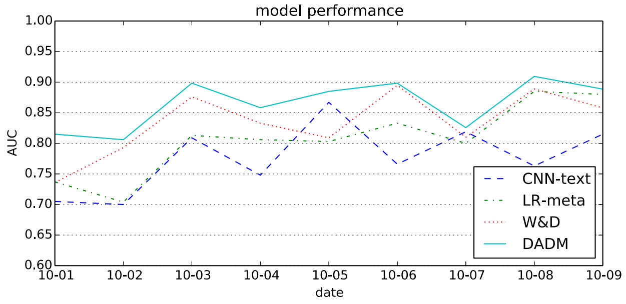

Figure 6: AUC performance of quality article recommendation over 9 days (Oct. 1-9, 2016).

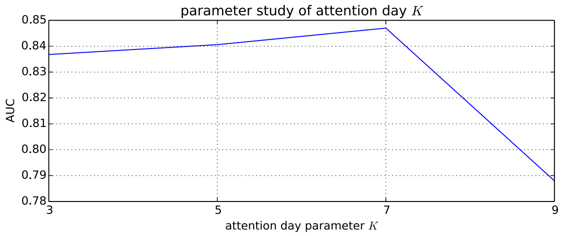

Figure 7: AUC performance against the number of attention days of DADM.

It should be noted that in our problem, with the same F1 score, precision is usually more meaningful than recall because the article volume is always too large for the editor team to check even the recommended subset of articles. For example there are 10^6 articles coming into the large pool, among which 10^5 articles are qualified to be selected into the small pool for recommending to end users. But the news feed platform requires that every delivered article should be checked by the editor team, which can check at most 10^4 articles per day. In such a case, the recommended 10^4 scale articles should be with a high precision, no matter the recall could be as low as 10%.

Figure 6 presents the AUC performance of recommending the quality candidate articles for each of the 9 tested days. As can be observed clearly, the DADM consistently outperforms all compared methods over 9 days, which indicates the feasibility of using attention mechanism to capture the dynamics of editor's selection behavior and leveraging recent knowledge to improve current model. Furthermore, as the proposed hybrid attention model adaptively allocates attention to previously well-trained individual models, DADM is capable of making more accurate decisions with lower variance. This is because the attention mechanism can be viewed as a smart and adaptive ensemble of the several well-trained models for each specific data entry. For example, when the current model is incapable of performing accurate classification for some specific kind of data entry, attention mechanism will strategically combine the potentially useful recent knowledge from previously well-trained model to perform better prediction, which increases the robustness and stability significantly.

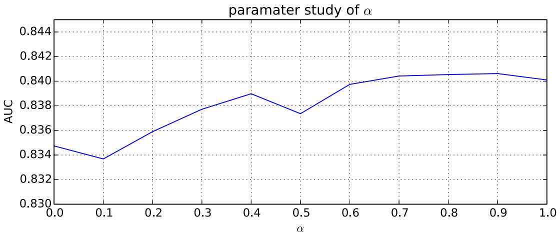

Figure 8: AUC performance against α in Eq. (6).

Furthermore, we have an ablation study about the hyperparameters of DADM. Figure 7 shows the AUC performance of DADM against difference numbers of attention days. We can observe that the empirical optimal attention day number is 7, which is intuitive since the weekly patterns are obvious in our scenario, as shown in Figures 4 and 5. The prediction on this Tuesday's articles should make use of the editors' demonstration data on last Tuesday. Further previous model may not be that helpful because of the timeliness of the learned patterns.

Figure 8 shows the AUC performance against different values of α in Eq. (6). As can be observed, the empirically optimal α is around 0.8, which means model specialty plays a more important role in the attention allocation than the timeliness. Neither of the extreme cases of $\alpha = 0$ or $\alpha = 1$ is the best, which means both types of attention factors should be consider in such a hybrid attention model.

5 CONCLUSION

In this paper we have proposed a dynamic attention deep model to deal with the problems of non-explicit selection criteria and non-stationary data in the editors' article selection stage of content recommendation pipeline. For each single model, we leverage the CNNs and wide model to automatically learn the editors' underlying selection criteria; for attention assignment over multiple models trained in previous days, we incorporate both the model specialty factor and model timeline factor into the attention network to strategically assign attentions given each specific article. The experiments were conducted over a commercial API platform linking to a Chinese finance article feeds platform. A 9-day online A/B testing has shown that our proposed dynamic attention deep model performs the best in terms of both prediction AUC and F1 score as well as the low variance in handling the dynamic data and editors' behavior. For the future work, we will consider the influence of the article images on the editors' selection behavior, which could be an effective feature [10]. On the modeling aspect, we plan to further investigate the learning scheme of the whole hierarchical network since we found the learning behaviors of the CNN (deep net) and the logistic regression (shallow net) are different [12]. Also we will study how our recommendations influence the editors' further actions since their observed data is 'biased' due to our provided article ranking. It is likely that the exploitation-exploration techniques would be leveraged to handle such a problem [54].

ACKNOWLEDGEMENT

The work done by Shanghai Jiao Tong University is financially supported by Shanghai Sailing Program (17YF1428200) and National Natural Science Foundation of China (61632017).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, and others. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*.

[2] Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 19–28.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Marko Balabanović and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. *Commun. ACM* 40, 3 (1997), 66–72.

[5] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the GRU: Multi-task Learning for Deep Text Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 107–114.

[6] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 195–202.

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[8] Zsolt Bitvai and Trevor Cohn. 2015. Non-Linear Text Regression with a Deep Convolutional Neural Network.. In *ACL (2)*. 180–185.

[9] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.

[10] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep CTR Prediction in Display Advertising. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 811–820.

[11] Ting Chen, Wei-Li Han, Hai-Dong Wang, Yi-Xun Zhou, Bin Xu, and Bin-Yu Zang. 2007. Content recommendation system based on private dynamic user profile. In *Machine Learning and Cybernetics, 2007 International Conference on*, Vol. 4. IEEE, 2112–2118.

[12] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[13] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.

[14] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very Deep Convolutional Networks for Text Classification. In *European Chapter of the Association for Computational Linguistics EACL'17*.

[15] Prem K Gopalan, Laurent Charlin, and David Blei. 2014. Content-based recommendations with poisson factorization. In *Advances in Neural Information Processing Systems*. 3176–3184.

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.

[17] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. 2010. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 194–201.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[21] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615* (2015).

[22] Yehuda Koren, Robert Bell, Chris Volinsky, and others. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[24] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*. 331–339.

[25] Yann LeCun, Yoshua Bengio, and others. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[26] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[27] Ting-Peng Liang, Hung-Jen Lai, and Yi-Cheng Ku. 2006. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems* 23, 3 (2006), 45–70.

[28] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[29] Andrew McCallum, Kamal Nigam, and others. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Vol. 752. Citeseer, 41–48.

[30] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. 2002. Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai*. 187–192.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[32] Raymond J Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 195–204.

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[34] Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 385–388.

[35] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. *arXiv preprint arXiv:1611.00144* (2016).

[36] Quantcast. 2017. buzzfeed.com Traffic. https://www.quantcast.com/buzzfeed.com#trafficCard. (2017).

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization.. In *Nips*, Vol. 1. 2–1.

[38] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 373–382.

[39] Feliz Solomon. 2016. The Owner of This Hot Chinese App Is Seeking a $10 Billion Valuation. http://fortune.com/2016/11/08/china-toutiao-media-tech-uber-weibo-bytedance/. (2016).

[40] Jeong-Woo Son, A Kim, Seong-Bae Park, and others. 2013. A location-based news article recommendation with explicit localized semantic analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 293–302.

[41] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*. 2377–2385.

[42] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. 2773–2781.

[43] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[44] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[45] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.. In *ICML*, Vol. 14. 77–81.

[46] Nianwen Xue and others. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* 8, 1 (2003), 29–48.

[47] Diyi Yang, Tianqi Chen, Weinan Zhang, Qiuxia Lu, and Yong Yu. 2012. Local implicit feedback mining for music recommendation. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 91–98.

[48] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 287–296.

[49] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*. 1480–1489.

[50] Qi Zhang, Yeyun Gong, Jindou Wu, Haoran Huang, and Xuanjing Huang. 2016. Retweet Prediction with Attention-based Deep Neural Network. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 75–84.

[51] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 785–788.

[52] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European Conference on Information Retrieval*. Springer, 45–57.

[53] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. 649–657.

[54] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 1411–1420.