Leveraging Title-Abstract Attentive Semantics for Paper Recommendation

Guibing Guo,[1] **Bowei Chen,**[1] **Xiaoyan Zhang,**[2] **Zhirong Liu,**[3] **Zhenhua Dong,**[3] **Xiuqiang He**[3]
[1] Northeastern University, China, [2] Shenzhen University, China, [3] Noah's Ark Research Lab, Huawei, China
guogb@swc.neu.edu.cn , boweichen_public@outlook.com, {liuzhirong, dongzhenhua, hexiuqiang1}@huawei.com

Abstract

Paper recommendation is a research topic to provide users with personalized papers of interest. However, most existing approaches equally treat title and abstract as the input to learn the representation of a paper, ignoring their semantic relationship. In this paper, we regard the abstract as a sequence of sentences, and propose a two-level attentive neural network to capture: (1) the ability of each word within a sentence to reflect if it is semantically close to the words within the title. (2) the extent of each sentence in the abstract relative to the title, which is often a good summarization of the abstract document. Specifically, we propose a Long-Short Term Memory (LSTM) network with attention to learn the representation of sentences, and integrate a Gated Recurrent Unit (GRU) network with a memory network to learn the long-term sequential sentence patterns of interacted papers for both user and item (paper) modeling. We conduct extensive experiments on two real datasets, and show that our approach outperforms other state-of-the-art approaches in terms of accuracy.

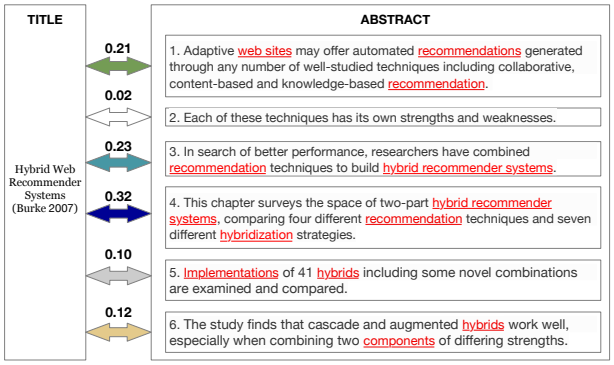

Figure 1: The semantic relationship between title and abstract taken from (Burke 2007). Words highlighted in red are semantically close to those in title. The weight (importance) of each sentence is shown above the arrow.

Introduction

Ever-increasing number of research papers have been published over the last decades, resulting in a problem known as 'information overload'. Researchers have to spend more time searching for articles articles they are interested in. Therefore, paper recommendation is more important than before. Collaborative filtering (CF) has been widely adopted in recommendation systems, which explores user-item historical interactions (e.g., purchase). However, CF often generates poor performance since the user-item interaction matrix is very sparse in many fields. Thus, auxiliary information is introduced to enhance recommendation performance.

In recent years, many approaches have been proposed to exploit various auxiliary information. For paper recommendation, two types of auxiliary information are widely adopted for better recommendations, including structural and textual information. The former type refers to paper citation relationships, i.e., papers that a paper cites or those cite it (Sugiyama and Kan 2010; 2013; Mohammadi et al. 2016). The structure of paper citations may indicate the influence of one paper to another, but it ignores the real content and semantics of the paper. (Hassan 2017) points out another issue that some newly published papers may not be cited and some researchers prefer to cite their own less relevant papers. The latter type tries to make use of title and abstract to better represent a paper, for example, by extracting its topics (Wang, Wang, and Yeung 2015; Kim et al. 2016). Some researchers also integrate keywords and domain of a paper as well as user profiles to generate better recommendation lists (Sun et al. 2013). Our paper follows this research direction that uses textual information, which refers to title and abstract of papers in our case, for paper recommendation. We opt not to use full paper text for recommendation since it contains a lot of noise and may deteriorate the overall performance. We leave it as a part of future work for further exploration.

However, most existing text-based approaches suffer from the limitation that they simply treat title and abstract of a paper without any difference, failing to grasp the semantic relationship between them. In our viewpoint, we regard the title as a conclusive and informative sentence relative to the whole abstract document, and thus combining both of them can better capture the semantics of a specific paper. To

explain, we try to analyze a paper in both word level and sentence level, as illustrated in Figure 1. Specifically, in the word-level part, we presume that words in title are the best indicator to reflect the research topic of a paper. We thus put more weights on these informative words when constructing the representation of a paper's abstract sentences. Take the fourth sentence in the abstract as an example, the highlighted words 'hybrid', 'recommendation' and 'systems' are more representative than other words, given the fact that they offer more semantic information that is consistent with the title. Then, in the sentence-level part, we note that each sentence in the abstract has various degrees of ability to reflect the semantic meaning of the paper. For example, the second sentence in Figure 1 is a general statement that may appear in many other papers. Apparently, this sentence has little contribution to grasp paper's topic and semantics. In contrast, the fourth sentence is the most significant one because it elaborates on the main idea of the paper in question.

To resolve these issues, in this paper we propose a **T**itle-**A**bstract **A**ttentive **S**emantic (or **TAAS** for short) network to capture the semantic relationship between title and abstract for paper recommendation. It consists of two attentive sub-networks, namely word-level and sentence-level attentive sub-networks. We treat the abstract as a sequence of sentences, and regard the title as a conclusive sentence for the abstract. To be specific, in the word-level sub-network, we propose an attentive Long-Short Term Memory (LSTM) network to learn sentence representation by considering the importance of a word (in an abstract sentence) with respect to those in the title. In the sentence-level sub-network, we integrate a Gated Recurrent Unit (GRU) network [1] with a memory network seamlessly to capture the relationship between the title and each sentence in the abstract. In this way, we will construct fine-grained user preference by capturing the sequential sentence patterns.

Our main contributions are summarized as follows:

- We propose a novel approach TAAS to capture the semantic relationship between title and abstract in both word level and sentence level. To the authors' best knowledge, this is the first model to take into account their semantic relationship for paper recommendation.
- To learn fine-grained user preference, we present a key-value memory network to memorize the user preference for sequential sentence patterns on the basis of title representation, which overcomes the shortcoming that traditional GRU networks prone to forgetting effective memory. In addition, the title embedding is used as global memory consecutively updated by abstract sentences, i.e., to learn user's long-short interactive preference.
- We have conducted extensive experiments on two real-world datasets (citeulike-a, PRSDataset) to evaluate the effectiveness of our approach. The experimental results show that our model outperforms several state-of-the-art approaches in terms of ranking accuracy.

[1] In word-level attentive sub-network, the LSTM network outperforms the GRU network in terms of recommendation accuracy, while in sentence-level attentive sub-network, the GRU network works better than LSTM network for recommendation

The TAAS Model

In this section, we will first present the general framework and show how to seamlessly integrate the two attentive sub-networks. Then, the detailed structure of these sub-networks will be illustrated, especially on how to capture the semantic relationship between title and abstract. By doing so, we can construct fine-grained user profiles in our model.

General Framework

Our TAAS model is a two-level attentive recommendation neural network, which mainly focuses on learning the semantic relationship between title and abstract of a paper in both word level and sentence level, so as to learn a fine-grained user profile. For the sake of discussion, we will introduce a number of notations. Suppose we have N users in the user set U, and M papers in the paper (item) set I. The symbols $\boldsymbol{u}_i$ and $\boldsymbol{i}_j$ are used to denote the embedding vectors of the user *i* in U and item *j* in I, respectively. The general framework of our approach is illustrated in Figure 2. We devise two attentive sub-networks to learn user preference vector $\boldsymbol{m}_i$ based on sequential sentence patterns, as well as item content vector $\boldsymbol{c}_j$ derived from the textual information in title and abstract. Taking the same combination method suggested by (Chen et al. 2018), the feature representations of user $\boldsymbol{p}_i$ and item $\boldsymbol{q}_j$ can be formulated as follows.

$$\boldsymbol{p}_i = \boldsymbol{u}_i + \alpha \boldsymbol{m}_i \tag{1}$$

$$\boldsymbol{q}_j = \boldsymbol{i}_j + \beta \boldsymbol{c}_j \tag{2}$$

where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are parameters to indicate the importance of semantic embeddings $\boldsymbol{m}_i$ and $\boldsymbol{c}_j$. In the experimentation, we will tune these two parameters to study the effect of semantic relationship between abstract and title.

Ranking Function. After optimizing the above semantic weight parameters, for every user *i*, we calculate the estimated ranking score $\hat{x}_{ij}$ for each candidate item *j* in order to generate a ranking list of items. The following inner product of two vectors is taken as the ranking function.

$$\hat{x}_{ij} = \boldsymbol{p}_i^\top \boldsymbol{q}_j \tag{3}$$

Objective Function. To generate a top-k paper recommendation, we adopt a well-known pair-wise personalized ranking objective function (Rendle et al. 2009) to train our model. The purpose is to ensure that users will have stronger preference on interacted items than those without any interactions, given by:

$$\arg\min_{\Theta} \sum_{(i,j,k) \in D_s} -\ln \sigma(\hat{x}_{ij} - \hat{x}_{ik}) + \lambda \|\Theta\|^2 \tag{4}$$

where $\sigma(\cdot)$ and Θ denote the sigmoid function and all the parameters to learn, respectively; $\| \cdot \|$ is the Frobenius norm. The training set D_s consists of many tuples (i, j, k), implying user *i* interacted with item (paper) *j* but not with *k*.

Model Workflow. The workflow of our model is shown in Figure 2. Suppose user i has a preference for paper j. We first extract word embedding $\boldsymbol{w}_{\cdot,x}$ from the *x-th* word in the title, and word embedding $\boldsymbol{w}_{y,x}$ from the *x-th* word in the *y-th* abstract sentence. For simplicity, we omit the symbol j

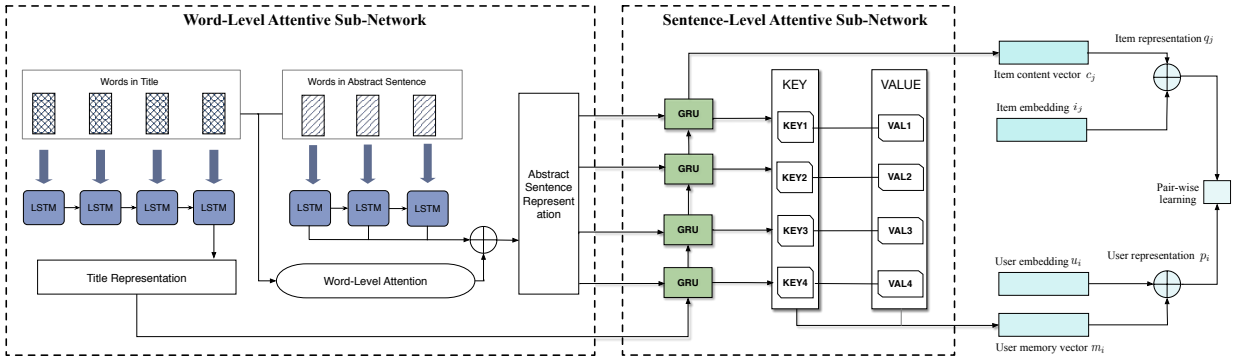

Figure 2: The architecture of our TAAS framework, which mainly consists of two sub-networks, namely word-level attentive sub-network (left) for sentence embeddings and sentence-level attentive sub-network (middle) for preference learning.

since they all belong to the same paper. The extracted word embeddings from title and abstract sentences will be taken as sequential input to the word-level attentive sub-network, which produces the feature representation of title $\boldsymbol{t}$ and abstract sentence $\boldsymbol{a}_y$, i.e., the *y-th* sentence of abstract. Specifically, these word embeddings will be used to compute the relative importance (attention) of abstract words by comparing them with title words. The underlying assumption is that words in the abstract have more importance if they are semantically close to those in the title.

Next, we denote abstract as a sequential set of sentences: $A = \{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_n\}$, where n is the number of sentences in the abstract. We take the set of abstract sentences A and title $\boldsymbol{t}$ as input to the sentence-level attentive sub-network, where the significance of each abstract sentence relative to the title is taken into account. The output of the sub-network includes: (1) the content vector $\boldsymbol{c}_j$ for paper j, representing the item features learned from abstract and title. (2) the preference vector $\boldsymbol{m}_i$ for user i, representing the user features learned from sequential sentence patterns. Thus, we can obtain the user and item representation by Equations 1 and 2, respectively. Finally, our TAAS model can be trained by optimizing the objective function by Equation 4.

Word-Level Attentive Sub-Network

This sub-network aims to capture the semantic relationship between words in each abstract sentence and those in title, whereby providing title-aware sentence representation. We adopt a pre-trained Word2Vec model[2] to retrieve the embedding vectors for all the words in the title and abstract. The sequential positions of words in a sentence are important features to learn a sentence representation. Hence, we devise a LSTM network to model the word sequences. Specifically, the LSTM network contains a chain of repeating modules with a cell state in each module to store important sequential information, which will be updated at every time step. In addition, it is capable of accommodating variable-length sequences, which is suitable for our word sequences of title and abstract. We denote the set of title words as $\overline{W} = \{\boldsymbol{w}_{\cdot,1}, \boldsymbol{w}_{\cdot,2}, ..., \boldsymbol{w}_{\cdot,n}\}$, and the set of abstract words as $W_y = \{\boldsymbol{w}_{y,1}, \boldsymbol{w}_{y,2}, ..., \boldsymbol{w}_{y,m}\}$ in *y-th* sentence, where n and m are the length of title and abstract sentence, respectively. These word embeddings are subsequently input to two separate LSTM networks (but sharing same weights θ), which will update the current hidden state vector $\boldsymbol{h}_t$ according to the previous state $\boldsymbol{h}_{t-1}$ and the new input word vector.

$$\overline{\boldsymbol{h}}_{t'_w} = \text{LSTM}(\overline{\boldsymbol{h}}_{t'_{w-1}}, \boldsymbol{w}_{\cdot,t'_w}, \theta) \tag{5}$$

$$\boldsymbol{h}_{t_w} = \text{LSTM}(\boldsymbol{h}_{t_{w-1}}, \boldsymbol{w}_{y,t_w}, \theta) \tag{6}$$

where $\overline{\boldsymbol{h}}_{t'_w}$ and $\boldsymbol{h}_{t_w}$ represent the outputs of the two LSTM networks at time steps t'_w and t_w, respectively. The network is trained to learn network parameters θ.

To effectively grasp the correlations between words in title and those in abstract, we design an attention mechanism to learn the relative importance of each sentence word, whereby the embedding of abstract sentence can be represented more accurately. Specifically, for each word in an abstract sentences, we compute the correlation by inner products between its embedding and every title word's embedding, yielding the cumulated similarity score $\text{sim}(\boldsymbol{w}_{y,x}, \boldsymbol{t}')$:

$$\text{sim}(\boldsymbol{w}_{y,x}, \boldsymbol{t}') = \sum_{j=1}^{n} \boldsymbol{w}_{\cdot,j}^{\top} \cdot \boldsymbol{w}_{y,x} \tag{7}$$

where $\boldsymbol{t}' = (\boldsymbol{w}_{\cdot,1}, \boldsymbol{w}_{\cdot,2}, ..., \boldsymbol{w}_{\cdot,n})$ denotes a sequence of title words, n is the length of title. In general, the higher score indicates the stronger correlation of a sentence word relative to title. Then, we can obtain the weight of each word in an abstract sentence by normalizing the similarity score via a softmax function, given by:

$$\alpha_x = \frac{\exp(\text{sim}(\boldsymbol{w}_{y,x}, \boldsymbol{t}'))}{\sum_{j=1}^{m} \exp(\text{sim}(\boldsymbol{w}_{y,j}, \boldsymbol{t}'))} \tag{8}$$

where m is the length of abstract sentence. Then, we can aggregate all the hidden state of words in the sentence to obtain its feature representation $\boldsymbol{a}_y$, given by:

$$\boldsymbol{a}_y = \sum_{x=1}^{m} \alpha_x \boldsymbol{h}_x \tag{9}$$

[2]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

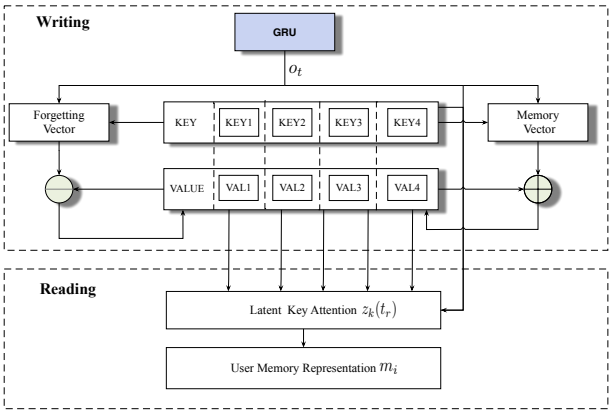

Figure 3: The detailed realization of the Key-Value Memory Network. It contains two operations: reading and writing.

Note that the title representation can be retrieved by the final state of the LSTM network $\overline{\boldsymbol{h}}_n$.

$$\boldsymbol{t} = \overline{\boldsymbol{h}}_n \tag{10}$$

Finally, we take title $\boldsymbol{t}$ and abstract sentences A as input to the sentence-level attentive sub-network.

Sentence-Level Attentive Sub-Network

The goal of this sub-network is to grasp the importance of each abstract sentence with respect to the title, constructing fine-grained user preference and paper representation correctly. We regard title as a sentence that can best summarize a paper, while sentences in abstract provide rich semantic information to sequentially explain the detailed meanings or implication of the title. Specifically, we introduce a GRU network, where the title representation $\boldsymbol{t}$ is used to initialize global memory. We contend that it is important to carefully initialize the network weights, whereby an RNN-based network can be easier to train and receive better performance. Then, we proceed to keep updating the global memory when new abstract sentence arrives at time step t_s.

$$\boldsymbol{o_0} = \boldsymbol{t} \tag{11}$$

$$\boldsymbol{o}_{t_s} = \text{GRU}(\boldsymbol{o}_{t_{s-1}}, \boldsymbol{a}_{t_s}, \bar{\theta}) \tag{12}$$

where $\boldsymbol{o}_{t_s}$ indicates the hidden state of the GRU network at time step t_s and $\bar{\theta}$ is the network parameters to learn.

Although a GRU network can capture the semantic correlation between title and abstract sentences to some extent, it still suffers from two possible shortcomings. First, it considers the semantic relationships between different abstract sentences and title in the same latent space, which may ignore the potential differences of abstract sentences in semantics. Second, as the sequential sentences keep arriving, the GRU network may discard long-term preferences of users, and only retain the recent memories. To resolve these issues, inspired by (Chen et al. 2018), we integrate our GRU network with a Key-Value Memory Network (KV-MN), from which a user vector can be derived to reflect the long-term preferences. Specifically, KV-MN decomposes each memory slot into a key vector and a value vector. An advantage of KV-MN is that we can associate multiple latent key vectors with their value vectors, where each key represents a semantic aspect of the paper in question. Therefore, given key vectors, we can read and merge their value vectors accordingly to construct a user preference.

Suppose there are K latent keys in our KV-MN, the whole network shares K embedding key vectors in the latent key set $F = \{\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_k\}$. For user i, we define her corresponding K memory value vectors as a set of $V_i = \{\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, ..., \boldsymbol{v}_{ik}\}$. Note that the value vectors are varying from person to person. The input of the KV-MN is the hidden state of GRU network, representing the accumulated knowledge about sequential sentence patterns at a certain time step. The memorizing process of the KV-MN contains two main operations, i.e., *reading* and *writing* as shown in Figure 3.

Reading. In the reading operation, we utilize multiple latent key vectors to match their own value vectors, then we sum them up according to their weights of importance to the user preference. Specifically, we first define the attention score $r_k(t)$ of key feature $\boldsymbol{f}_k$ in a time step t, given by:

$$r_k(t) = {\boldsymbol{o}_t}^\top \cdot \boldsymbol{f}_k \tag{13}$$

Then we can calculate the importance of each semantic space relative to the user preference by a softmax function:

$$z_k(t) = \frac{\exp(\gamma r_k(t))}{\sum_{f \in F} \exp(\gamma r_f(t))} \tag{14}$$

where γ is the strength parameter to tune. Now, we can derive the user preference vector $\boldsymbol{m}_i$ by summing the value vectors with the importance weight of each semantic space $z_k(t_r)$ in the last time step t_r, defined by:

$$\boldsymbol{m}_i = \sum_{k=1}^{K} z_k(t_r) \cdot \boldsymbol{v}_{ik} \tag{15}$$

Writing. The value vectors for user i will be updated in the current time step t_c to memorize effective semantic information of the paper as well as forgetting useless memories. The ratio g of memory forgetting is calculated as follows:

$$g = \sigma({\boldsymbol{E}}^\top \boldsymbol{o}_{t_c} + b_g) \tag{16}$$

where E and b_g are the forgetting parameters to study. Thus, we update the value vectors by considering the importance of each semantic space $z_k(t_c)$:

$$\boldsymbol{v'_{ik}} \leftarrow \boldsymbol{v}_{ik} - \boldsymbol{v}_{ik} \odot (z_k(t_c) \cdot g) \tag{17}$$

where $\odot$ indicates the element-wise product. After that, the memory vector can be computed by taking a normalization function, followed by the update of user's value vector:

$$y_{t_c} = \tanh({\boldsymbol{H}}^\top \boldsymbol{o}_{t_c} + b_y) \tag{18}$$

$$\boldsymbol{v}_{ik} \leftarrow \boldsymbol{v'_{ik}} + \boldsymbol{z}_k(t_c) \cdot y_{t_c} \tag{19}$$

where H and b_y are the memory parameters to learn. Intuitively, the more importance of a semantic space is, the more influence it imposes on the corresponding value vector.

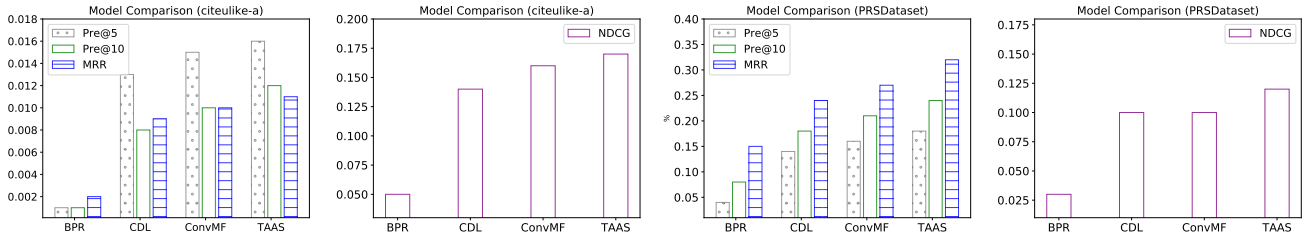

Figure 4: Performance comparison of all the approaches, while the left two subfigures illustrate the results on the citeulike-a dataset, and the right two subfigures are on the PRSDataset.

The output vector of the GRU network represents the item content vector, i.e., $\boldsymbol{c}_j$. The user preference vector $\boldsymbol{m}_i$ can be obtained by applying a reading operation on the output of hidden state at the last time step. Finally, we take the content-based embeddings $\boldsymbol{m}_i$ and $\boldsymbol{c}_j$ as input to the pair-wise ranking objective function for model training.

Experiments

In this section, we study three research aspects of our model: (1) the effectiveness of the proposed approach in comparison with other state-of-the-art methods; (2) the impact of the two-level attentive sub-networks; and (3) the influence of the semantic weight parameters used in our model.

Experimental Setup

Datasets. Two real-world datasets are adopted in our experiments, namely *citeulike-a* and *PRSDataset*. Both datasets contain the information of title and abstract for papers, and the interactions between users and papers. For each dataset, we randomly split it into two subsets, where 80% of user-paper interactions are classified as the training set and the rest 20% as the testing set.

The first dataset *citeulike-a* is extracted from CiteULike[3], and the other dataset *PRSDataset* comes from CSPubGuru[4]. For both datasets, we remove the items with missing and defective abstracts as well as their relative interactions. We also filter out the users who have interactions with at most one item. Finally, the *citeulike-a* dataset is composed of 5548 users, 10987 items (papers), and 134510 user-item pairs. The *PRSDataset* dataset consists of 2453 users, 21940 items, and 35969 user-item pairs. Basic text processing is adopted to remove stop words from title and abstract, as well as the segmented (abstract) sentences with less than 20 characters.

Baselines. We mainly compare our TAAS model with the following state-of-the-art paper recommendation methods.

- **BPR** (Rendle et al. 2009) is a famous pair-wise personalize ranking model based on implicit feedback. Our model takes the same objective function as the BPR, but differs in the formulation of user and item representations.
- **CDL** (Wang, Wang, and Yeung 2015) attempts to combine an auto-encoder neural model (for better item representation based on textual information) and a traditional collaborative filtering method.
- **ConvMF** (Kim et al. 2016) applies a convolution neural network (CNN) to learn the representation of items, and then jointly model user preference by integrating with a traditional matrix factorization model.

Recently, some novel recommendation approaches (Tran, Sweeney, and Lee 2019; Chen et al. 2019; He et al. 2018) have been introduced to deal with different practical recommendation problems when various information is available. Nevertheless, given textual information for papers, the proposed baseline methods are most suitable and representative for paper recommendation compared with other approaches. All the implementation of selected methods is obtained by either the source code from the authors (CDL and ConvMF) or a famous recommendation algorithm library LibRec (Guo et al. 2015) (BPR). In this way, we are able to guarantee all the results of baseline methods are promising and reliable.

Parameter Settings. The best model parameters are either suggested by the original papers or empirically set by our experiments. We have tested the neural batch size in $\{128, 256, 512, 1024\}$, the L2 loss weight in $\{0.1, 0.01, 0.001\}$, and tuned the semantic weight parameters α and β from 0 to 1 stepping by 0.2 and the learning rate in $\{0.1, 0.01, 0.001, 0.0001\}$. Furthermore, we test the number of neurons in each hidden layer and that of latent memory keys from 10 to 100 stepping by 10. All the network parameters are initialized by normal distribution (0, 0.1). We optimize our models with the Adam gradient descent.

Evaluation Metrics. We adopt four widely taken ranking metrics to evaluate recommendation accuracy of all comparison methods, including Precision at n (Pre@n), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulated Gain (NDCG). The detailed definitions of these metrics can be found in (Ricci et al. 2010). Generally, the higher these metrics are, the better performance we can reach.

Performance Comparison with Other Models

Figure 4 presents the recommendation performance of all comparison methods. The results show that our TAAS model can consistently and significantly outperform the other models in all ranking metrics.

[3]http://www.citeulike.org/faq/data.adp

[4]https://sites.google.com/site/tinhuynhuit/dataset

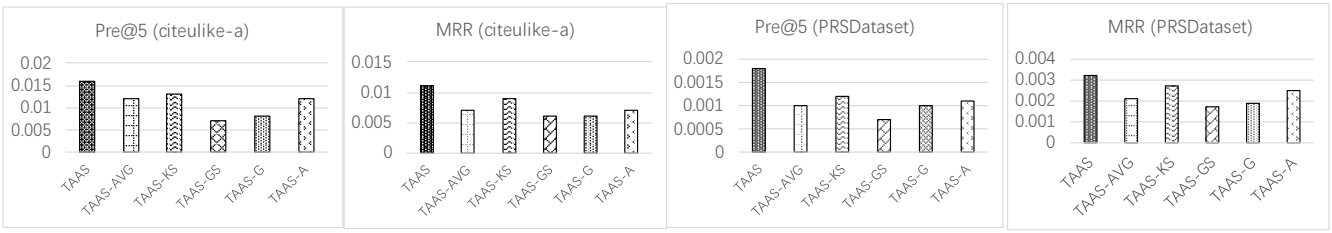

Figure 5: The comparison among variant models of the TAAS.

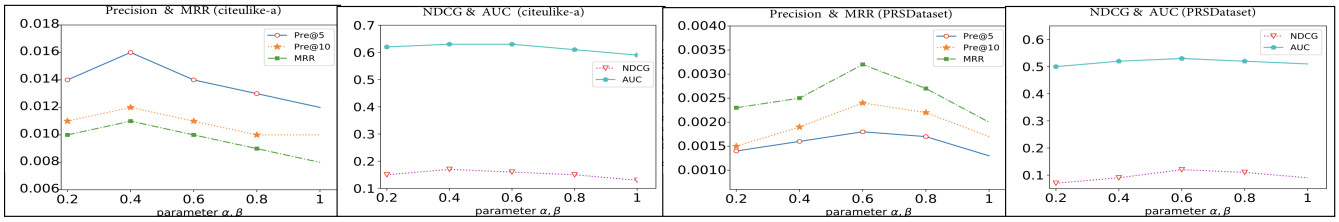

Figure 6: The influence of the semantic information.

As expected, due to high sparsity of our datasets, the most basic approach BPR, which only takes into account the historical interactions of users and items, produces the worst performance among the four approaches. It implies the importance of textual content information (i.e., title and abstract) in improving performance of paper recommendation.

Among all the methods with textual content, ConvMF achieves better performance than CDL. One possible explanation is that the item representation method (i.e., convolution neural network, CNN) of ConvMF is more powerful than that of CDL, which is a relatively shallow model (i.e., an auto-encoder model) *per se*. In other words, we conclude that carefully modeling textual information will lead to better representation of items, and thus enhance recommendation performance eventually.

In addition to extracting features from title and abstract, our TAAS model further captures the semantic relationship between them. The fact that significantly better performance is obtained also helps demonstrate the usefulness of such interactions between title and abstract.

Model Ablation

We conduct extensive experiments to evaluate the effect of the two sub-networks by designing a number of variant TAAS models below.

- **TAAS-AVG** removes the word-level attention framework and uses the average of LSTM output vectors to represent an abstract sentence. The difference lies in the word-level sub-network.
- **TAAS-KS** adopts a pre-trained sent2vec model[5] to get sentence representation. It removes the word-level attentive sub-network.
- **TAAS-GS** further excludes the part of Key-Value Memory Network on the basis of TAAS-KS.
- **TAAS-G** only excludes the part of Key-Value Memory network, but retains the word-level sub-network.
- **TAAS-A** randomly initializes the input of the GRU network instead of inputting the title representation.

The results of these variant models are illustrated in Figure 5, from which we can observe the following insights.

(a) Word-Level Attentive Sub-Network

From Figure 5, it is noted that TAAS and TAAS-G have better performance than TAAS-KS and TAAS-GS respectively. It suggests that sentences in abstract can be modeled more appropriately with the involvement of word correlations between title and abstract sentences. The fact that TAAS outperforms TAAS-AVG also confirms that the word-level attention is quite effective in our model. To sum up, the word-level attentive sub-network provides a better representation of both title and abstract sentences, and it is necessary to leverage the word-level semantic relationship between title and abstract.

(b) Sentence-Level Attentive Sub-Network

A major difference between TAAS and TAAS-A lies in the initialization methods for the GRU network. The better performance of TAAS than TAAS-A reflects that initialization by title representation can capture valuable knowledge, which can be further fine-tuned by the following arrived abstract sentences. Meanwhile, the comparison between TAAS and TAAS-G suggests that key-value memory network is valuable in memorizing long-term sequential sentence patterns for each user, and thus helps generate better user preference modeling. As a summary, both key parts of sentence-level attentive sub-network, i.e., the title representation as initialization, and long-term preference by memory network shade a light on better performance.

(c) Integration of Two-Level Attentive Sub-Network

[5]https://drive.google.com/file/d/0B6VhzidiLvjSa19uYWlLUEkzX3c/view

Since our TAAS model has successfully beat the other variants, it is safe to draw a conclusion that integrating both word-level and sentence-level sub-networks can generate the best recommendation performance by considering the title-abstract semantic relationships.

Effect of Parameters α, β

The semantic weight parameters α, β in Equations 1 and 2 indicate the importance of semantic information in modeling paper and user representations. We tune the parameters from 0 to 1 stepping by 0.2, and the results are given in Figure 6.

The performance trends are quite similar in two datasets. They both get better performance when increasing the values of these parameters, and reach the best performance during (0.4, 0.6). However, further tuning these parameters will greatly decrease the recommendation performance. In other words, a proper combination of intrinsic embedding and semantic embedding (from title-abstract) is likely to better model both user and paper representations.

Related Work

Structure-based Paper Recommendation

The first type of paper recommendation is based on the citation structure, i.e., the papers it cites and those citing it. The constructed paper graph is further mined to calculate paper similarity and generate paper recommendations. For example, (Sugiyama and Kan 2010) construct paper representation based on the TF-IDF technology, which is heavily relied on the term frequency. The similarity between papers based on citation references is used as weights to build user and paper profiles. However, not all relevant works can be fully covered in one paper. To alleviate this issue, (Sugiyama and Kan 2013) further improve their previous model by extending a paper's reference list with the involvement of the top-N relevant papers. Moreover, (Mohammadi et al. 2016) build a basic paper graph based on the reference citations. A random walk algorithm is devised to generate recommendation items. To sum up, the underlying assumption of this research line stresses that the citation topology can accurately reflect paper relatedness. However, in many cases, such an assumption cannot hold because: (1) most recently published papers cannot be referred to by previous papers; (2) some valuable references may be missing due to the unawareness of researchers; and (3) some irrelevant or less relevant papers may be adopted in the reference list, for example, some other papers from the same authors.

Content-based Paper Recommendation

A more straightforward line of research is to model a paper based on its content, and learn user profiles by summarizing the representation of papers. For paper recommendation, the content information refers to the title, keywords, abstract and so on. Many models have been proposed to better understand papers under estimation. For example, (Wang and Blei 2011) propose the collaborative topic regression (CTR) model, which combines a Latent Dirichlet Allocation (LDA) topic model with a probabilistic matrix factorization (PMF) (Salakhutdinov and Mnih 2007) model for better recommendation. However, LDA topic models require rich word contents which may not be available in paper recommendation. The full-text of papers cannot be accessed in open-source datasets, and only relatively short abstract exists in most cases. To tackle this problem, (Wang, Wang, and Yeung 2015; Vincent et al. 2010) aim to enhance PMF by a deep representation of item contents. Furthermore, (Kim et al. 2016) introduce a ConvMF model to capture the contextual information by applying a convolution neural network (CNN) network. They seamlessly integrate the CNN network with PMF, and enhance the recommendation accuracy. Recently, (Hassan 2017) introduce an LSTM network to learn the semantic paper representation, where title and abstract of the paper are used as input to the LSTM network. This work is very relevant to our work in that they also highlight the importance of both title and abstract. However, they do not consider the title-abstract semantic relationship, and provide no realization to verify their basic idea.

Memory Network For Recommendation

To handle sequential data more effectively, memory network (MN) has been proposed to memorize long-term dependencies (Weston, Chopra, and Bordes 2014). MN introduces a memory matrix to store historical information by updating it when new information is available. Recently, MN has been adopted in recommendation system. (Chen et al. 2018) propose a RUM model that integrates the insights of CF with MN, constructing user preference by accommodating long-term historical interactions. (Huang et al. 2018) combine GRU and MN to enhance sequential recommendation, where knowledge base is leveraged to better update the MN. Our TAAS model also integrates MN andb GRU seamlessly to learn title-abstract semantic relationship and long-term sequential sentence pattern of papers.

Conclusions

In this paper, we proposed a two-level attentive neural network called *TAAS* to capture the semantic correlation between title and abstract for paper recommendation. The word-level attentive sub-network aimed to generate sentence representations, the assumption behind which is that words appearing in title are more informative than others. The sentence-level attentive sub-network took the title representation as the global memory, which was then iteratively updated by abstract sentences, and sequentially memorized by a key-value memory network. Our experimental results on two real datasets have confirmed that the proposed method can reach superior performance to other counterparts.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61972078 and 61702084, and by the Fundamental Research Funds for Central Universities under Grant N181705007. It was also partially sponsored by Huawei Innovation Research Program.

References

Burke, R. 2007. *Hybrid Web Recommender Systems*. Springer Berlin Heidelberg.

Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; and Zha, H. 2018. Sequential recommendation with user memory networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*, 108–116.

Chen, C.; Zhang, M.; Wang, C.; Ma, W.; Li, M.; Liu, Y.; and Ma, S. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 225–234.

Guo, G.; Zhang, J.; Sun, Z.; and Yorke-Smith, N. 2015. Librec: A java library for recommender systems. In *Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP)*.

Hassan, H. A. M. 2017. Personalized research paper recommendation using deep learning. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP)*, 327–330.

He, X.; He, Z.; Du, X.; and Chua, T.-S. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, 355–364.

Huang, J.; Zhao, W. X.; Dou, H.; Wen, J.-R.; and Chang, E. Y. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research; Development in Information Retrieval (SIGIR)*, 505–514.

Kim, D.; Park, C.; Oh, J.; Lee, S.; and Yu, H. 2016. Convolutional matrix factorization for document context-aware recommendation (recsys). In *Proceedings of the 10th ACM Conference on Recommender Systems*, 233–240.

Mohammadi, S.; Kylasa, S.; Kollias, G.; and Grama, A. 2016. Context-specific recommendation system for predicting similar pubmed articles. In *IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 1007–1014.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 452–461.

Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P. B. 2010. *Recommender Systems Handbook*. Berlin, Heidelberg: Springer-Verlag, 1st edition.

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS)*, 1257–1264.

Sugiyama, K., and Kan, M.-Y. 2010. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries (JCDL)*, 29–38.

Sugiyama, K., and Kan, M.-Y. 2013. Exploiting potential citation papers in scholarly paper recommendation. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, 153–162.

Sun, J.; Ma, J.; Liu, X.; Liu, Z.; Wang, G.; Jiang, H.; and Silva, T. 2013. A novel approach for personalized article recommendation in online scientific communities. *46th Hawaii International Conference on System Sciences* 1543–1552.

Tran, T.; Sweeney, R.; and Lee, K. 2019. Adversarial mahalanobis distance-based attentive song recommender for automatic playlist continuation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 245–254.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 3371–3408.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 448–456.

Wang, H.; Wang, N.; and Yeung, D.-Y. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1235–1244.

Weston, J.; Chopra, S.; and Bordes, A. 2014. Memory networks. *Computing Research Repository*.