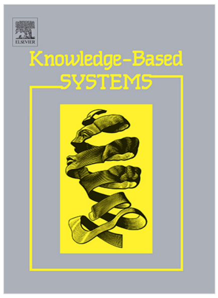
Journal Pre-proof



HASVRec: A modularized Hierarchical Attention-based Scholarly Venue Recommender system

Tribikram Pradhan, Abhinav Gupta, Sukomal Pal

PII: S0950-7051(20)30413-5
DOI: https://doi.org/10.1016/j.knosys.2020.106181
Reference: KNOSYS 106181

To appear in: *Knowledge-Based Systems*

Received date : 30 January 2020
Revised date : 20 June 2020
Accepted date : 22 June 2020

Please cite this article as: T. Pradhan, A. Gupta and S. Pal, HASVRec: A modularized Hierarchical Attention-based Scholarly Venue Recommender system, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.106181.

HASVRec: A modularized Hierarchical Attention-based Scholarly Venue Recommender system

Tribikram Pradhan[a,b,*], Abhinav Gupta[a], Sukomal Pal[a]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, India*
[b] *Department of I&CT, Manipal Institute of Technology (MIT), Manipal Academy of Higher Education (MAHE), Manipal, India*

Abstract

Manually selecting appropriate scholarly venues is becoming a tedious and time-consuming task for researchers due to many reasons that include relevance, scientific impact, and research visibility. Sometimes, high-quality papers get rejected due to mismatch between the area of the paper and the scope of the journal. Recommending appropriate academic venues can, therefore, enable researchers to identify and take part in relevant conferences and publish in journals that matter the most. A researcher may certainly know of a few leading venues for her specific field of interest. However, a venue recommendation system becomes particularly helpful when exploring a new domain or when more options are needed. Due to high dimensionality and sparsity of text data, and complex semantics of the natural language, journal identification presents difficult challenges. We propose a novel and unified architecture that contains Bi-directional LSTM (Bi-LSTM) and Hierarchical Attention Network (HAN) to address the above problems. We call the proposed architecture modularized Hierarchical Attention-based Scholarly Venue Recommender system (HASVRec), which only requires the abstract, title, keywords, field of study, and author of a new paper to recommend scholarly venues. Experiments on the DBLP-Citation-Network V11 dataset exhibit that our proposed approach outperforms several state-of-the-art methods in terms of accuracy, $F1$, nDCG, MRR, average venue quality, and stability.

Keywords: Recommender system, Bidirectional LSTM, Attentive pooling, Deep learning, Hierarchical attention network (HAN)

1. Introduction

Recommender systems are used for recommending different objects to users based on their preferences [1, 2]. Recommender systems that aim to suggest items of potential interest to users are gaining attention from the research community [3]. There are now academic recommendation systems which provide recommendations for collaborators [4], papers [5], citations [6], academic venues [7], reviewers [8], and conferences [9]. These systems have been successful in providing personalized information services to researchers [7]. Among various recommendations these systems provide, an important task is to recommend appropriate publication venues. Researchers intend to publish in academic venues that accept high-quality papers and are relevant to their area of research [10].

The task of academic venue recommendation has become increasingly difficult due to a large number of collaborations amongst disciplines in the research community and frequent changes in the scope of journals [11]. This phenomenon has not only created ample opportunities for researchers but also given birth to new challenges, especially the task of identifying relevant academic journals and conferences [12]. For example, the DBLP[1] dataset, a collection of scientific publication records along with their relationships has 9,585 computer science conferences[2] and more than 4,152 journals[3] [13], making identification of the relevant journals or conferences difficult. Moreover, with time, researchers' interests expand, evolve, or adapt in rapidly changing subject areas, resulting in an increasing need for information on appropriate venues in the changed scenario [14]. An increase in interdisciplinary research areas also poses significant challenges to research institutes and their libraries as they strive to understand information-seeking behaviors and dynamic information needs of the users [11].

On the other hand, researchers also need to know about new venues to remain updated. They usually get updates from colleagues/supervisors, friends, internet, and books, but often the information is not sufficiently comprehensive and appropriate as per their research requirements. The researchers, therefore, sometimes end up submitting their work in inappropriate venues resulting in rejection, delays

*Corresponding author
Email addresses: tpradhan.rs.cse16@itbhu.ac.in (Tribikram Pradhan), abhinav.gupta.cse16@itbhu.ac.in (Abhinav Gupta), spal.cse@itbhu.ac.in (Sukomal Pal)

[1] http://dblp.uni-trier.de/db/
[2] http://dblp.uni-trier.de/db/conf/
[3] http://dblp.uni-trier.de/db/journals/

in publication, and compromise in the quality of the publication.

For scholarly venue recommendations, Collaborative Filtering (CF) based approaches have been quite popular in the last decade. Most of the traditional CF approaches examine the effect of past publication records of researchers to recommend venues [15–17]. However, CF approaches are less effective when there are not enough ratings present in the researcher-venue matrix. The recommendations may not be useful in the case of a new researcher who lacks publication history. Content Based Filtering (CBF) or topic-based models use the authors' profile, the content of their papers, as well as that of the papers published at a specific venue [18, 19] to generate recommendations. However, CBF approaches suffer from limited content analysis, which can significantly reduce the quality of recommendations [20].

To alleviate the problems of the CF and CBF approaches, in recent years, Network-Based (NB) or co-author based approaches have been proposed [7, 21, 22]. Here, a social graph is built among the authors based on co-authorship. Irrespective of the actual content, each paper authored by the same set of authors will receive the same recommendation. Recommendations are very poor for a new researcher who does not have any past publication records. Venues with less popularity among the co-authors of a given author are seldom recommended, although content-wise, they may be appropriate.

Most of the approaches discussed above are unable to capture the semantics and, therefore, most of the time, fail to recommend suitable publication venues. Recently, hierarchical attention-based deep-learning approaches have succeeded in document classification [23, 24], owing to their ability to discover the intricate structure and deep semantics in high dimensional data [25, 26]. Various tasks such as automatic essay scoring (AES) [27], and automatic academic paper rating (AAPR) [28] have used such approaches. Although there has been a lot of work dealing with AES, AAPR, and document classification problems, researchers have not attempted to investigate the effectiveness of a hierarchical attention model for the task of academic venue recommendation.

Therefore, we aim to design a unified architecture that contains a Bidirectional LSTM (Bi-LSTM) along with a Hierarchical Attention Network (HAN) to address this challenging task. The proposed architecture is called a modularized Hierarchical Attention-based Scholarly Venue Recommender system (HASVRec), which only requires the title, abstract, keywords, field of study, and the authors of a new paper along with their publication record to recommend the most suitable venues. Generally, bidirectional LSTMs have the advantage of remembering longer sequences than GRUs and outperform them in tasks that require modeling long-distance relations [29, 30]. As the abstract of a research paper contains longer sequences and we focus more on accuracy (relevant venues) than time, we adopt a Bi-LSTM in place of a GRU here due to its capability of learning the meaning of a word using both forward and backward context [31]. We have also extended our model by incorporating modularized attentive pooling techniques to get a high-level module representation integrating abstract, title, keywords, the field of study, and authors of a source paper along with their publication records.

Finally, the key contributions of this work are threefold, as described below.

- To deal with cold-start[4] issues posed by new researchers and venues, we integrate latent features from the abstract, title, keywords, and field of study of a source paper using a Bi-LSTM model, Hierarchical Attention, and Attentive Pooling. We employ a method involving a weighted average of one-hot encodings for the authors of a source paper and their publication records to enhance the quality of recommendations. Even if the researchers' past publication records are sparse, HASVRec works fine because it uses a modular structure to greatly focus on the work at hand.

- To address data sparsity[5] issue and to assist in identifying the relevance of research papers in order to maintain stability[6], we modularize the given architecture to capture relevance and extract low dimensional latent factors from high dimensional input. Bi-LSTMs can explicitly capture both short-term and long-term dependencies among words in an end to end process. A separate attention mechanism is applied to the word and sentence level to capture important semantics at every level of the hierarchy. Consideration of both structural information and contributions of various parts helps the model to achieve better accuracy and stability.

- We conduct comprehensive experiments using a real-world dataset, i.e., DBLP-Citation-Network V11, to evaluate the performance of the proposed HASVRec system. The system outperforms several other state-of-the-art venue recommendation models with substantial improvements in accuracy, $F1$, nDCG@k, MRR, average venue-quality (ave-quality), and stability.

The rest of the paper is organized as follows. We visit the related literature in Section 2. We provide preliminaries in Section 3 and more elaborate problem description in Section 4. The explanation of the different features we adopted in our proposed framework is provided in Section 5. Experimental details, including data description,

[4]Cold start problems mainly pertain to new researchers and new venues in academia.

[5]Sparsity denotes average distance amongst pairs of related papers.

[6]Stability denotes the change in ranked recommendations with the introduction of new papers.

evaluation metrics, parameter selection, and experimental results are illustrated in Section 6. The study of the proposed approach is reported in Section 7. We finally conclude in Section 8.

2. Related Work

Adomavicius et al. [1] mention mainly three types of recommender systems on the basis of working principles in their review paper on recommender systems. We include an additional type of recommendation approach called network-based recommendation [7]. They are discussed below.

2.1. Collaborative Filtering Based Recommendation (CF)

Yang et al. [32] proposed a model that determines whether publication venues are distinguishable by writing style or not. In another paper, Yang et al. [15] used a collaborative filtering model incorporating writing style and topic information of papers in order to recommend venues. Yang et al. [33] proposed another joint multi-relational model (JMRM) of venue recommendation for author-paper pairs. Hyunh et al. [16] proposed a collaborative knowledge model (CKM) that organizes collaborative relationships among researchers. Yu et al. [34] proposed a prediction model that uses collaborative filtering for a personalized academic recommendation. Liang et al. [17] proposed a new probabilistic approach that specifically consolidates user exposure to items into collaborative filtering. Alhoori et al. [11] presented a system that recommends scholarly venues taking into account the researcher's reading behavior.

2.2. Content Based Filtering Recommendation (CBF)

Medvet et al. [35] proposed a model that considers the title and abstract of papers to recommend scholarly venues. Errami et al. [36] proposed a model called eTBLAST to recommend journals based on abstract similarity. Schurmie et al. [19] proposed the Journal/Author Name Estimator (Jane) [7] that uses a biomedical database called MEDLINE in order to recommend journals based on abstract similarity. Similarly, Wang et al. [20] presented a content-based publication recommender system (PRS) in the computer science domain, exploiting softmax regression, and chi-square feature selection techniques. Recently, a few online services have started providing support for suggesting journals using keywords, title, and abstract matching. These services include the Elsevier Journal Finder [8] [18], Springer Journal Suggester [9], Edanz Journal Selector [10] and EndNote Manuscript Matcher [11], etc. Elsevier Journal Finder recommends journals from the Elsevier publishers only [18].

[7] http://jane.biosemantics.org
[8] http://journalfinder.elsevier.com
[9] http://journalfinder.com
[10] https://www.edanzediting.com/journal-selector
[11] http://endnote.com/product-details/manuscript-matcher

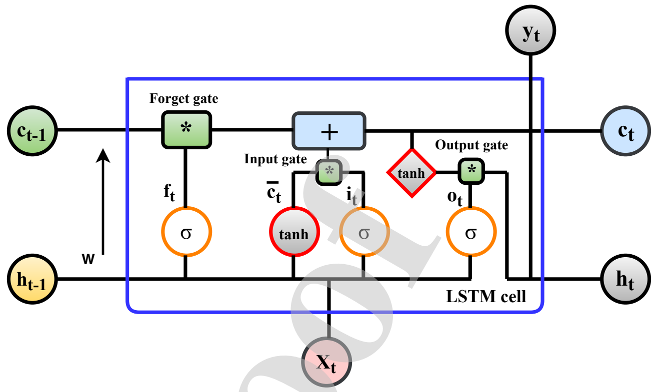


Figure 1: Single LSTM Cell

2.3. Network Based Recommendation (NB)

Klamma et al. [37] proposed a social network analysis (SNA) based method that recommends academic events to researchers. Silva et al. [21] proposed a novel three-dimensional research analytics framework (RAF) incorporating relevance, productivity, and connectivity parameters to provide an effective recommendation system. Pham et al. [38] used the number of papers of a researcher in a venue to determine her rating for that particular venue using social network analysis. Later, Pham et al. [39] presented clustering techniques on a social network of researchers to identify communities in order to generate venue recommendations. Chen et al. [22] introduced a model called AVER that recommends scholarly venues to a target researcher. Later, Yu et al. [7] extended AVER and presented a novel personalized academic venue recommendation model called PAVE. Luong et al. [12] introduced a methodology to identify suitable publication venues for researchers by investigating the co-authorship network. Later, Luong et al. [40] presented a social network-based approach that naturally finds suitable publication venues for authors by investigating their co-authorship networks in a similar field.

2.4. Hybrid Recommendation (HR)

Wang et al. [20] proposed a hybrid article recommendation approach incorporating CBF and CF approaches. Pradhan et al. [41] proposed a diversified yet integrated social network analysis and contextual similarity-based scholarly venue Recommender system (DISCOVER). Boukhris et al. [42] suggested a hybrid recommendation approach to provide venue recommendations based on the venues of the co-citers, co-affiliated researchers, the coauthors of the target researcher. Feng et al. [43] proposed a journal recommender system to suggest suitable PubMed journals for biomedical literatures based on paper's abstract. Zhang et al. [44] proposed a cross-domain recommender system with consistent information transfer. He at al. [45] presented a novel neural collaborative filtering based recommendation method. Rendle [46] proposed a unified framework for bayesian personalized ranking exploiting implicit feedback. Minkov et al. [47] introduced a method of recommending future events. Yang et al. [48] presents a long-short-term memory (LSTM) based model for context-aware citation recommendation. Model first learns the distributed representations of the citation contexts and the scientific papers seperately and then measure the relevance based upon the learned features. Xia et al. [49] proposed a socially aware recommendation system for conferences. Pradhan et al. [50] proposed a content and network-based academic venue recommender system (CNAVER).

3. Preliminaries

In this segment, we talk about various notations and terminology.

3.1. Architecture of LSTM Model

An LSTM model consists of various components that capture temporal information from sequential inputs. It also reduces high dimensional input data into lower-dimensional output data via an embedding layer, a feature extraction layer, and an output layer. Training conventional Recurrent Neural Networks with gradient descent based backpropagation is difficult due to problems like vanishing gradients and exploding gradients. Long Short Term Memory (LSTM) networks have been designed to address the above problems.

LSTMs contain units called memory blocks in the recurrent hidden layer [51]. These memory blocks store the temporal state of the network using multiplicative units called gates to control the flow of information, as displayed in Fig. 1. There are three gate controllers, namely, the forget gate, the input gate, and the output gate [52]. The structure and functioning of these gates are described below.

(i) **Forget gate**: It is denoted by f_t. It controls parts of the information from the long term state that should be erased and information that should be passed to next long term state c_t.

(ii) **Input gate**: It is denoted by i_t. It controls parts of g_t i.e. the input and the previous short term state h_{t-1} that should be passed to next long term state i.e. c_t. Hence, the input of the current state is partially passed in to long term memory.

(iii) **Output gate**: It is denoted by o_t. It controls parts part of the long term state c_{t-1} that should be read and output at this time step (both to h_t and y_t).

More formally, each cell in an LSTM can be computed as follows:

$$\mathbf{i}_{(t)} = \sigma(W_{xi}^T \cdot \boldsymbol{X}_{(t)} + \boldsymbol{W}_{hi}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \quad (1)$$

$$\boldsymbol{f}_{(t)} = \sigma(W_{xf}^T \cdot \mathbf{X}_{(t)} + \boldsymbol{W}_{hf}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \quad (2)$$

$$\boldsymbol{o}_{(t)} = \sigma(\boldsymbol{W}_{xo}^T \cdot \boldsymbol{X}_{(t)} + \boldsymbol{W}_{ho}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \quad (3)$$

Table 1: Symbols used in the paper

Symbols	Descriptions
p_i	Given paper
r_i^j	Components of a given paper p_i (modules)
v_i	Venue of a given paper
p_0	Seed paper
V	Set of available venues
s_i	Sentence-level representation of i-th sentence
M_i	High-level module representation of i-th module
L	Maximum no. of sentences in abstract
A	Maximum length of the author sequence
D	Maximum no. of words in a title of research article
B	Maximum no. of keywords of a research article
T_i	Maximum no. of words in a given sentence
w_{it}	No. of words in the i-th sentence
$\overrightarrow{LSTM}$	Forward LSTM
$\overleftarrow{LSTM}$	Backward LSTM
W_e	Embedding matrix
x_{ij}	Word embedding vector of a given word w_{ij}
h_{it}	High-level feature vector of x_{it}
u_{it}	Hidden representation of h_{it}
W_w	Weight matrix of the MLP
b_w	Bias vector of the MLP
u_w	Word-level context vector
α_{it}	Attention weight of feature vector h_{it}
h_i	High-level representation of sentence s_i
u_i	Hidden representation of h_i
W_s	W_s is the weight matrix of the MLP
b_s	Bias vector of the MLP
u_s	Sentence-level context vector
α_i	Attention weight of feature vector h_i
γ_i	Weight parameter of author a_i
F_i^v	Freshness distribution of venue v_i
N_i^v	Frequency distribution of venue v_i
W_i^v	Normalized weighted score of venue v_i
β_i	Weight parameter of venue v_i
M	Paper-level representation of P_i
z_i	Hidden representation of M_i
W_m	Weight matrix of the MLP (attentive pooling)
b_m	Bias vector of the MLP
α_{mi}	Attention weights of feature vector M_i
u_{mw}	Module-level context vector
S_j	Softmax probabilities for each label j
$L(\theta)$	Categorical cross-entropy loss
Y_j	Gold-standard output for each label j

$$\mathbf{g}_{(t)} = \tanh(\boldsymbol{W}_{xg}^T \cdot \boldsymbol{X}_{(t)} + \boldsymbol{W}_{hg}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \quad (4)$$

$$\boldsymbol{c}_{(t)} = \boldsymbol{f}_{(t)} \otimes \boldsymbol{c}_{(t-1)} + \boldsymbol{i}_{(t)} \otimes \boldsymbol{g}_{(t)} \quad (5)$$

$$\boldsymbol{y}_{(t)} = \boldsymbol{h}_{(t)} = \boldsymbol{o}_{(t)} \otimes \tanh\left(\boldsymbol{c}_{(t)}\right) \quad (6)$$

Where, W_{xi}, W_{xf}, W_{xo}, W_{xg} are weight matrices for each of the three gates and the intermediate representation for their connections to the input vector X_t. W_{hi}, W_{hf}, W_{ho}, W_{hg} are weight matrices for each of the three gates and the intermediate representation for their connections to the previous short term state h_{t-1}. b_i, b_f, b_o, b_g are the bias terms for each of the three gates and the intermediate representation. This way, an LSTM can recognize important inputs and store them in the long term state and learn to extract the same, wherever necessary.

4. Problem Formulation

Definition 1. ***Venue Recommendation***. *Let us consider a set of unique papers $\{p_1, p_2, ..., p_n\}$ where p_i is published in a particular venue $v_j \in V = \{v_1, v_2, ..., v_m\}, (m \leq n)$, V being a predefined set of publication venues. Note that, more than one paper can be published at a single venue, and thus $m \leq n$.*

Given an input paper (seed paper) p_0, the venue recommendation task is to recommend an ordered list of suitable publication venues (v_{0_1}, v_{0_2} ,$\cdots$, v_{0_k}) such that v_{0_1} is the most relevant and v_{0_k} is k-th most relevant venue in the decreasing order of relevance or suitability where each $v_{0_i} \in V$.

The above can be seen as a multi-class classification problem. Initially, we need to uniquely identify available publication venues and set them as individual classes. So, we have, $C = \{c_1, c_2, ..., c_m\}$ ($m \leq n$) be the number of possible labels. So a softmax classifier can be used to estimate the probability of individual classes $P(\mathsf{c_i}|\mathsf{p_0})$ ($i = \{1, 2, ..., m\}$) such that the sum of probabilities of all classes to be 1.

For a given seed paper p_0, the venue recommendation task can be modelled as recommending an ordered list of classes (c_{0_1}, c_{0_2} ,$\cdots$, c_{0_k}), such that c_{0_1} is the most relevant and c_{0_k} is k-th most relevant class (or venue) in the decreasing order of estimated probabilities $P(\mathsf{c_i}|\mathsf{p_0})$.

Venue recommendations are provided if the title, keywords, abstract, field of study, and author of a seed paper are given to the system as input (Fig. 2).

The complete list of symbols used in this paper are listed in Table 1.

5. Proposed Framework

In this section, we provide an overview of HASVRec and further explain its functional architecture in detail.

5.1. Overview

We present a modularized Hierarchical Attention-based Scholarly Venue Recommender system (HASVRec), as depicted in Fig. 2. We present a step-insightful layered architecture where each layer realizes a specialized task. The four essential layers are portrayed as given underneath:

(i) **Embedding Layer (Layer-1)**: This layer aims to structure the dataset via the extraction of relevant features for further processing. It is also called the Word Representation Layer as it is mainly introduced to get low dimensional vector representations from the high dimensional sparse inputs. Word embeddings contain a set of feature selection methods that transform each input into a fixed-size dense vector representations. They can capture context of the word and can provide information about relation of a word with other words. Hence, meaning of a word can be predicted accurately as it can capture syntactic and semantic information about the words.

(ii) **Module-level Representation Layer (Layer-2)**: This layer consists of two sub-parts: Attention-based Bi-LSTM (ABLSTM), and weighted average method based one hot encoding. The objective of the Bi-LSTM is to transform each input into a fixed-size high dimensional vector. Since different words and sentences in a document are differentially informative, a hierarchical attention mechanism is applied on top of the Bi-LSTM. This layer includes two levels of attention mechanisms–one at the word level and one at the sentence level. We employ ABLSTM to get module-level representation for abstract, title, keywords, and field of study of the source paper. However, weighted average method is applied to obtain the module-level representation of author and venue of the source paper.

(iii) **Modularized Attentive Pooling Layer (Layer-3)**: This layer is used to aggregate the information from each module-level representation to form a paper-level representation. Attentive Pooling (AP) is adopted in this layer to achieve such aggregation.

(iv) **Dense Layer (Layer-4)**: The Fully connected layer (FC) connects every neuron in this layer to all the activations of paper-level representation (M). The objective of this layer is to map the paper-level representation to the desired output vector (top-K venues) by incorporating hidden layers with dropouts in a densely connected network.

5.2. Architecture of HASVRec

HASVRec comprises of majorly four essential layers, namely Embedding Layer, Module-level Representation Layer, Modularized Attentive Pooling Layer, and Dense Layer. The functional architecture of HASVRec is depicted in Fig. 3.

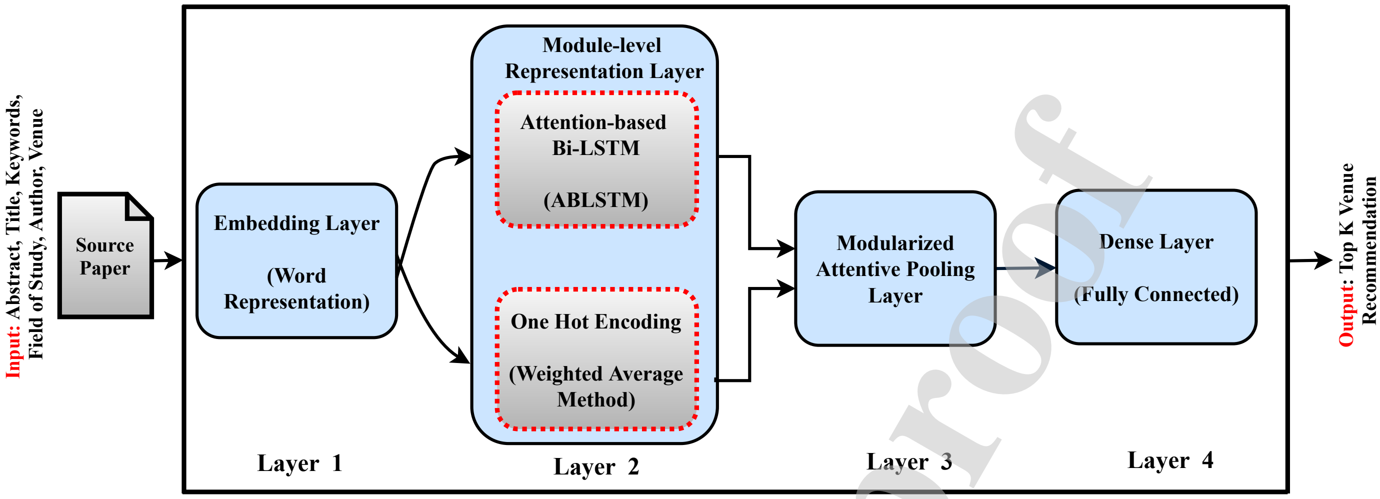


Figure 2: Block diagram of HASVRec

5.2.1. *Embedding Layer (Layer-1)*

This layer consists of two types of operations, namely, data preprocessing and embedding operations. We use the 'abstract', 'title', 'keywords', 'field of study', 'authors', 'affiliation', and 'venue' of a particular paper for our experiments. The titles, abstracts, keywords and field of study are preprocessed by converting them to lower case, removing stopwords, punctuation, and lemmatizing the verbs in the text. We initialize W_e using pre-trained 300 dimensional GLoVE Embeddings. The output is processed by encoding the publication venue names into One-hot Vectors to feed into the algorithm.

The input representation is constructed as follows:

(i) Let us assume that the abstract of a research article has L sentences s_i, and each sentence contains T_i words. w_{it} with t ∈ [1, T] represents the words in the i-th sentence.

(ii) Similarly, the title of a research article contains D number of words c_i with i ∈ [1, D].

(iii) Let us assume the keywords of a research article has B number of words.

(iv) In this work, we have considered the size of the dimension D and B as 10 and 6, respectively.

(v) We have considered maximum number of field of study and authors of research article to be as 6 in this embedding layer.

(vi) In all the cases, the input is zero-padded wherever necessary. Word vectors are initialized by zeros if they are not in the pre-trained vocabulary.

5.2.2. *Module-level Representation Layer (Layer-2)*

A research article has a hierarchical structure going from word-level (words form sentences) to sentence-level (sentences form a document). Different words and sentences in a document are differentially informative. Moreover, the importance of words and sentences is highly context-dependent, i.e., the same word or sentence may have varying importance in different contexts. To include sensitivity to this fact, our model includes two levels of attention mechanisms–one at the word level and one at the sentence level. This hierarchical structure pays more or less attention to individual words and sentences while constructing the representation of the research article.

Generally, a research paper p_0 is divided into various modules such as abstract, title, keywords and so on. Let r_0^1, r_0^2, r_0^3, r_0^4, r_0^5 and r_0^6 denote abstract, title, keywords, field of study, authors and venues of the source paper p_0. The module r_0^1 consists of a few sentences, and each sentence is composed of a few words. Similarly, r_0^2, r_0^3, and r_0^4 are composed of few words. Thus, there is a hierarchical structure consisting of modules, sentences, and words at various levels. The hierarchical structure of the source paper p_0 is shown in Fig 3.

Therefore, our Hierarchical Attention-based Bi-LSTM consists of two attention components: (i) Word-level attention, and (ii) Sentence-level attention (Fig. 4). At each level, these components are used in conjunction with a Bi-directional LSTM (Bi-LSTM). The LSTM is used to extract higher-level phrase representations from the word embedding vectors considering both the preceding and succeeding context representation. There are two benefits of employing attention mechanisms into the model. Firstly, most of the time, it results in better performance. Secondly, it also gives an insight into which words and sen-

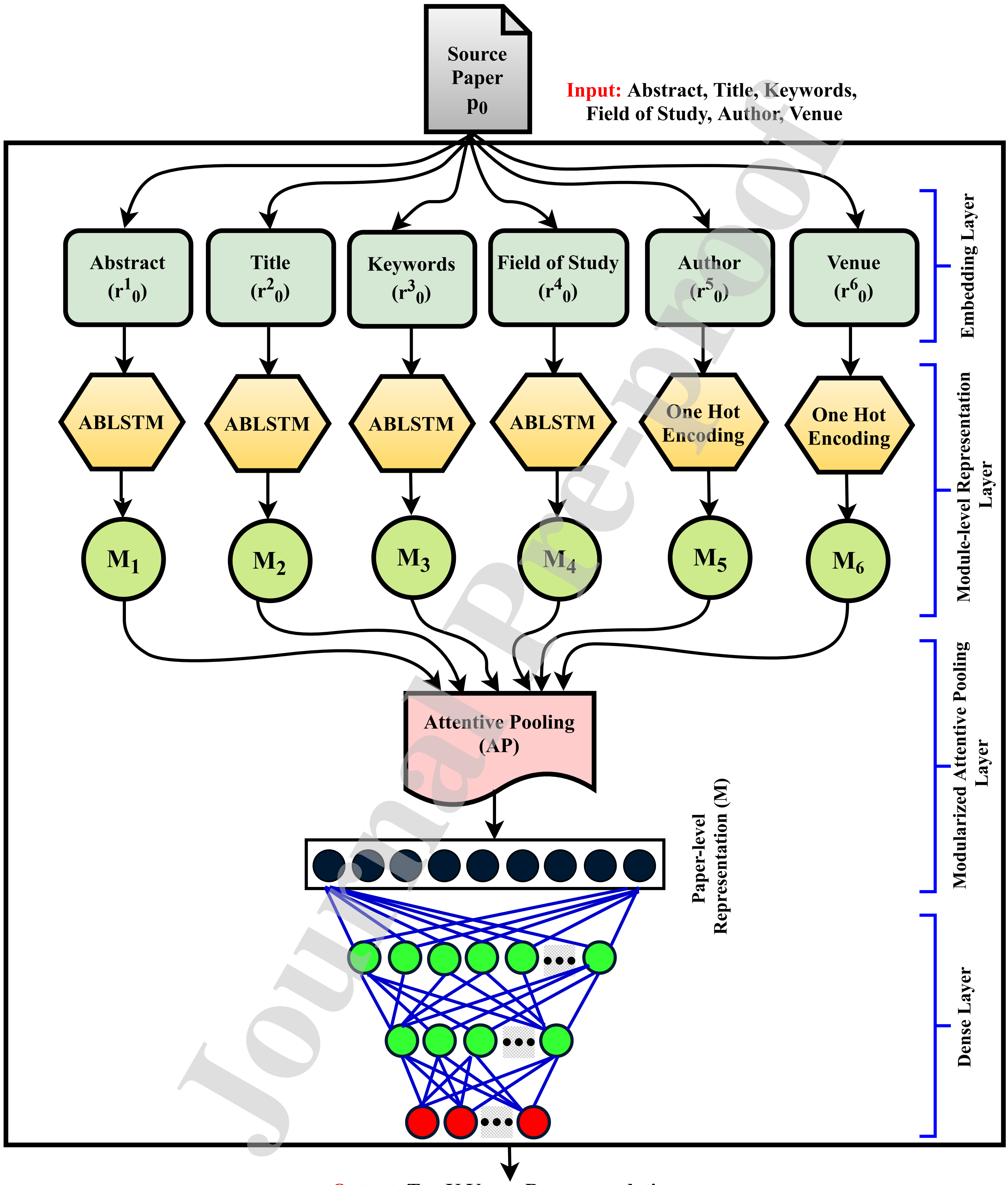


Figure 3: Functional architecture of HASVRec

tences contribute to the classification decision, which can be of value in applications and analysis.

5.2.3. *Word-level Attention*

Given a sentence with words w_{it}, t ∈ [1, T], we first embed the words into vectors through an embedding matrix W_e, $x_{ij} = W_e\, w_{ij}$. We then employ a Bi-directional LSTM (Bi-LSTM) to get representations of the input words by summarizing information from both directions. We do this to incorporate contextual information in the representation. The Bi-directional LSTM contains the forward LSTM (represented by $\overrightarrow{LSTM}$) which reads the sentence s_i from w_{i1} to w_{iT} and the backward LSTM (represented by $\overleftarrow{LSTM}$) which reads the feature sequences from w_{iT} to w_{i1}.

$$x_{it} = W_e w_{it}, t \in [1, T], \tag{7}$$

$$\overrightarrow{h_{it}} = \overrightarrow{LSTM}(x_{it}), t \in [1, T] \tag{8}$$

$$\overleftarrow{h_{it}} = \overleftarrow{LSTM}(x_{it}), t \in [T, 1] \tag{9}$$

$$h_{it} = \overrightarrow{h_{it}} || \overleftarrow{h_i t} \tag{10}$$

Thus, we obtain a representation for a given word w_{it} by concatenating the forward hidden state $\overrightarrow{h_{it}}$ and backward hidden state $\overleftarrow{h_{it}}$ giving h_{it}, which summarizes the information of the whole sentence centered around w_{it}.

Not all words contribute equally to the representation of the sentence meaning. Hence, we introduce an attention mechanism to extract words that are important to the meaning of the sentence and aggregate the representation of those words to form an overall sentence representation.

Given a sequence of vectors $h_{i1}, h_{i2}, \cdots h_{iT}$, we apply an attention layer to obtain the comprehensive sentence-level context representation s_i of the whole word sequence as defined in Eqn. 11, Eqn. 12, and Eqn. 13. The attention mechanism is formally defined as follows:

$$u_{it} = tanh(W_w * h_{it} + b_w) \tag{11}$$

$$\alpha_{it} = \frac{exp(u_{it}^T u_w)}{\sum_t exp(u_{it}^T u_w)} \tag{12}$$

$$s_i = \sum_t \alpha_{it} * h_{it} \tag{13}$$

Where h_{it} represents the high-level feature vector or sequential output vector from the Bi-LSTM for the i-th input vector x_{it}. u_{it} represents the hidden representation of h_{it}, obtained by feeding h_{it} into a one-layer multi-layer perceptron (MLP). W_w and b_w represent the weight matrix and bias vector of the MLP, respectively. u_w represents the word-level context vector, which is a randomly initialized vector, and can be learned at the training stage. α_{it} represents the attention weight for the high-level feature vector h_{it}.

We measure the importance of a word through the similarity between u_{it} and the word-level context vector u_w. After that, we obtain a normalized importance weight α_{it} using the softmax function. Finally, the sentence-level representation s_i is constructed as the sum of all the high-level feature vectors $h_{i1}, h_{i2}, \cdots h_{iT}$, weighted by the parameter α_{it}.

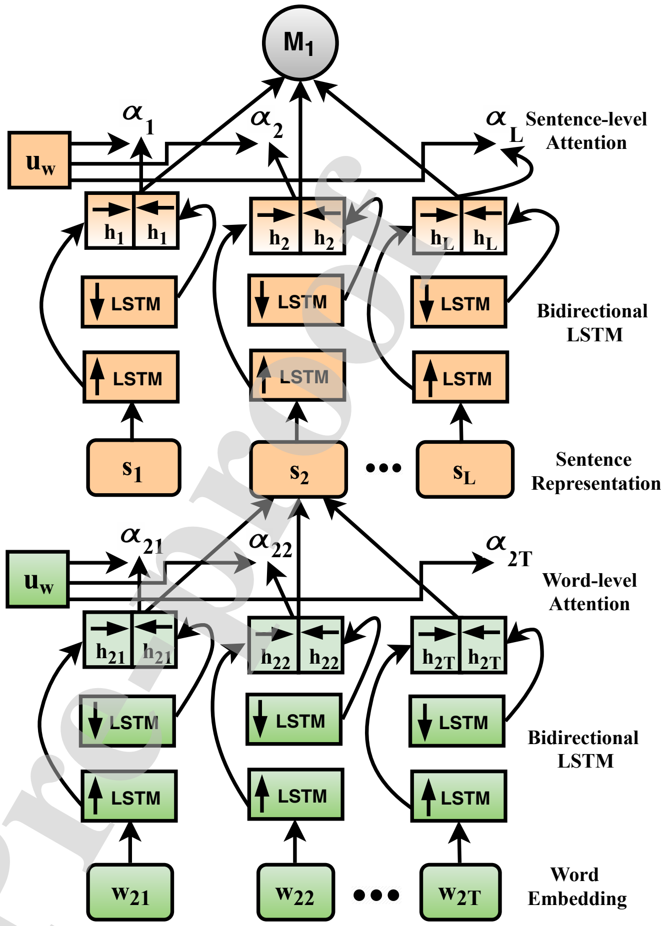


Figure 4: Hierarchical Attention Network (HAN)

5.2.4. *Sentence-level Attention*

Given a sentence vector s_i, we can get a module vector in a similar fashion as mentioned in Section 5.2.3. We employ a Bi-directional LSTM (Bi-LSTM) to get representations of the input sentences by summarizing information from both directions. We do this to incorporate contextual information in the representation. The Bi-directional LSTM contains the forward LSTM (represented by $\overrightarrow{LSTM}$) which reads the module from sentence vector s_1 to s_L and the backward LSTM (represented by $\overleftarrow{LSTM}$) which reads the feature sequences from s_L to s_1.

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(s_i), i \in [1, L] \tag{14}$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(s_i), i \in [L, 1] \tag{15}$$

$$h_i = \overrightarrow{h_i} || \overleftarrow{h_i} \tag{16}$$

Thus, we obtain a representation for a given sentence s_i, by concatenating the forward hidden state $\overrightarrow{h_i}$ and backward hidden state $\overleftarrow{h_i}$, giving h_i, which summarizes the information of the whole module centered around sentence s_i.

Not all sentences contribute equally to the representation of the research article. To reward sentences that are better clues to classify the related journal of a research article correctly, we introduce an attention mechanism to extract sentences that are important to the meaning of the overall research article. We then aggregate the representation of those sentences to form a module vector (vector of an overall research module such as the abstract).

Given a sequence of vectors $h_1, h_2, \cdots h_L$, we apply an attention layer to obtain the comprehensive module-level context representation M_1 of the whole sentence sequence as defined in Eqn. 17, Eqn. 18, and Eqn. 19. The attention mechanism is formally defined as follows:

$$u_i = tanh(W_s * h_i + b_s) \tag{17}$$

$$\alpha_i = \frac{exp(u_i^T u_s)}{\sum_i exp(u_i^T u_s)} \tag{18}$$

$$M_1 = \sum_i \alpha_i * h_i \tag{19}$$

Where h_i represents the high-level feature vector or sequential output vector from the Bi-LSTM for the i-th input vector s_i. u_i represents the hidden representation of h_i, obtained by feeding h_i into a one-layer multi-layer perceptron (MLP). W_s and b_s represent the weight matrix and bias vector of the MLP, respectively. u_s represents the sentence-level context vector, which is a randomly initialized vector, and can be learned at the training stage. α_i represents the attention weight for the high-level feature vector h_i.

We measure the importance of a sentence through the similarity between u_i and the sentence-level context vector u_s. After that, we obtain a normalized importance weight α_i using the softmax function. Finally, the module-level representation M_1 is constructed as the sum of all the high-level feature vectors $h_1, h_2, \cdots h_L$, weighted by the parameter α_i.

There is only one sentence in the title of the research article, so it is reasonable to get the module-level representation M_2 for the title using an attention mechanism only at the word-level. Similarly, for keywords, and the field of study, attention mechanism only at the word-level is employed to get a module-level representation M_3 and M_4, respectively.

5.2.5. Weighted average method based representation

Generally, the authors are independent of each other and hence, we adopt a weighted average method to obtain the module-level representation M_5 of authors. The weighted average method is defined in Eqn. 20.

$$M_5 = \sum_i^A \gamma_i * a_i \tag{20}$$

where $\gamma = (\gamma_1, \cdots, \gamma_A)^T$ is the weight parameter. a_i is the embedding vector of the i-th author in the source paper, which is randomly initialized and can be learned at the training stage. A is the maximum length of the author sequence.

In general, the researchers intend to publish in academic venues that acknowledge high-quality papers and participate in academic conferences or workshops relevant to their area of research. When a researcher publishes at the same venue frequently, it implies that she works in the same/similar fields of study. Hence, there is a high chance that the researcher may choose the same or similar venues in near future publications. It is because her area of research is not changing in a given period of time.

Therefore, the past venues information is likely to be helpful in making more accurate predictions. Besides, the model can be improved by considering both the frequency (number of papers published in journals) and the freshness (recent articles published in journals) of the venues. Papers published in recent years can describe the current scope of a venue more accurately. Researchers usually desire to contact venues that are currently publishing articles similar to their area of research.

We use inverse log-weighting to give more weight to the research papers published at venues in the current year, and the weight reduces in a logarithmic fashion over the years going backward. Initially, for a given source paper, we need to identify the past publications venues of all its authors. All the identified venues are then ranked based on a venue-specific weighted score computed using a venue's frequency and freshness.

Example: Table 2 shows the publication venue distributions for four co-authors a_{01}, a_{02}, a_{03}, and a_{04} of a source paper p_0. Suppose, all the co-authors previously published their papers across four different venues v_1, v_2, v_3, and v_4. For example, author a_{03} published her papers only at venue v_4 in the years 2018, and 2019.

For each venue v_i, we get a weighted freshness score (F_i^v) as depicted in Eqn. 21.

$$\boldsymbol{F}_i^v = \sum_{y_i \in Y} \frac{1}{log_2(y_o - y_i + 2)}, \text{ where} \tag{21}$$

$$Y = \{1800, \cdots, 2020\}, \ y_o \text{ is the latest year in } Y, \tag{22}$$

$$\text{and } y_i \text{ is the publication year of venue } v_i \tag{23}$$

Thus, using Eqn. 21, the freshness distribution (F_1^v) of venue v_1 is shown in Eqn. 24. The publications associated with venue v_1 were made in the years 2018, and 2020.

$$F_1^v = \frac{1}{log_2(2020-2018+2)} + \frac{1}{log_2(2020-2020+2)} + \frac{1}{log_2(2020-2020+2)} \tag{24}$$

$$F_1^v = \frac{1}{log_2(4)} + \frac{1}{log_2(2)} + \frac{1}{log_2(2)} \tag{25}$$

$$F_1^v = \frac{1}{2} + \frac{1}{1} + \frac{1}{1} = 2.5 \tag{26}$$

Table 2: Author-Year publication details

Authors	v_1	v_2	v_3	v_4
a_{01}	2018, 2020	2018	2015, 2018, 2020	
a_{02}		2019	2015	2014
a_{03}				2018, 2019
a_{04}	2020	2014	2016	2014

Table 3: Normalized weighted score based venues ranking

Venues	F_i^v	N_i^v	W_i^v	*Ranks*
v_1	2.5	0.2	0.5	2
v_2	1.464	0.2	0.292	4
v_3	2.598	0.333	0.865	1
v_4	1.797	0.266	0.478	3

Similarly, we get the freshness distributions $F_2^v = 1.464$, $F_3^v = 2.598$, and $F_4^v = 1.797$. The frequency distribution of each venue is calculated by using Eqn. 27.

$$\boldsymbol{N}_i^v = \frac{\#\text{ papers published at venue } v_i}{\sum_{i=1}^{m} \#\text{ papers published at venue } v_i} \tag{27}$$

where m is the number of unique venues identified for a set of co-authors. In the example above, since the total number of publications are fifteen (including all venues), and there are three papers published at venue v_1, the frequency distribution (N_1^v) of venue v_1 is shown in Eqn. 28.

$$\boldsymbol{N}_1^v = \frac{3}{15} = 0.2 \tag{28}$$

Similarly, we get the frequency distributions of $N_2^v = 0.2$, $N_3^v = 0.333$, and $N_4^v = 0.266$. After getting both the freshness F_i^v and frequency N_i^v distributions, the normalized weighted score (W_i^v) is calculated as defined in Eqn. 29.

$$\boldsymbol{W}_i^v = F_i^v * N_i^v \tag{29}$$

Thus, the normalized weighted score (W_1^v) of venue v_1 is shown in Eqn. 30.

$$\boldsymbol{W}_1^v = F_1^v * N_1^v \tag{30}$$

$$\boldsymbol{W}_1^v = 2.5 * 0.2 = 0.5 \tag{31}$$

Table 3 shows the freshness distribution (F_i^v), frequency distribution (N_i^v), and normalized weighted score (W_i^v) for all venues. Finally, the venues are sorted in the descending order of normalized weighted score (W_i^v). As we can see, both venues v_1 and v_2 have three publications, but the normalized weighted score W_1^v of v_1 is higher than that of v_2. More recent publications at v_1 lead to a higher freshness score for venue v_1 than v_2. Similarly, although there are four publications at venue v_4, the rank of venue v_4 is still lower than the rank of venue v_1. Moreover, the publications associated with both venues v_2 and v_4 were made in the same set of years, that is, 2014, 2018, and 2019. However, the rank of venue v_4 is higher than the rank of venue v_2 due to its higher frequency. Thus, we can conclude that both freshness and frequency are equally important to identify top ranking venues.

We consider only the top five venues based on W_i^v and adopt a weighted average method to obtain the module-level representation M_6 for the venues/past publication records. The weighted average method is defined in Eqn. 32.

$$M_6 = \sum_{i}^{n} \beta_i * v_i \tag{32}$$

where $\beta = (\beta_1, \cdots, \beta_n)^T$ is the weight parameter. v_i is the embedding vector of the i-th venue in the ranking order of source paper, which is randomly initialized and can be learned at the training stage. n is the maximum number of venues sequence (n $\leq$ m, n=5 here).

5.2.6. *Modularized Attentive Pooling Layer (Layer-3)*

An attentive pooling layer is used to aggregate the input representation of the sequences by measuring the contribution of each input vector to form a hybrid representation. After obtaining the module-level representations M_1 (abstract), M_2 (title), M_3 (keywords), M_4 (field of study), M_5 (author), and M_6 (venues), we aggregate them using attentive pooling to learn the final paper-level representation M. Attentive pooling helps us to compute the weights of modules' contribution to final paper-level representation M. Formally, we have

$$z_i = tanh(W_m M_i + b_m) \tag{33}$$

$$\alpha_{mi} = \frac{z_i^T u_{mw}}{\sum_k exp(z_k^T u_{mw})} \tag{34}$$

$$M = \sum_i \alpha_{mi} * z_i \tag{35}$$

Where M_i represents the high-level module representations. z_i represents the hidden representation of M_i, obtained by feeding M_i into a one-layer multi-layer perceptron (MLP). W_m and b_m represent the weight matrix and bias vector of the MLP, respectively. u_{mw} represents the module-level context vector, which is a randomly initialized vector and can be learned at the training stage. α_{mi} represents the attention weight for the high-level module vector M_i.

We measure the importance of a module through the similarity between z_i and the module-level context vector u_{mw}. After that, we obtain a normalized importance weight α_{mi} using the softmax function. Finally, the paper-level representation M is constructed as the sum of all the module-level feature vectors M_1, M_2, M_3, M_4, M_5, and M_6 weighted by the parameter α_{mi}.

Algorithm 1: Algorithm of HASVRec

Input: Abstract(r_0^1), Title (r_0^2), Keywords (r_0^3), Field of Study (r_0^4), Author (r_0^5), and Venues (r_0^6) of seed paper p_o

Output: Top K venue recommendation

```
1  Let us assume the abstract(r_0^1) of a research article p_o has L sentences s_i and each sentence contains T_i words.
2  Consider w_it with t ∈ [1, T] represents the words in the i-th sentence
3  Construct word embedding table for r_0^1, r_0^2, r_0^3, r_0^4 using pretrained word-vector with Eqn. 7
4  Use separate vocabulary to construct one hot encoding vector for authors and venues
5  foreach words w_it ∈ s_i do
6      Employ Bi-LSTM to obtain preceding contextual features →h_it and sucedding contextual features ←h_it from feature sequences, using Eqn. 8 and 9
7      Combine →h_it and ←h_it to obtain comprehensive contextual representation h_it, using Eqn. 10
8      Employ word-level attention layer to obtain high-level context representation s_i, using Eqn. 13
9  foreach sentences s_i ∈ r_0^1 do
10     Employ Bi-LSTM to obtain the preceding contextual features →h_i and sucedding contextual features ←h_i from the feature sequences, using Eqn. 14 and 15
11     Combine →h_i and ←h_i to obtain comprehensive contextual representation h_i, using Eqn. 16
12     Employ sentence-level attention to obtain high-level context representation M_1, using Eqn. 19
13 Repeat the steps 5 to 8 for r_0^2, r_0^3 and r_0^4
14 Apply weighted average method based one-hot encoding scheme for authors r_0^5 to obtain module-level representation M_5 using Eqn. 20
15 Apply weighted average method based one-hot encoding scheme for venues r_0^6 to obtain module-level representation M_6 using Eqn. 32
16 foreach modules M_i ∈ p_o do
17     Employ attentive pooling layer to obtain high-level context representation M, using Eqn. 35
18 Feed the comprehensive context representations into the soft-max classifier to get class labels using Eqn. 36;
19 Update parameters of the model using the loss function Eqn. 37 with the Adam method
20 Return top K venue recommendation
```

5.2.7. *Dense Layer (Layer-4)*

After obtaining the paper-level representation M, a dense fully-connected network is applied to map the input vector to the desired output vector. The network has the paper-level representation M as the input, two hidden layers, and an output layer. Rectified Linear Units (ReLU) are applied as an activation function to the hidden layers. To avoid potential overfitting, we apply dropout in between the layers of the network. We try various possibilities of dropout rates to obtain the best configuration. The output from the last hidden layer is then fed into the output layer, with softmax as the activation function. Let L is the number of possible labels in the training sample and z_j for each label j $\in$ [1, 2, $\cdots$, L] be the estimated unnormalized probabilities, then the softmax $S_j \in [0, 1]$ is defined as:

$$S_j = \frac{e^{z_j}}{\sum_{k=1}^{L} e^{z_k}} \tag{36}$$

We use the categorical cross-entropy loss to minimize the prediction error between the predicted venues and the actual venues:

$$L(\theta) = \sum_{j=1}^{L} Y_j log S_j \tag{37}$$

Where Y is the actual output. We adopt a one-hot encoding of size L for the output, where all elements except one are 0, and one element is 1. This element marks the correct class for the data being classified. We use the Adam Optimizer with mini-batching to learn the model parameter θ.

The entire algorithm comprising of all the layers along with inputs and outpits is mentioned in Algorithm 1.

6. Experiments

First, we outline the dataset and the evaluation metrics that are used for the assessment of the proposed system. Then, we explain the baseline methods, the experimental setting, and parameter tuning in further sub-sections. All experiments were conducted on a 64-bit, 2.4GHz Intel Core i5, 32-GB memory, and an 8-GB Titan XP GPU system. We implement our model based on the Pytorch framework and use a TITAN XP graphic card for learning.

6.1. *Evaluation Dataset*

We adopt the widely used DBLP-citation-network V11 [12] dataset to demonstrate the effectiveness of our proposed method. In this dataset, the citation data is extracted from several sources, such as DBLP, ACM, and MAG (Microsoft academic graph) [53]. The eleventh version originally contains 4, 107, 340 papers and after removing venues having less than 5 papers, we obtain 12, 669 publication venues and 3, 539, 049 papers. The venue-wise

[12] https://www.aminer.cn/citation

Table 4: Venue-wise paper statistics of the overall dataset

Range of papers	Number of papers	Number of journals
$5 \leq X \leq 100$	212,837	9,104
$100 \leq X \leq 400$	386,820	1,816
$400 \leq X \leq 2000$	1,195,285	1,378
$2000 \leq X \leq 10{,}000$	1,294,024	341
$X \geq 10{,}000$	450,083	30
Max. class size	30,879	1
Min. class size	5	42
Avg. class size	270	
All	3,539,049	12,669

paper statistics of the overall dataset are depicted in Table 4.

After removing duplicate papers, papers with missing fields, inconsistent entries in the database, journals having less than 50 number of papers, etc., we are left with $3,404,565$ papers and $4,670$ venues. We also ignore non-textual content from the abstracts of the papers. We assign 81% of preprocessed dataset as the training set, 9% as the validation set, and the rest 10% is considered as the test set.

Due to the vast amount of data, the number of labels (unique journals) also increases. The experiment is performed in two stages to demonstrate the efficacy of the proposed model and resolve the difficulty of training the model.

(i) Preparation of dataset for offline evaluation

(ii) Preparation of dataset for online evaluation

6.1.1. Preparation of the Dataset for Offline Evaluation

For offline evaluation, we identify only those venues having 50 to 400 papers and remove venues having more than 400 papers. The complete statistics of the overall dataset are depicted in Table 5. The split into training, validation, and testing sets are made, keeping in mind the cold-start issues for new venues and new researchers.

Table 5: Statistics of offline dataset

Types	Training Dataset	Validation Dataset	Testing Dataset
No. of papers	470,939	52,326	58,140
No. of venues	2,921	2,921	2,921

We consider four categories of venues and four categories of researchers based on venue count (v_c) (number of papers published at a given venue) and publication count (p_c) (the number of publications of a researcher) [7, 22]. The categories are defined below.

(i) Category 1 : $50 \leq v_c < 100$

(ii) Category 2 : $100 \leq v_c < 200$

(iii) Category 3 : $200 \leq v_c < 400$

(iv) Category 4 : $400 \leq v_c$

(v) Category 5 : $5 \leq p_c < 20$

(vi) Category 6 : $20 \leq p_c < 50$

(vii) Category 7 : $50 \leq p_c < 100$

(viii) Category 8 : $100 \leq p_c$

We ensure that each category is well represented in each of the training, validation, and testing set.

6.1.2. Preparation of Dataset for Onine Evaluation

For online evaluation, we identify only those venues having more than 400 papers and remove venues having less than 400 papers. The complete statistics of the overall dataset are depicted in Table 6.

Table 6: Statistics of online dataset

Types	Training Dataset	Validation Dataset	Testing Dataset
No. of papers	2,380,907	264,546	293,939
No. of venues	1,749	1,749	1,749

Due to operational constraints (difficulty to incorporate user study for all testing papers), only 20 sub-domains of Computer Science were selected as the gold-standard in our experiments. A total of 160 seed papers (8 from each sub-domains, 1 from each category) are chosen manually from the 20 sub-domains, namely, information retrieval (IR), image processing (IP), security (SC), wireless sensor network (WSN), machine learning (ML), software engineering (SE), computer vision (CV), artificial intelligence (AI), data mining (DM), theory of computation (TC), databases (DB), human-computer interaction (HCI), algorithms and theory (AT), natural language processing (NLP), parallel and distributed systems (PDS), worldwide web (WWW), web semantics (WS), computer architecture (CO), compiler design (CD) and multimedia (MM).

6.2. Evaluation Strategy

We adopt the following two kinds of evaluations to measure the performance of HASVRec against other state-of-the-art methods.

(a) **Coarse-level or offline evaluation**: As the name suggests, it provides some raw-level quick notion of how the proposed HASVRec system fares vis-a-vis other systems. We focus on the prediction accuracy to see whether the original publication venue for the test paper is predicted or not, and if yes, at what rank within some top N recommendations. Accuracy, MRR, and $F1$ evaluation metrics are used during the evaluation. We call this scenario *offline* because we can evaluate a system this way only when we have annotated test data.

(b) **Fine-level or online evaluation**: This evaluation-scenario is more realistic (and, that's why we call it *online*) as a researcher needs to have multiple as well as holistic venue recommendations from a system for her paper-in-writing. Here, we go a little deeper and aim to see the relevance, usefulness, and quality of the recommended results. The system recommends an ordered list of venues that are assessed by experts in terms of graded relevance (Eqn. 51). Precision, nDCG, average venue quality, and stability are used as evaluation metrics in this category.

6.3. Evaluation Metrics

We employ various metrics to evaluate the performance of HASVRec, as discussed below.

(a) **Precision**: Precision is defined as the fraction of retrieved items that are relevant. In our context, it is the fraction of recommended venues that are relevant.

$$Precision = \frac{|\text{relevant items} \cap \text{recommended items}|}{\text{total number of recommended items}} \tag{38}$$

Precision@k is defined when k venues are recommended, i.e.,

$$Precision@k = \frac{|\text{relevant items} \cap \text{recommended items}|}{k} \tag{39}$$

(b) **Recall**: It is another fundamental metric and is defined as the proportion of predicted relevant items in the set of actually relevant items.

$$Recall = \frac{\text{relevant items retrieved}}{\text{number of relevant items}} \tag{40}$$

(c) **F1** : It is defined as the balanced harmonic mean of precision and recall. Here we consider macro-averages for both precision and recall. The macro-average is the average of the same measures calculated for all classes. It treats all classes equally. For an individual class C_i (number of venues), if within-class true positives are tp_i , true negatives tn_i, false positives fp_i, and false negatives fn_i [54], then the following are the definitions of necessary metrics.

$$Precision_{macro} = \frac{\sum_{i=1}^{N} \frac{tp_i}{tp_i+fp_i}}{N} \tag{41}$$

$$Recall_{macro} = \frac{\sum_{i=1}^{N} \frac{tp_i}{tp_i+fn_i}}{N} \tag{42}$$

$$F1 = \frac{2Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{43}$$

(d) **Accuracy@N**:It is the ratio of no. of times a system correctly predicts the original entities within some fixed top-N recommendations for a set of test items [15, 35]. Here we consider N = 3, 6, 9, 12 and 15 respectively.

$$Accuracy@N = \frac{\text{\# times the system predicts original entities within top N}}{\text{Total number of test items}} \tag{44}$$

If N is small and/or the system is poor, it may fail to predict/recommend the original entity for a given item. Hence, we need to see it for several such items. Higher the number of papers, the better it reflects the potential of the system. The score can be any real number between 0 and 1.

(e) **Normalized discounted cumulative gain (nDCG)**: It is defined as the ratio of the discounted system gain and the discounted ideal gain accumulated at a particular rank p. The gain at some rank p is the sum of relevance values from rank 1 to rank p. Relevance value (rel_{sj}) in our context, is the score (0, 1 or 2) assigned by a researcher to the venue recommendation at position j. The ideal vector is constructed hypothetically where all relevance scores (rel_{ij}) are ordered in decreasing order to ensure the highest gain at any rank.

$$DCG_{sp} = rel_{s1} + \sum_{j=2}^{p} \frac{rel_{sj}}{log_2(j)} \tag{45}$$

$$IDCG_p = rel_{i1} + \sum_{j=2}^{p} \frac{rel_{ij}}{log_2(j)} \tag{46}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{47}$$

(f) **Mean Reciprocal Rank (MRR)**: MRR is the arithmetic mean of the reciprocal rank (RR) for all the papers in the validation/testing set. RR is defined as the inverse of the ranked position of the actual venue in the recommended result.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_{rel_i}} \tag{48}$$

where $rank_{rel_i}$ denotes the rank position of the actual venue in the recommended result for the i-th query in a query set Q.

(g) **Stability**: A recommender system is stable if the predictions do not change sharply over a short period of time [55]. It is also called the mean absolute shift (MAS) of the system. It is designed to capture the internal consistency among predictions made by a given recommendation algorithm [2].

We adopt a two-phase approach to compute the stability of a recommendation algorithm. In phase 1, let P_1 be a set of recommendations based on training data R_1, where $P_1(u, i)$ represents a system-predicted rating for user u and item i. Then a set of hypothetical incoming ratings is added to the original set of known ratings R_1. In phase 2, some subset S of predictions P_1 is added as the new incoming known ratings. Thus, in phase 2, the set of known ratings becomes $R_2 = R_1 \cup$ S and the set of unknown ratings becomes $P_2 = P_1 \backslash S$. Based on R_2, predictions on unknown ratings P_2 are made. MAS is then defined as

$$Stability = MAS = \frac{1}{|P_2|} \sum_{(u,i) \in P_2} |P_2(u,i) - P_1(u,i)| \quad (49)$$

where P_1, P_2 are the predictions made in phase 1 and phase 2, respectively.

(h) **Average-Venue Quality (Ave-quality)**: It evaluates the quality of the venues recommended by DISCOVER based on Google's h5-index [7].

$$\text{Average-venue quality} = \frac{\sum_{v \in V} H5_v}{|V|} \quad (50)$$

where V is the set of recommended venues and $H5_v$ is the h5-index of venue v. Higher the Ave-quality, the better is the recommendation.

Precision captures the overall performance of the system in terms of how many relevant venues a system can recommend. However, precision only considers whether a venue is relevant or not. In reality, the relevance of a venue can be more fine-grained or graded in terms of being exactly relevant, partially or moderately relevant, not relevant, and so on. nDCG takes into consideration this subtlety and provides an idea of system performance with respect to an ideal system that ranks the recommendation in decreasing the level of relevance. Both these metrics are bounded between 0 and 1 and are used for online evaluation. In offline evaluation, MRR is used to measure the capability of a system to predict the original entity of a test item. Although accuracy shows how often a system correctly predicts within a given rank, it does not focus on what rank it does so. MRR bridges the gap here and incentivizes the system that predicts correctly at the early ranks.

6.4. Baseline Methods

To measure the effectiveness of the proposed system, we compare HASVRec with the following state-of-the-art methods, as discussed below.

- **Collaborative Filtering Model (CF)** [15]: It is based on memory-based collaborative filtering for a given paper-venue matrix. The underlying assumption is that there is a high probability for a paper to get published in venues where other similar papers have been published.
- **Co-authorship Network-based Model (CN)** [40]: This model recommends venues based on the reputation of the author's social network and other information such as the name of the venue, sub-domain of the venue, and the number of publications.
- **Content-based Filtering Model (CBF)** [35]: The main idea behind this approach is to compute the similarity between the content mined from the researcher's publications and venue publications. The content is mined using topic modeling algorithms like LDA.
- **Personal venue Rating-based Collaborative Filtering Model (PVR)** [11]: It is based on implicit ratings given to individual venues, created from references of researchers' publications and the papers which cited researchers' past publications.
- **Random Walk with Restart Model (RWR)** [22]: It runs a Random Walk with Restart Model on a co-publication network with two types of nodes: authors and venues. This model is similar to AVER, but the probability of skipping to the next neighbor node is equal in RWR.
- **Personalized Academic venue Recommendation Model (PAVE)** [56]: It is similar to RWR, but the probability of skipping to the next neighbor node is biased using co-publication frequency, relation weight, and researchers' academic levels.
- **Publication Recommender System (PRS)** [13]: It is based on a new content-based filtering (CBF) recommendation model that uses chi-square and softmax regression.
- **Content and Network-based Academic Venue Recommender System (CNAVER)** [50]: It is based on an integrated framework employing a rank-based fusion of the paperpaper peer network (PPPN) model and the venuevenue peer network (VVPN) model.
- **Deep Learning Based Methods**: We also validate our proposed model HASVRec against a few deep learning based methods, including a CNN [57], an LSTM [30], and a Bi-LSTM [58]. While evaluating these deep learning methods, all modules (abstract, title, keywords, the field of study, and author) of the source paper are concatenated together into a long text sequence (embedding) and connected to a dense layer, like in HASVRec.

Among the methods discussed above, CF and PVR are based on collaborative filtering, PAVE and RWR are based on the random walk with restart algorithm, CN and FB are based on the co-authorship network. CBF and PRS are based on content-based filtering, CNAVER is a hybrid approach, and LSTM, CNN, and Bi-LSTM are deep learning based methods.

6.5. Parameters and Hyper-parameters Tuning

We use pre-trained GLoVE embeddings [59] to initialize the various word embeddings in our model. We set the word and author embedding dimension to be 300, and the dimension of the hidden layer of the Bi-LSTM to be 300. Thus, the forward and backward LSTMs give us 600 dimensional word/sentence annotations. The word and sentence context vectors are 300-dimensional vectors, initialized at random. One vocabulary of size 50000 is built for abstract, title, keywords, and field of study combined. A separate vocabulary of size 30000 is made for author names. The maximum number of keywords, authors, the field of study is taken as 6 each. The maximum number of words in the title is taken as 10. The maximum number of sentences in an abstract is set to 8, with a maximum of 20 words per sentence. Abstracts and titles having more or fewer sentences or words per sentence are stripped or padded accordingly.

The two hidden layers in the fully-connected dense part of the model are of size $4,096$ and $2,048$. The last hidden layer is connected to the softmax layer of size $2,921$ for offline evaluation, and $1,749$ for online evaluation, representing the total number of venues. The model is trained using the categorical cross-entropy loss function and the Adam optimizer [60]. We apply dropout regularization to avoid overfitting with the dropout rate of 0.5 [61].

The model is trained for 30 epochs using a batch size of 64 and a learning rate of 0.0003. The learning rate is scheduled to be decayed by a factor of 0.1 after every 7 epochs. Once the model is trained, we select the model with the highest accuracy on the validation set as our final model and evaluate its performance on the testing set.

6.6. Procedure of Online Evaluation

For online evaluation, we did not have a ready annotation. Thus, we collected an annotation or relevance assessment from volunteers through crowdsourcing on the best effort basis. 78 researchers with expertise in the mentioned sub-domains were provided with inputs and outputs of our recommender system where, for each paper, 15 venues are recommended. Out of 160(8 for each of the 20 sub-domains) papers in total, 29 researchers evaluated 3 papers each, 24 researchers evaluated 2 each, and the rest 25 were evaluated by 25 researchers.

All the experts were identified from academia with a minimum of 3 years of research experience. Most were having a Ph.D. except few, who were research students and research assistants pursuing a Ph.D. with a bachelor's or master's degree in science or technology. The experts or researchers were so chosen that their active areas of research perfectly match the topics of seed papers. Among the 78 researchers, there were 16 professors, 15 associate professors, 23 assistant professors, 18 senior research students, and the remaining 6 were research assistants. The ages of all professors were in the range of [48-55], ages of associate professors were in between [43-47], assistant professors were having an age range of [36-41], senior research students were in the age range of [28-31], and the remaining research assistants were having an age range of [29-33]. The overall gender distribution of male and female experts was 51 and 27, respectively.

The experts looked at the titles, abstracts, authors, year of publication, and recommended venues of the papers and assigned an appropriate relevance value (r) to each recommended venue evaluating the quality of the match between the scope of the recommended venue and the topic of the seed paper as below.

$$\text{Relevance } (r) = \begin{cases} 2 & \text{perfectly matching} \\ 1 & \text{partial matching} \\ 0 & \text{otherwise} \end{cases} \tag{51}$$

However, as precision is defined for binary relevance only, during precision score computation, only relevance grade 2 is considered relevant, and both relevance grade 1 and 0 non-relevant.

To comprehensively evaluate our proposed method and to address the existing issues discussed in Sec. 1, we examine the following research queries (RQs):

RQ1: How effective is HASVRec in comparison to other state-of-the-art methods?

RQ2: How is the quality of venues recommended by HASVRec as compared to state-of-the-art methods?

RQ3: How does HASVRec handle cold-start issues and other issues like data sparsity, and stability?

6.7. Results and Discussion

In this section, we report the performance of HASVRec against the existing state-of-the-art methods. For clarity and easy understanding, we provide the results in two subsections (offline and online), as given below. We also conduct paired-samples t-test on overall accuracy, MRR, precision, $F1$, MAS (stability), nDCG between HASVRec, and the second-best performers. Only p values less than 0.05 are considered statistically significant at 5% level of significance (α =0.05). The statistically significant results and the second-best performer are marked by the '*' and '+' symbols in each position.

6.7.1. Offline Evaluation of HASVRec

In this section, we examine the performance of the proposed HASVRec model in terms of evaluation metrics such as accuracy, MRR, and $F1$. The complete results of accuracy and MRR are depicted in Table 7. The proposed HASVRec shows a consistent performance improvement over all other standard approaches. HASVRec can predict the original venue of the seed paper within the top 3 recommendations with an accuracy of 0.6625.

Initially, HASVRec gives an accuracy of 0.7943 at position 6. Then slowly, it shows an upward trend and exhibits an excellent performance with an accuracy of 0.9481 at position 15. CNAVER performs the second-best among all other standard approaches (except deep learning methods)

Table 7: Accuracy and MRR results of HASVRec and other compared approaches

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
CF	0.0972	0.1111	0.1527	0.1805	0.2361	0.0451
CN	0.1111	0.1388	0.1805	0.2222	0.2500	0.0516
CBF	0.1703	0.2094	0.2273	0.2569	0.3041	0.0751
RWR	0.1976	0.2365	0.2645	0.2997	0.3308	0.0875
PVR	0.2083	0.2361	0.2368	0.3194	0.3472	0.0947
PAVE	0.2642	0.2916	0.3055	0.3927	0.4676	0.0994
PRS	0.2608	0.3046	0.3519	0.4182	0.5786	0.1497
CNAVER	0.5107+	0.6552+	0.7204+	0.8085+	0.8494+	0.2432+
CNN	0.5671	0.6786	0.7553	0.7874	0.8369	0.3874
LSTM	0.5569	0.6749	0.7459	0.7856	0.8352	0.3763
Bi-LSTM	0.5776	0.6978	0.7657	0.8067	0.8488	0.3995
HASVRec	0.6625*	0.7943*	0.8788*	0.9037*	0.9481*	0.5047*

'*' denote statistically significant results over the second best ('+')

Table 8: Macro-average analysis in terms of $F1$

Approach	$F1$@3	$F1$@6	$F1$@9	$F1$@12	$F1$@15
CF	0.0317	0.0394	0.0674	0.0642	0.0585
CN	0.0478	0.0562	0.0631	0.0594	0.0523
CBF	0.0703	0.0789	0.1028	0.0895	0.1156
RWR	0.1073	0.1284	0.2006	0.1781	0.1687
PVR	0.1149	0.1437	0.1673	0.1752	0.1692
PAVE	0.1512	0.1987	0.2291	0.2085	0.1964
PRS	0.1392	0.1673	0.1955	0.1840	0.1897
CNAVER	0.2392+	0.2507+	0.2512+	0.2167+	0.2045+
CNN	0.2678	0.2623	0.2234	0.1892	0.1692
LSTM	0.2788	0.2595	0.2204	0.1981	0.1682
Bi-LSTM	0.2967	0.2704	0.2306	0.2017	0.1895
HASVRec	0.3496*	0.3185*	0.2989*	0.2703*	0.2412*

'*' denote statistically significant results over the second best ('+')

Table 9: Precision of HASVRec and other compared approaches

Methods	P@3	P@6	P@9	P@12	P@15
CF	0.6234	0.6456	0.6389	0.6492	0.6376
CN	0.6793	0.6742	0.6696	0.6705	0.6623
CBF	0.7267	0.7243	0.7139	0.7129	0.7005
RWR	0.7382	0.7275	0.7081	0.7089	0.7073
PVR	0.8031	0.7654	0.7468	0.7549	0.7592
PRS	0.7943	0.8255	0.7829	0.7671	0.7747
PAVE	0.8144	0.8021	0.8286	0.8192	0.8174
CNAVER	0.8477+	0.8592+	0.8847+	0.8707+	0.8621+
CNN	0.8774	0.8746	0.8683	0.8571	0.8496
LSTM	0.8696	0.8708	0.8669	0.8569	0.8423
Bi-LSTM	0.8726	0.8867	0.8823	0.8672	0.8577
HASVRec	0.9485*	0.9671*	0.9706*	0.9607*	0.9479

'*' denote statistically significant results over the second best ('+')

Table 10: nDCG of HASVRec and other compared approaches

Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
CF	0.6583	0.6694	0.6698	0.6578	0.6542
CN	0.6877	0.6895	0.6903	0.6852	0.6794
CBF	0.7398	0.7384	0.7412	0.7307	0.7207
RWR	0.7499	0.7494	0.7437	0.7502	0.7562
PVR	0.7538	0.7463	0.7579	0.7498	0.7436
PRS	0.7649	0.7675	0.7763	0.7564	0.7649
PAVE	0.8073	0.7965	0.7995	0.7852	0.7847
CNAVER	0.8352+	0.8456+	0.8612+	0.8507+	0.8489+
CNN	0.8696	0.8685	0.8491	0.8426	0.8467
LSTM	0.8735	0.8662	0.8554	0.8397	0.8395
Bi-LSTM	0.8795	0.8717	0.8583	0.8459	0.8527
HASVRec	0.9582*	0.9683*	0.9716*	0.9583*	0.9551*

'*' denote statistically significant results over the second best ('+')

at positions 3, 6, and 9, respectively. The worst performance among all methods is shown by the CF method. Similarly, during the evaluation of MRR, HASVRec outperforms all other state-of-the-art methods and shows excellent behavior with an MRR result of 0.5047.

We also investigate the efficacy of the proposed HASVRec model against other state-of-the-art methods in terms of $F1$ Scores. The complete results showing $F1$ scores for different methods are shown in Table 8. HASVRec shows an upward trend at the beginning and achieves the $F1$ of 0.3496 at position 3. Then slowly, it decreases and reaches a $F1$ of 0.2412 at position 15. The analysis shown in Table 8 demonstrates the efficacy of HASVRec in terms of $F1$ to other state-of-the-art methods.

6.7.2. *Online Evaluation of HASVRec*

In this section, we examine the performance of the proposed HASVRec model in terms of evaluation metrics such as precision and nDCG. The overall results of the precision evaluation are shown in Table 9. We can see the improvement of HASVRec in terms of precision over all other standard approaches. Initially, the proposed HASVRec exhibits a precision of 0.9485 at position 3. After that, it slightly shows an upward trend and gives a precision of 0.9706 at position 9, and finally shows a precision of 0.9479 at position 15. HASVRec exhibits the highest precision of 0.9706, and a lower precision value of 0.9479 at positions 9 and 15, respectively. CNAVER performs the second-best among all other state-of-the-art methods, whereas the worst performance is shown by the CF method among all other state-of-the-art methods.

The overall results of nDCG@k evaluation of all methods are shown in Table 10. The overall nDCG results of HASVRec are consistent and show a significant improvement in nDCG over all other state-of-the-art methods. During the initial recommendations, the proposed model HASVRec gives an nDCG of 0.9582 at a position 3. Then it shows an upward trend and reaches an nDCG of 0.9716 at position 9, and afterwards, it shows a downward trend and reaches an nDCG of 0.9551 at position 15.

7. Study of the Proposed Approach

The main findings concerning our RQs as introduced in Sec. 6.6 are summarized below:

7.1. *RQ1: How Effective is HASVRec in Comparison to Other State-of-the-art Methods?*

We investigate the overall results, including the precision@k, nDCG@k, accuracy, $F1$, and MRR of the proposed HASVRec model and all other state-of-the-art techniques. HASVRec demonstrates the best execution while assessing promising outcomes in higher values of precision@k and nDCG@k, separately. Additionally, the performance of HASVRec in terms of accuracy, $F1$, and MRR demonstrates that the proposed approach has measurably significant results over all other state-of-the-art techniques. The outcomes are shown in Table 7, 8, 9, and 10 respectively.

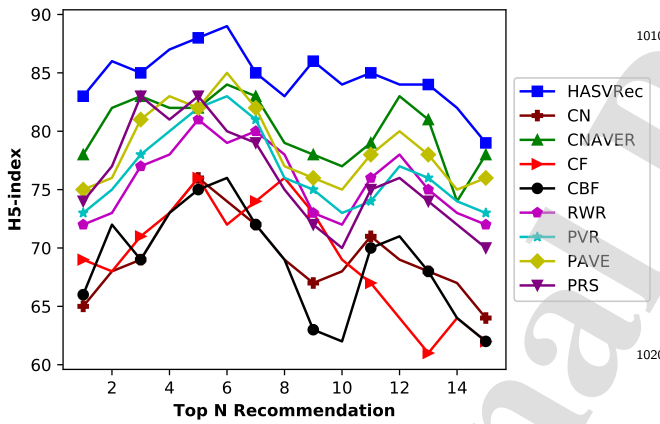


Figure 5: Venue quality of HASVRec and other approaches

.

7.2. *RQ2: How is the Quality of Venues Recommended by HASVRec as Compared to State-of-the-art Methods?*

We investigate the performance of venue quality recommended by HASVRec compared to other existing approaches, as depicted in Fig. 5. Overall, the average H5-Index of venues recommended by the HASVRec model is 85. The top-quality venues recommended by the HASVRec model are at position 6 with the highest H5-Index of 89. The most noteworthy H5-index recommended by HASVRec is 89, and the least is 79, whereas the most noteworthy H5-index suggested by CNAVER is 84, and the least is 74. CNAVER recommends venues having the second-best H5-index. The CF model provides the least quality recommendation.

7.3. *RQ3: How Does HASVRec Handle Cold-start Issues and Other Issues like Data Sparsity, and Stability?*

(i) **Cold-start Issues**: To address "cold-start" issues like new researchers and venues, we integrate extracted high-level features from abstract, title, keywords, the field of study, and authors along with their publication records by hierarchically applying an attention-based Bi-LSTM model. We investigate the performance of HASVRec for inputs (seed papers) associated with new researchers and venues. The examination in Table 11 reflects that, regardless of whether the seed paper is related to a new researcher and venue, HASVRec can anticipate the original venue at an early stage of recommendations. Even if the researchers' past publication records are sparse, HASVRec works fine because it uses a modular structure to greatly focus on the work at hand. Thus, HASVRec does not have the cold-start issues, as described in Table 12.

(ii) **Data Sparsity**: To address the "data sparsity" issue, we explicitly employ an attention-based Bi-LSTM model to capture the quality of essentialness, relevance and to extract low dimensional latent factors of high dimensional input. The LSTM model is specifically designed to process temporal, latent contextual aspects of high dimensional and sparse input. For example, the input dimension of the abstract is $8 * 20 * 300$, but after applying a word-level attention-based Bi-LSTM, it is reduced into an $8*600$-dimensional vector. Then, after applying sentence-level attention, we obtain a 600-dimensional abstract representation. The input dimension of the title is $10 * 300$, but after applying a word-level attention-based Bi-LSTM, it is reduced to 600. Similarly, the input dimension of keywords is $6 * 300$, and the output is a 600-dimensional vector when an attention-based Bi-LSTM is applied to it. The input dimension of the field of study is $6 * 300$, but after applying a word-level attention-based Bi-LSTM, it is reduced to a dimension of 600. The output dimension of the author is reduced to 300-dimensional vector from its input dimension of $6 * 300$ when an attention-based Bi-LSTM is applied. The output dimension of the venue is reduced to 300-dimensional vector from its input dimension of $5*300$ when an attention-based Bi-LSTM is applied. After applying an attentive pooling layer on each of the individual output modules, we get the final paper-level representation M, which is a 600-dimensional vector. Hence, HASVRec effectively performs dimensionality reduction and solves the problem of data sparsity.

(iii) **Stability**: To deal with the stability issue, we attempt to design a novel and unified architecture which contains an LSTM layer and an attention layer to cope with stability issue. The attention layer is employed to understand the sentiment and semantics of

Table 11: MRR results for HASVRec over new venue and new researcher

Methods	$50{<=}v_c{<}100$	$1005{<=}v_c{<}200$	$200{<=}v_c{<}400$	$400{<=}v_c$	$5{<=}p_c{<}15$	$15{<=}p_c{<}50$	$50{<=}p_c{<}100$	$100{<=}p_c$
CF	0.0473	0.0637	0.0725	0.0753	0.0746	0.0828	0.0866	0.0887
CN	0.0526	0.0773	0.0866	0.0899	0.0877	0.0914	0.0957	0.0974
CBF	0.0668	0.0848	0.0932	0.1094	0.0817	0.0895	0.0917	0.1095
RWR	0.0793	0.0849	0.0853	0.0864	0.0861	0.0874	0.0891	0.1064
PVR	0.0798	0.0869	0.0851	0.0853	0.0893	0.0847	0.0893	0.0847
PRS	0.0972	0.0995	0.1049	0.1128	0.0903	0.0945	0.0963	0.0995
PAVE	0.0977	0.1014	0.1298	0.1416	0.1091	0.1373	0.1539	0.1873
CNAVER	0.1449+	0.1745+	0.1984+	0.2064+	0.2249+	0.2161+	0.2293+	0.2549+
HASVRec	0.4156*	0.4659*	0.5069*	0.5477*	0.3992*	0.4462*	0.4719*	0.5571*

'*' denote statistically significant results over the second best ('+')

Table 12: Cold-start and other issues available

Methods	Cold-start	Sparsity	Stability
CF	yes (researcher and venue)	yes	yes
CN	yes (new venue)	no	yes
CBF	yes(new venue)	no	no
RWR	yes (new researcher)	no	yes
PVR	yes (researcher and venue)	yes	yes
PAVE	yes(new researcher)	no	yes
PRS	yes(new venue)	no	no
CNAVER	no	no	no
HASVRec	no	no	no

Table 13: Stability (MAS) of HASVRec and other approaches

Methods	MAS
CF	7.865
CN	8.339
CBF	4.575
RWR	6.965
PVR	7.165
PAVE	6.594
PRS	4.312
CNAVER	4.138+
HASVRec	3.010*

'*' denote statistically significant results over the second-best ('+')

the input text to get essential words and sentences of a seed paper. This helps in increasing the stability of predictions. In this paper, we have provided a comprehensive investigation into the stability of the recommendation algorithm, as defined in Eqn. 49. As shown in Table 13, HASVRec shows the minimum MAS than all other standard approaches. It shows a MAS of 3.010 on the DBLP dataset, meaning that, on average, every predicted rating will shift by 3.010 after adding another 3% of testing data into the training data. We have considered the average MAS-score as a threshold to decide whether a particular method provides stability or not.

Table 14: Impact of Internal Structure of the Model

Models	**Accuracy**	**Decline**
HASVRec	94.81%	-
w/o Attention Mechanism	90.43%	4.38% ↓
w/o Modularized Structure	85.83%	8.98% ↓

'↓' indicates the decline in results than HASVRec

7.4. Impact of Internal Structure of the Model

The underlying intuition of our model is that it incorporates knowledge of the structure of the source paper and automatically selects the most informative words that are likely to be helpful to make accurate predictions. To investigate the impact of the internal structure of the model, we remove the modularized hierarchical structure and attention mechanism in steps. We employ two strategies: w/o attention mechanism (modularized structure but without attention mechanism), and w/o modularized structure (do not use both the modularized structure and the attention mechanism) to perform such tasks (Table 14).

When we remove the attention mechanism from the model, the accuracy drops by 4.38%. This reflects that different modules contribute to a paper differently. Generally, the abstract provides a summary containing the main idea of a source paper and is more important than its title. The attention mechanism can automatically decide which part is more critical in determining the original venue of the source paper. When we remove the modularized structure, the accuracy drops by 8.98%. This proves that by incorporating the modularized hierarchical structure of the source paper, the model becomes capable of making more accurate predictions.

7.5. Impact of Modules of the Source Paper

The underlying intuition of our model is that not all parts of a paper are equally relevant to decide the appropriate venues for a given source paper. To investigate this issue, we remove each module from the input text of the source paper and observe the change in the performance of HASVRec (Table 15). The performance of HASVRec shows different degrees of decline during experimentations. For example, when we remove the title module, the accuracy drops by 0.92%.

Apart from the abstract of the paper, the two modules that maximally affect the probability of predicting the original venue are the field of study and keywords of a source paper. The impact of the field of study module is the largest, and removing it results in an accuracy decrement of 1.53%. This shows that the field of study of the source paper largely determines the original venue, and is intuitive as the field of study can provide the narrow subarea of a given source paper and can primarily distinguish the venues having similar domains. Also, one might think

Table 15: Modules impact on overall accuracy of HASVRec

Contexts	Accuracy	Decline
All modules (HASVRec)	94.81%	-
w/o Title	93.89%	0.92% ↓
w/o Keywords	93.68%	1.13% ↓
w/o Field of Study	93.28%	1.53% ↓
w/o Authors	93.77%	1.04% ↓
w/o Venues	93.86%	0.95% ↓

'↓' indicates the decline in results than HASVRec

that the author module can largely determine the probability of predicting the correct venue. However, without the author module, there is a decline of only 1.04%, owing to the high variance of this module, meaning that the authors of a source paper vary widely. Similarly, without the venue module, there is a decline of only 0.95%, owing to the high variance of this module, meaning that the venues of a source paper vary widely.

7.6. More Insightful Discussion on the Results

The overall performance results obtained and discussed in Sec. 6.7 showcase the efficacy of the proposed HASVRec. However, there are a few limitations to our work. The proposed HASVRec system includes multiple parameters involved in the embedding layer, the module-level representation layer, the attention layer, and the dense layer, requiring rigorous experimentation. There are multiple classes to classify, and each class requires a vast amount of data to train the model. It eventually leads to an increase in computation costs. HASVRec may not recommend suitable venues if there are less number of related papers (number of papers published in a venue) that exist in the training dataset.

8. Conclusion

Academic venue recommendation is an emerging area of research in recommender systems. The proposed techniques are few in numbers and suffer from various limitations. One of the major issues is that of cold-start for a new venue and a new researcher. Additionally, there are problems of sparsity and stability in venue recommender systems that are not adequately addressed by existing state-of-the-art methods. This paper proposes HASVRec: A deep learning-based scholarly venue recommender system. It is an integrated framework comprising of a Bi-LSTM with a hierarchical attention mechanism. It makes scholarly venue recommendations using a combination of modularized hierarchical patterns such as abstract, title, keywords, field of study, authors, and venues of the source paper. HASVRec only requires the abstract, title, keywords, field of study, and authors of a source paper along with their publication records to identify scholarly venues.

HASVRec reasonably addresses all the specified issues. We have conducted an extensive set of experiments on a real dataset: DBLP-Citation-Network V11, and showed that HASVRec consistently outperforms the state-of-the-art methods in terms of precision@k. It demonstrates substantially higher scores of nDCG@k, accuracy, $F1$, and MRR over other best-in-class techniques. HASVRec proposes top-notch venues better than those recommended by other state-of-the-art methods as far as H5-index is concerned, along with better stability. We also observed that apart from the abstract of the paper, the two modules that maximally affect the probability of predicting the original venue are the field of study and keywords of a source paper. Both modularized structure information and attention mechanism are likely to be helpful to make accurate prediction of original venues.

In the future, we intend to investigate how to apply various meta-path features and other deep learning methods such as autoencoders to fuse the embeddings of multiple paths better. Also, user feedback can be taken into account in our recommender system, and experiments can be conducted on ensembling heterogeneous bibliographic datasets. Therefore, it is exciting and natural to explore the domain of venue recommendation, incorporating controllability of different parameters to enhance its precision, accuracy, diversity, novelty, coverage, serendipity even further.

9. Compliance with Ethical Standards

The authors declare no conflicts of interest. The article does not contain any examinations with human or creature subjects.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE transactions on knowledge and data engineering 17 (6) (2005) 734–749.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-based systems 46 (2013) 109–132.

[3] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decision Support Systems 74 (2015) 12–32.

[4] X. Zhou, W. Liang, I. Kevin, K. Wang, R. Huang, Q. Jin, Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data, IEEE Transactions on Emerging Topics in Computing.

[5] X. Kong, M. Mao, W. Wang, J. Liu, B. Xu, Voprec: Vector representation learning of papers with text information and structural identity for recommendation, IEEE Transactions on Emerging Topics in Computing.

[6] W. Huang, Z. Wu, L. Chen, P. Mitra, C. L. Giles, A neural probabilistic model for context based citation recommendation., in: AAAI, 2015, pp. 2404–2410.

[7] S. Yu, J. Liu, Z. Yang, Z. Chen, H. Jiang, A. Tolba, F. Xia, Pave: Personalized academic venue recommendation exploiting co-publication networks, Journal of Network and Computer Applications 104 (2018) 38–47.

[8] M. M. U. Rathore, M. J. J. Gul, A. Paul, A. A. Khan, R. W. Ahmad, J. Rodrigues, S. Bakiras, Multilevel graph-based decision making in big scholarly data: An approach to identify

expert reviewer, finding quality impact factor, ranking journals and researchers, IEEE Transactions on Emerging Topics in Computing.

[9] N. Y. Asabere, B. Xu, A. Acakpovi, N. Deonauth, Sarve-2: Exploiting social venue recommendation in the context of smart conferences, IEEE Transactions on Emerging Topics in Computing.

[10] M. C. Pham, D. Kovachev, Y. Cao, G. M. Mbogos, R. Klamma, Enhancing academic event participation with context-aware and social recommendations, in: Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on, IEEE, 2012, pp. 464–471.

[11] H. Alhoori, R. Furuta, Recommendation of scholarly venues based on dynamic user interests, Journal of Informetrics 11 (2) (2017) 553–563.

[12] H. P. Luong, T. Huynh, S. Gauch, K. Hoang, Exploiting social networks for publication venue recommendations., in: KDIR, 2012, pp. 239–245.

[13] D. Wang, Y. Liang, D. Xu, X. Feng, R. Guan, A content-based recommender system for computer science publications, Knowledge-Based Systems 157 (2018) 1–9.

[14] N. M. Villegas, C. Sánchez, J. Díaz-Cely, G. Tamura, Characterizing context-aware recommender systems: A systematic literature review, Knowledge-Based Systems 140 (2018) 173–200.

[15] Z. Yang, B. D. Davison, Venue recommendation: Submitting your paper with style, in: Machine learning and applications (ICMLA), 2012 11th international conference on, Vol. 1, IEEE, 2012, pp. 681–686.

[16] T. Huynh, K. Hoang, Modeling collaborative knowledge of publishing activities for research recommendation, Computational collective intelligence. Technologies and applications (2012) 41–50.

[17] D. Liang, L. Charlin, J. McInerney, D. M. Blei, Modeling user exposure in recommendation, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 951–961.

[18] N. Kang, M. A. Doornenbal, R. J. Schijvenaars, Elsevier journal finder: recommending journals for your paper, in: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, 2015, pp. 261–264.

[19] M. J. Schuemie, J. A. Kors, Jane: suggesting journals, finding experts, Bioinformatics 24 (5) (2008) 727–728.

[20] G. Wang, X. He, C. I. Ishuga, Har-si: A novel hybrid article recommendation approach integrating with social information in scientific social network, Knowledge-Based Systems 148 (2018) 85–99.

[21] T. Silva, J. Ma, C. Yang, H. Liang, A profile-boosted research analytics framework to recommend journals for manuscripts, Journal of the Association for Information Science and Technology 66 (1) (2015) 180–200.

[22] Z. Chen, F. Xia, H. Jiang, H. Liu, J. Zhang, Aver: random walk based academic venue recommendation, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 579–584.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 2016, pp. 1480–1489.

[24] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, J. Wang, Dynamic attention deep model for article recommendation by learning human editors' demonstration, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 2051–2059.

[25] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (1) (2019) 5.

[26] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies, Artificial Intelligence Review 52 (1) (2019) 1–37.

[27] F. Dong, Y. Zhang, J. Yang, Attention-based recurrent convolutional neural network for automatic essay scoring, in: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), 2017, pp. 153–162.

[28] P. Yang, X. Sun, W. Li, S. Ma, Automatic academic paper rating based on modularized hierarchical convolutional neural network, arXiv preprint arXiv:1805.03977.

[29] M. Sundermeyer, R. Schlüter, H. Ney, Lstm neural networks for language modeling, in: Thirteenth annual conference of the international speech communication association, 2012.

[30] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[31] W. Yin, K. Kann, M. Yu, H. Schütze, Comparative study of cnn and rnn for natural language processing, arXiv preprint arXiv:1702.01923.

[32] Z. Yang, B. D. Davison, Distinguishing venues by writing styles, in: Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries, ACM, 2012, pp. 371–372.

[33] Z. Yang, D. Yin, B. D. Davison, Recommendation in academia: A joint multi-relational model, in: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, IEEE, 2014, pp. 566–571.

[34] J. Yu, K. Xie, H. Zhao, F. Liu, Prediction of user interest based on collaborative filtering for personalized academic recommendation, in: Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, IEEE, 2012, pp. 584–588.

[35] E. Medvet, A. Bartoli, G. Piccinin, Publication venue recommendation based on paper abstract, in: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, IEEE, 2014, pp. 1004–1010.

[36] M. Errami, J. D. Wren, J. M. Hicks, H. R. Garner, etblast: a web server to identify expert reviewers, appropriate journals and similar publications, Nucleic acids research 35 (suppl_2) (2007) W12–W15.

[37] R. Klamma, P. M. Cuong, Y. Cao, You never walk alone: Recommending academic events based on social network analysis, in: International Conference on Complex Sciences, Springer, 2009, pp. 657–670.

[38] M. C. Pham, Y. Cao, R. Klamma, Clustering technique for collaborative filtering and the application to venue recommendation, in: Proc. of I-KNOW, Citeseer, 2010.

[39] M. C. Pham, Y. Cao, R. Klamma, M. Jarke, A clustering approach for collaborative filtering recommendation using social network analysis., J. UCS 17 (4) (2011) 583–604.

[40] H. Luong, T. Huynh, S. Gauch, L. Do, K. Hoang, Publication venue recommendation using author networks publication history, Intelligent Information and Database Systems (2012) 426–435.

[41] T. Pradhan, S. Pal, A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity, Future Generation Computer Systems.

[42] I. Boukhris, R. Ayachi, A novel personalized academic venue hybrid recommender, in: Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on, IEEE, 2014, pp. 465–470.

[43] X. Feng, H. Zhang, Y. Ren, P. Shang, Y. Zhu, Y. Liang, R. Guan, D. Xu, The deep learning–based recommender system pubmender for choosing a biomedical publication venue: Development and validation study, Journal of medical Internet research 21 (5) (2019) e12957.

[44] Q. Zhang, D. Wu, J. Lu, F. Liu, G. Zhang, A cross-domain recommender system with consistent information transfer, Decision Support Systems 104 (2017) 49–63.

[45] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.

[46] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, arXiv preprint arXiv:1205.2618.

[47] E. Minkov, B. Charrow, J. Ledlie, S. Teller, T. Jaakkola, Col-

laborative future event recommendation, in: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, 2010, pp. 819–828.

[48] L. Yang, Y. Zheng, X. Cai, H. Dai, D. Mu, L. Guo, T. Dai, A lstm based model for personalized context-aware citation recommendation, IEEE access 6 (2018) 59618–59627.

[49] F. Xia, N. Y. Asabere, J. J. Rodrigues, F. Basso, N. Deonauth, W. Wang, Socially-aware venue recommendation for conference participants, in: Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), IEEE, 2013, pp. 134–141.

[50] T. Pradhan, S. Pal, Cnaver: A content and network-based academic venue recommender system, Knowledge-Based Systems (2019) 105092.

[51] H. Sak, A. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Fifteenth annual conference of the international speech communication association, 2014.

[52] A. Géron, Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems, " O'Reilly Media, Inc.", 2017.

[53] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 990–998.

[54] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information processing & management 45 (4) (2009) 427–437.

[55] G. Adomavicius, J. Zhang, Improving stability of recommender systems: a meta-algorithmic approach, IEEE Transactions on Knowledge and Data Engineering 27 (6) (2014) 1573–1587.

[56] S. Yu, J. Liu, Z. Yang, Z. Chen, H. Jiang, A. Tolba, F. Xia, Pave: Personalized academic venue recommendation exploiting co-publication networks, Journal of Network and Computer Applications.

[57] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[58] S. Zhang, D. Zheng, X. Hu, M. Yang, Bidirectional long short-term memory networks for relation classification, in: Proceedings of the 29th Pacific Asia conference on language, information and computation, 2015, pp. 73–78.

[59] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[60] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[61] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

AUTHORSHIP STATEMENT

Manuscript title: HASVRec: A modularized Hierarchical Attention-based Scholarly Venue Recommender system

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the journal of **Knowledge-Based System**.

Authorship contributions

Please indicate the specific contributions made by each author (list the authors' initials followed by their surnames, e.g., Y.L. Cheung). The name of each author must appear at least once in each of the three categories below.

Category 1

Conception and design of study: Tribikram Pradhan, Abhinav Gupta;

Acquisition of data: Tribikram Pradhan, Abhinav Gupta;

Analysis and/or interpretation of data: Tribikram Pradhan, Abhinav Gupta;

Category 2

Drafting the manuscript: Tribikram Pradhan, Sukomal Pal, Abhinav Gupta;

Revising the manuscript critically for important intellectual content: Tribikram Pradhan, Sukomal Pal;

.

Category 3

Approval of the version of the manuscript to be published (the names of all authors must be listed):
Tribikram Pradhan, Abhinav Gupta, Sukomal Pal

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

This statement is signed by all the authors (*a photocopy of this form may be used if there are more than 10 authors***):**

Author's name (typed)	Author's sign	Date
Tribikram Pradhan	Tribikram Pradhan	17/04/2020
Abhinav Gupta	Abhinav Gupta	17/04/2020
Sukomal Pal	Sukomal Pal	17/04/2020

Conflict of Interest and Authorship Conformation Form

Please check the following as appropriate:

- All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

- This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

- The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript

- The following authors have affiliations with organizations with direct or indirect financial interest in the subject matter discussed in the manuscript:

Author's name	Affiliation
Tribikram Pradhan-	Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, India
Abhinav Gupta-	Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, India
Sukomal Pal -	Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, India