Chapter 5

DeepRec: A Deep Learning-based Journal Recommender System

"The essence of trust building is to emphasize the similarities between you and the customer."

-Thomas watson (1874-1956)

5.1 Introduction

We employed two kinds of analysis: citation analysis and contextual similarity analysis in both DISCOVER, and CNAVER. These processes require good amount both space and time to store and organize shortlisted papers properly (by storing title, abstract, and citations relationship among papers). CNAVER is also sensitive to the structure of bibliographic citation network and may result in some irrelevant recommendations. Both DISCOVER and CNAVER approaches addresses cold start issues, diversity, and scalability issues to some great extent. However, relevance (accuracy in recommending relevant venues), stability, and sparsity issues are not adequately addressed.

Recently due to the ability to discover intricate structure and deep semantics in high dimensional data, deep-learning approaches have succeeded in many areas of recommender system such as cross-domain recommendation [191], web recommendation [192], query recommendation [193], tag recommendation [194], e-learning recommendation [195], recommender system for medical diagnosis [196], recommender system for researcher [197]. Besides, due to multiple processing layers, deep learning models can able to learn multiple-abstract representations of data to capture both syntactic and semantic information [135]. One of the primary usages of deep learning techniques in the recommender system is to enhance the accuracy of the overall recommendations. Since deep learning techniques are mainly used to extract hidden features, researchers utilize them to obtain latent factors.

Therefore, we propose DeepRec: a stacked generalized ensemble learning-based scholarly venue recommender system to address this challenging task. Our ensemble learning-based model is elaborately designed based on a Convolution Neural Network (CNN), and Long Short-Term Memory (LSTM). CNN is mainly adopted to extract local structure of the data, while LSTM can capture the temporal correlation and dependencies in the text snippet.

To enhance the recommendation quality in terms of relevance we extract latent features from abstract and title with CNN and LSTM model and combined them into the proposed model DeepRec. To address data sparsity issue, we transformed high dimensional and sparse embedding matrix into a lower-dimensional and dense set using CNN based deep learning technique. CNN is specifically designed to process temporal, latent contextual aspects of high dimensional and sparse input. The stacked generalized ensemble learning model also helps to maintain a stability by capturing the relevance of papers. Here for contextual similarity, both abstract and title are considered.

5.2 Problem Descriptions

Definition 9 *Modular Structure of Research Paper. Generally, a paper p_m is divided into various modules such as Abstract, Title, and so on. Let R_1 denotes the abstract and R_2 denotes the title of the source paper p_m. The module R_1 is consists of a few sentences, and each sentence is composed of few words. Similarly, a R_2 is composed of few words.*

5.3 Functional Architecture of DeepRec

We present a systematized framework of the proposed ensemble model DeepRec alongside its operational strategies. We first describe the functional architecture of our model and then introduce various layers to provide a detail description. DeepRec is comprised of majorly three blocks, namely Data Preprocessing (Block-1), Computational Learning

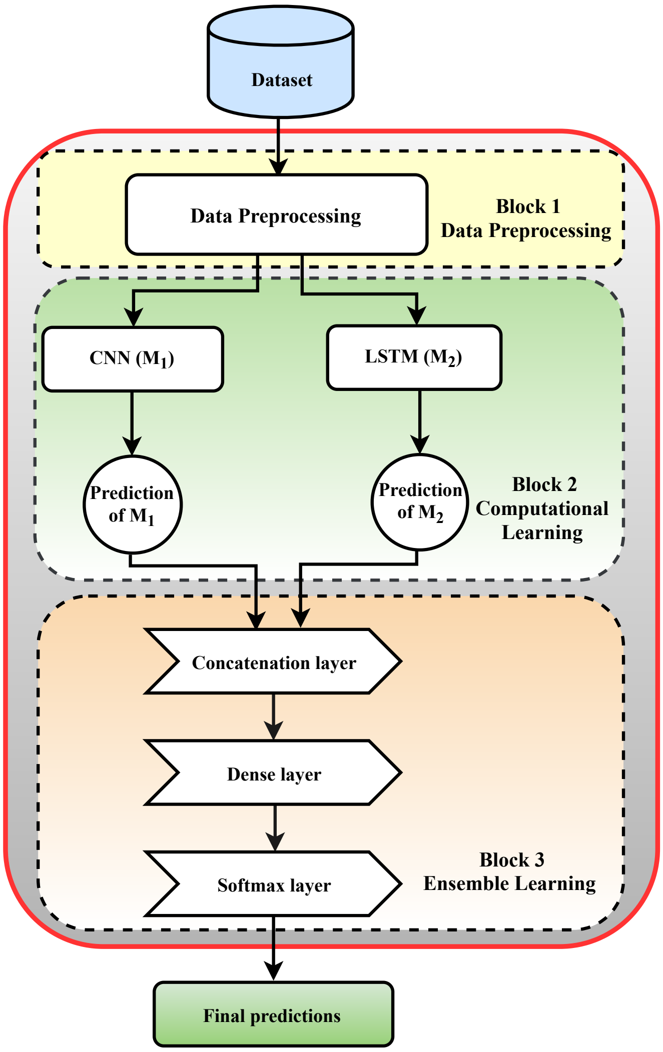


Figure 5.1: Functional architecture of DeepRec

(Block-2), and Ensemble Learning (Block-3) as depicted in Fig. 5.1. These three primary blocks are portrayed as given underneath:

A. **Data Preprocessing (Block-1)**: This step aims to structure, arrange, and organize the dataset suitable for further processing. This layer is also called the feature extraction layer as it is mainly introduced to extract abstract and title as relevant features for further use.

B. **Computational Learning (Block-2)**: This block is introduced to apply a deep learning model to identify relevant venues for a given seed paper. This layer comprises of two distinct models, for example:

   (i) **Convolution Neural Network (CNN) model**: The CNN model consists of various components that can transform the input volume into an output volume, namely embedding layer, convolution layers, dense layer, and a softmax layer. These layers are stacked to form a deep convolution neural network (CNN) and can be utilized multiple times to provide optimum recommendations.

(ii) **Long Short-Term Memory (LSTM) model**: The LSTM network has the capability to resolve the issues of long time dependencies and gradient vanishing. It makes use of various gates to manipulate the data flow in the recurrent neural unit. Gates are layers that are carried out multiplicatively and therefore, can either preserve the value from the gated layer if the gates are 1 or 0 this value if the gate is 0. It essentially makes use of three varieties of gates, namely forget gate, input gate, and output gate.

C. **Ensemble Learning (Block-3)**: The stacking ensemble model takes the output of both the sub-models such as CNN and LSTM as inputs and attempts to learn how to combine inputs best to get better output results. The concept of stacking is to examine the individual learners, which includes CNN and LSTM, and integrate them with the aid of training a meta-version to output predictions primarily based on multiple predictions returned via those weak models. The final ranking of scholarly venues is done based on the trained results of stacked generalization to leverage the advantages of both the models.

5.3.1 Data Preprocessing (Block-1)

We collected the data from DataBase systems and Logic Programming(DBLP). Dataset originally contained 2,236,968 research papers from 4,565 Journals and Conferences after removing rows containing missing values. Dataset contains 'abstract', 'authors', 'id', 'references', 'title', 'venue', and 'year' columns. We used 'abstract', 'title', and 'venue' for our experiments. Dataset was filtered by removing venues having less than 5 number papers. The final dataset was left with 2,234,771 paper and 3,216 venues.

Text (titles+abstracts) were first concatenated and then cleaned. Cleaning of text involved converting this text to lower case, removing stopwords, punctuation, and lemmatizing the verbs in the text. Once the text has cleaned, this text was converted to sequence by Keras-Tokenizer setting maximum words to 5,000. Then this sequence was padded to a maximum length of 300. Output (venue) was processed by first encoding the venue names to labels by sklearn-LabelEncoder, and then these labels were converted to one-hot vectors by sklearn-OneHotEncoder to feed into algorithm.

5.3.2 Computational Learning (Block-2)

This block is introduced to apply a deep learning model to identify appropriate venues for a given seed paper. In this step, both CNN model and LSTM model are used individually for the same given input, and results are stored in order to prepare the final lists of venues. We have discussed the CNN model, followed by the LSTM model.

Architecture of CNN Model

Fig. 5.2 represents the functional architecture of the CNN model. CNN model consists of various components that can reduce the high dimensional input data into a lower-dimensional output data via Embedding layer, Feature extraction layer, and Dense layer. These layers are stacked to form a deep CNN and can be utilized multiple times to provide optimum recommendations. In our model, there are three convolution and max-pooling layers, one flattened layer, two hidden layers, and one softmax layer for classification. The reason behind adopting such CNN model in DeepRec lies in the state-of-the-art literature [79].

Embedding Operation Layer (Layer 1)

A few of the major problems in traditional word representations, such as one-hot vectors, are mainly losing word order, and oversize of dimensionality. To resolve the issues mentioned above, in this work, we adopted a distributed representation of word embedding. The inputs of our model are the word sequences of the abstract text in A_{text} and the title text in T_{text}. The texts A_{text} contains n sentences, and each sentence is composed of several words. Similarly, the texts T_{text} is written of several words. We trained our dataset with the Keras word embeddings technique, which can represent each word $w_i \in R^d$ as a fixed-size vector, where d is the dimension of the word vector.

In this work, we have considered the size of the dimension d as 300. Due to different sizes of abstract and title, we set L as the maximum number of words appear in both A_{text}

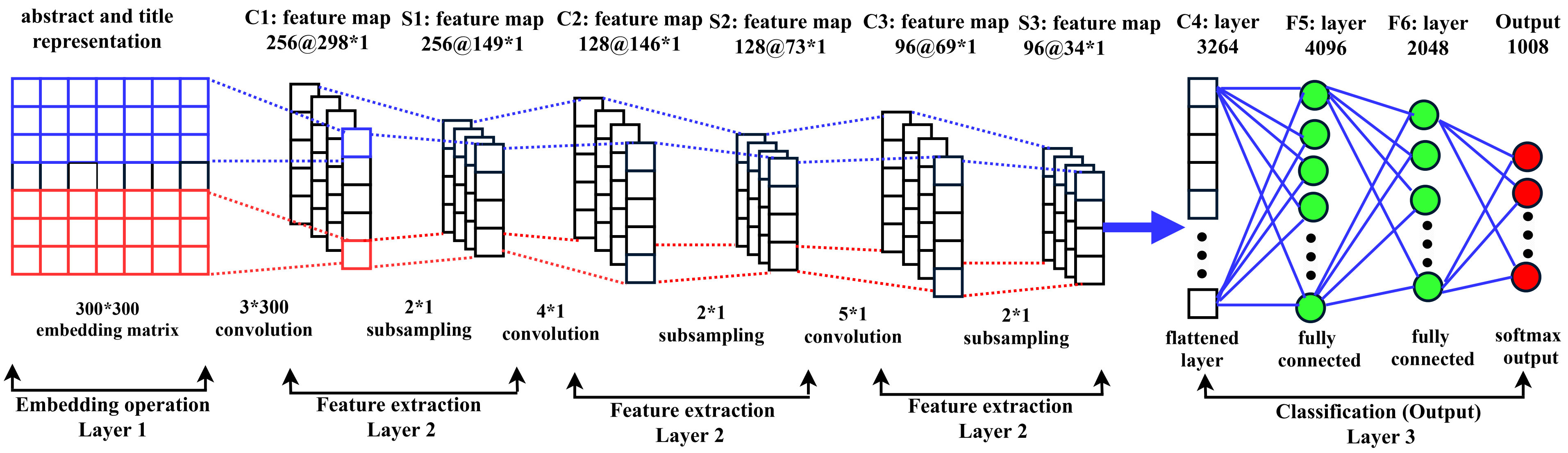


Figure 5.2: The architecture of proposed CNN model

.

and T_{text}. Let S denotes the original representation of length L (padded where necessary) appear in the texts A_{text}, and the texts of T_{text} of a given paper is represented as

$$S_{1:L} = w_1 \oplus w_2 \oplus \cdots \oplus w_L, S \in R^{L*d}, \tag{5.1}$$

Where $\oplus$ is the concatenation operator, L as the maximum number of words, which is a scalar, and $S_{1:i+j}$ refers to the vector of the concatenation of the words $w_1, w_2 \cdots w_{i+j}$.

Word vectors are initialized by zeros if they are not in the pre-trained vocabulary. We obtain the representation of an abstract is matrix S with a dimensionality of $L * d$. Thus we use the representation of an abstract matrix S to represent the text (words) appear in both A_{text} and T_{text}, respectively.

Feature Extraction Layer (Layer 2)

After getting the word embedding vectors, we need to apply the convolution layer to get high-level representations of the input texts in S. In this work, the convolutional layer is used to capture the sequence information and to reduce the dimensions of the input data. To extract more abstract and semantic features, we adopted CNN in this work.

The CNN model consists of various operations, namely convolution operation, non-linearity, and pooling. These layers are stacked to form a deep convolution neural network (CNN) and can be utilized to get a high-level representation of the input texts.

(i) **Convolution Operation**: This layer is introduced to get high-level representations of the sentences in the input text. It is mainly used to extract features from the input. Feature maps are obtained by applying convolution filters with a set of mathematical operations.

(ii) **Pooling Operation**: This layer is also called the feature extraction layer and mainly introduced to extract word-wise relevant features by applying pooling operation for further use. Pooling mainly reduces the dimensionality of the feature maps to decrease processing time.

Convolution Operation

The convolutional layer coupled with max-pooling extracts rich feature representations from each convolved word window of length l (i.e., 3 for the first convolutional layer) over the text and performs a convolution within each sliding window and the output of the k-th sliding window is computed as

$$f_k = ReLU(W_c.W_{k-l+1:k} + b_c) \tag{5.2}$$

Where ReLU is the non-linear activation function, $W_{k-l+1:k}$ denotes the concatenation of l word embeddings within the k-th window in word sequences in S (text appear in both A_{text} and T_{text}), W_c is the convolution matrix and $b_c \in R$ is the bias.

In this work, we adopted ReLU (Rectified Linear Unit) as a nonlinear activation function because it can improve the learning dynamics of the networks and significantly reduce the number of iterations required for convergence in deep networks.

We use multiple filters, and for the q-th filter, it is applied to each possible window of words in S $\{W_{1:l}, W_{2:l+1}, \cdots, W_{L-l+1:L}\}$ to produce a feature map

$$f_q = [f_{q1}, f_{q2}, \cdots, f_{q,L-l+1}] \tag{5.3}$$

with $f_q \in R^{L-l+1}$.

There can be m different number of filters which can be used to extract multiple features maps $f_1, f_2, \cdots f_m$. We get new feature representations $F \in R^{L-l+1*m}$ as the column concatenation of feature maps F=$[f_1, f_2, \cdots f_m]$. The i-th row $f^{(i)}$ of F is the new feature representation generated at position i.

So, the result of first convolution operation on S will be

$$f^{(1)} = [f_1^{(1)}, f_2^{(1)}, \cdots, f_m^{(1)}] \tag{5.4}$$

As shown in Fig. 5.2, after applying the window size of 3 in the first convolutional layer, we obtain 256 number of feature vectors (convolutional kernel) each having a dimension $298 * 1$. Similarly, after the second convolution, the output dimension will be $146 * 1$ having 128 number of features map.

Pooling Operation (Feature Maps)

The objective of the pooling operation is to successively reduce the spatial size of the representation to extract the key-features and reduce the number of dimensions in the network. We need to apply a max-pooling operation to get the most salient feature in every two-unit window for each $f_q^{(1)}$ of the input texts of S. j_q^1 is the result of the max-pooling operation.

$$j_q^{(1)} = [j_{q1}^{(1)}, j_{q2}^{(1)}, \cdots, j_{q,(L-l+1)/2}^{(1)}] \tag{5.5}$$

Where q is the q-th filter of the convolution operation.

$$j_{qi}^{(1)} = max\{f_{q,2i-1}^{(1)}, f_{q,2i}^{(1)}\} \tag{5.6}$$

Similarly, the second and third convolution and pooling layers will be executed as the first layer. Generally, the convolutional layer is an effective way for dimensionality reduction. As shown in Fig. 5.2, in the convolutional layer, 256 filters with window size 3 move on the textual representation to extract the features. As the filters move on, many sequences that capture the syntactic and semantic features are generated.

In this work, we have considered the dimension of input data is $300 * 300$, and the dimension of the first layer output data is $149 * 1$. Similarly, the dimension of the third layer is $34 * 1$. Hence the convolutional layer is an effective way for dimensionality reduction.

Classification Layer (Layer 3)

After getting the pooling results of third layer, we need to apply a flattening step to make use of fully connected layers after getting the paper-level representation. As the name flattened implies, we need to flatten our pooled feature map obtained after third pooling into a column vector. The primary reason of emplying such layer is to reshapes the pooled feature map to one single column vector to apply artificial neural network. After that, the dense layer is used to obtain new high-level representations of sentences in the review text by incorporating a hidden layer with a drop out rate to map the input vector to the desired output vector.

Given the training sample R_1 and R_2, where T is the number of possible labels and the estimated probabilities $S_j \in [0, 1]$ for each label j ∈ [1, 2, ⋯, T], the softmax is defined as:

$$S_j = \frac{e_j^z}{\sum_{k=1}^{T} e_k^z} \tag{5.7}$$

Note that in the training dataset, we only have the gold-standard original venue of each source papers. Therefore, we use the categorical cross-entropy loss to minimize the

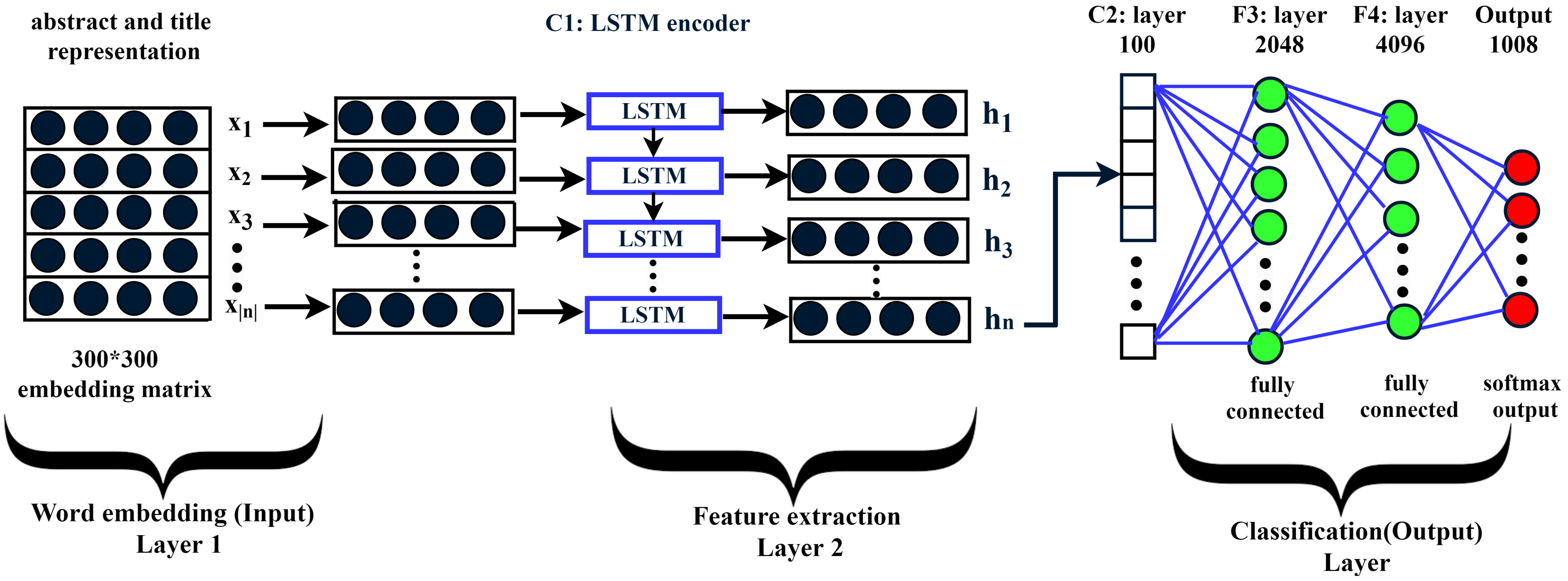


Figure 5.3: The architecture of proposed LSTM model

prediction error between the predicted venues and the gold-standard original venues:

$$L(\theta) = \sum_{j=1}^{T} Y_j log S_j \tag{5.8}$$

Where Y is the gold-standard output. We have adopted a one-hot encoding of size L, 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 Adam with minibatch to learn the model parameter θ.

Architecture of LSTM Model

The functional architecture of the LSTM model is shown in Fig. 5.3. LSTM model consists of various components that can basically capture the sequential information and also reduce the high dimensional input data into a lower-dimensional output data via Word Embedding Layer, Feature Extraction Layer, and a Classification Layer.

Word Embedding Layer (Layer 1)

We followed a similar word embedding technique, as described in Sec. 5.3.2. We adopted a distributed representation of word embedding. The inputs of our model are the word sequences of the abstract text in A_{text} and the title text in T_{text}. The texts A_{text} contains n sentences, and each sentence is composed of several words. Similarly, the texts T_{text} is written of several words. We trained our dataset with the Keras word embeddings technique, which can represent each word $w_i \in R^d$ as a fixed-size vector, where d is the dimension of the word vector.

Feature Extraction Layer (Layer 2)

Training conventional RNNs with gradient descent based backpropagation is difficult due to vanishing gradient and exploding gradients. To address this problem Long Short Term Memory (LSTM) [136] has been designed. The LSTM contains special units called memory blocks in the recurrent hidden layer [137]. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information, as displayed in Fig. 2.3 (Sec. 2.4.3).

Classification Layer (Layer 3)

After getting the final representation of the LSTM cell (h_n), we need to apply the dense layer is used to obtain new high-level representations of words in the input matrix S by incorporating a hidden layer with drop out rate to map the input vector to the desired output vector.

Algorithm 7: Stacking Ensemble Algorithm

```
Input: training data D= $\{x_i, y_i\}_{i=1}^{m}$
Output: Ensemble classifier H
Step 1: Learn base level classifiers
for $t$=1 to $T$ do:
    learn $h_t$ based on dataset $D$
end for
Step 2: Construct new dataset of predictions
for $i$=1 to $m$ do:
    $D_h$= $\{x'_i, y_i\}$, where $x'_i = \{h_1(x_i), h_2(x_i), ..., h_T(x_i)\}$
end for
Step 2: Learn a meta classifier
learn H based on $D_h$
return H
```

Given the training sample R_1 and R_2, where T is the number of possible labels and the estimated probabilities $S_j \in [0, 1]$ for each label j $\in$ [1, 2, $\cdots$, T], the softmax is defined as:

$$S_j = \frac{e^z_j}{\sum_{k=1}^{T} e^z_k} \tag{5.9}$$

Note that in the training dataset, we only have the gold-standard original venue of each source papers. Therefore, we use the categorical cross-entropy loss to minimize the prediction error between the predicted venues and the gold-standard original venues:

$$L(\theta) = \sum_{j=1}^{T} Y_j log S_j \tag{5.10}$$

Where Y is the gold-standard output. We have adopted one-hot encoding of size L, 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 Adam with minibatch to learn the model parameter θ.

5.3.3 Ensemble Learning (Block-3)

Ensemble methods are popular research directions in machine learning and pattern recognition [198]. The primary objective of ensemble methods is to combine decisions from a set of weak learning algorithms in order to enhance the accuracy and the robustness of the overall results. The generalization ability of ensemble methods is better compared to single base learners. There are statistical, computational, and representational reasons to build multiple classifier systems [199].

A stacking ensemble model is used for training RNN and CNN based architecture together in our model. In stacking, the algorithm takes the output of sub-models as inputs and attempts to learn how to combine data best to get better output results. The idea of stacking is to learn several weak learners and combine them by training a meta-model to output predictions based on multiple predictions returned by these weak models [200]. The complete algorithm is depicted in Algo. 7.

5.4 Experiments

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

5.4.1 Dataset Description

We use real-world dataset DBLP-citation-network V10 [1] (Sec. 2.7.2) to demonstrate the effectiveness of our proposed method. The tenth version contains 3,079,007 papers and 25,166,994 citations. Each paper is associated with abstract, authors, title, publishing year, venue, and references list. After removing duplicate papers, papers with missing fields, inconsistent entries in the database, journals having more than 5 numbers of papers etc., we are left with 2,234,771 papers. We also ignore non-textual content from the

[1] https://aminer.org/citation

abstracts of the papers. We have divided the dataset into three parts, including 81% of preprocessed dataset as the training set, 9% of as validation set, and the rest 10% considered as test set, respectively.

5.4.2 Evaluation Strategy

We adopt two kinds of evaluations, such as Coarse-level or offline evaluation and Fine-level or online evaluation, to measure the performances of DISCOVER against other state-of-the-art methods (Sec. 2.5).

5.4.3 Evaluation Metrics

We employed various metrics such as precision@k, nDCG@k, accuracy, MRR, and Average venue-quality (Ave-quality), to evaluate the performance of DeepRec (see Sec. 2.6).

5.4.4 Experimental Setting

In this section, we present the experimental dataset for offline and online evaluations. Then the procedure of online assessment is described in further sub-section. Due to the vast amount of data, the number of labels also increases. Due to which there is a difficulty to be trained and learned the proposed model.

In order to resolve these above-mentioned issues along with other operational constraints (resource and time), the experiment is performed in two stages to demonstrates the efficacy of DeepRec.

(i) Preparation of dataset for offline evaluation

(ii) Preparation of dataset for online evaluation

Preparation of Dataset for Offine Evaluation

Initially, we identified only those venues whose number of papers were less than 500 and removed such venues having a number of papers more than 500. Dataset contains total of $342,258$ research papers after preprocessing and is split into three parts training set= 81%, validation set=9%, and test set=10%. The complete statistics of the overall dataset is depicted in Table 5.1.

Table 5.1: Statistics of offline dataset

Types	Training Dataset	Validation Dataset	Testing Dataset
No. of papers	277,229	30,803	34,226
No. of venues	2,208	2,208	2,208

Seed papers are chosen from the testing dataset, keeping in mind the cold-start issues for new venues and new researchers. We consider 3 categories of venues and 3 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) [31, 74] on the following six categories.

(i) Category 1 : $5 \leq v_c < 20$

(ii) Category 2 : $20 \leq v_c < 50$

(iii) Category 3 : $50 \leq v_c$

(iv) Category 4 : $5 \leq p_c < 20$

(v) Category 5 : $20 \leq p_c < 50$

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

It is ensured that each category is well represented in the seed papers.

Preparation of Dataset for Onine Evaluation

Initially we remove such venues having less than 500 papers to prepare the dataset for online evaluation. The final dataset was left with $1,892,513$ paper and $1,008$ venues. We have divided the dataset into three parts including 81% of preprocessed dataset as the training set, 9% of as validation set, and the rest 10% considered as test set respectively. The venue-wise papers statistics and complete statistics of the overall dataset are depicted in Table 5.2 and Table 5.3 respectively. Due to operational constraints (difficult to incorporate user study for all testing papers), only 20 sub-domains of computer science were selected as a testing dataset in our experiment. A total of 160 seed papers (8 from each sub-domains) are chosen manually from 20 sub-domains: information retrieval (IR), image processing (IP), security (SC), wireless sensor network (WSN), machine learning (ML),

Table 5.2: Statistics of online evaluation dataset

Range of papers	Number of papers	Number of journals
$5 \leq X \leq 100$	240,667	1,009
$100 \leq X \leq 400$	229,547	1,027
$400 \leq X \leq 2000$	825,940	934
$2000 \leq X \leq 10{,}000$	915,876	232
$X \geq 10{,}000$	22,741	14
Max. class size	121,328	1
Min. class size	5	67
Avg. class size	694	
All	22,34,771	3,216

Table 5.3: Statistics of online dataset

Types	Training Dataset	Validation Dataset	Testing Dataset
No. of papers	1,532,935	170,327	189,251
No. of venues	1,008	1,008	1,008

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).

Procedure of Online Evaluation

For this evaluation, we did not have the ready annotation, but we need one. The annotation or relevance assessment is collected from the volunteers through crowdsourcing in the best effort basis. There are 85 researchers with expertise in the mentioned subdomains are provided with input and output of our recommender system where for each paper, 15 venues are recommended. Out of 85 researchers, 23 evaluated 3 papers each, 29 researchers evaluated 2 each, and the rest 33 were evaluated by 33 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 research students and research assistants who were pursuing Ph.D. with bachelor's or masters' degrees in science or technology. The experts or researchers were so chosen that their active areas of research perfectly match with the topics of seed papers. Among 85 researchers, there were 19 professors, 18 associate professors, 26 assistant professors, 14 senior research students, and the remaining 8 were research assistants.

All experts were from a reputed institution like Indian Institute of Technology Kharagpur, Indian Institute of Technology Roorkee, Indian Institute of Technology Kanpur, Indian Institute of Technology (BHU) Varanasi, Central University Hyderabad, Manipal University, and Banaras Hindu University (BHU). The age range of all professors are in the range of [48-55], age range of associate professors are in between [43-47], assistant professors are having an age of [36-41], senior research students are in the age range of [28-31], and remaining research assistants are having an age range of [29-33]. The overall gender distribution of male and female experts were 53 and 32, respectively.

The experts check the titles, abstracts, authors, year of publication, and recommended venues of the papers. An expert assigns an appropriate relevance value (r) to each recommended venue as she deems the quality of the match between the scope of the recommended venue and the topic of the seed paper as below.

$$\text{Relevance } (r) = \begin{cases} 2 & \text{perfectly matching} \\ 1 & \text{partial matching} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

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

To comprehensively evaluate our proposed method and more specifically, to address the broad research questions (RQs) discussed in Sec. 1.5, we prefer to examine the following sub-queries (SQs):

SQ1: How effective is DeepRec in comparison to other state-of-the-art methods?

SQ2: How is the quality of venues recommended by DeepRec as compared to state-of-the-art methods?

SQ3: How does DeepRec handle cold-start issues and other issues like data sparsity, diversity, and stability?

5.4.5 Parameter Tuning and Optimization

In this section, we outline various experimental parameter settings of DeepRec. DeepRec has a few essential parameters during its process pipeline as follows.

(i) Parameter tuning in CNN

(ii) Parameter tuning in LSTM

(iii) Parameter tuning in DeepRec

Parameter Tuning in CNN

The same dataset (DBLP) was used to train CNN as in the LSTM model. Dataset was split into three parts training set= 81%, validation set= 9%, and test set=10%. We trained data using CNN model with three convolution layers having filter sizes of (3,4, and 5), respectively, three max-pooling layers with the filter size of 2 and stride of 2, dense layers of size 4,096 and 2,048 with the Dropout rate of 0.2. The dense layer is connected to the softmax layer of size 1,008, i.e., the number of total venues.

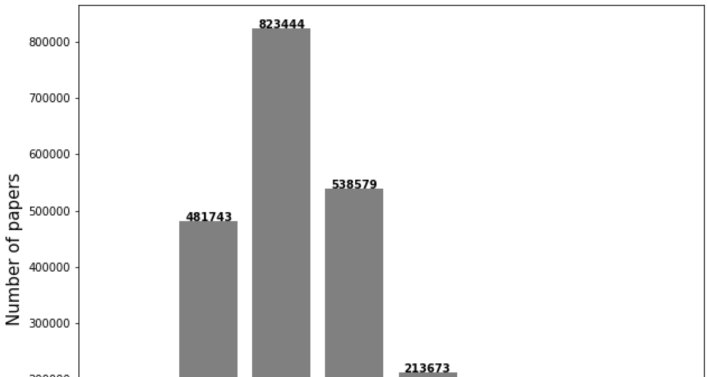


Figure 5.4: Statistics of word count of abstracts

.

The model is trained on categorical cross-entropy loss function, Adam optimizer, and metrics as accuracy, top-3, top-5, top-10, and top-15 accuracy. The title and abstract of paper are concatenated as (text) to pass through the embedding layer of Keras. Size of text is fixed at L=300 (95% of texts have length below 300), each word is embedded to d=300 length vectors. Text having a length less than 300 words are padded, and text has more than 300 words are truncated. We have taken the maximum size of the abstract as 300 because it is the maximum number of the words that most of the abstracts contain, as depicted in Fig. 5.4.

Parameter Tuning in LSTM

Dataset contains total of 1,892,513 research papers after preprocessing and is split into three parts training set= 81%, validation set=9%, and test set=10%. For the Recurrent Neural Network based model, we have used a single LSTM layer and four dense layers with Dropout (dropout rate = 0.3) and BatchNormalization. The final layer is the softmax layer having 1,008 labels representing a total number of Journals for training.

The model is trained on categorical cross-entropy loss function, Adam optimizer, and metrics as accuracy, top-3-accuracy, top-5-accuracy, top-10-accuracy, and top-15-accuracy. The title and abstract of paper are concatenated as (text) to pass through the embedding layer of Keras. Size of text is fixed at L=300 (95% of texts have length below 300), each word is embedded to d=300 length vectors. Text having a length less than 300 words are padded, and text has more than 300 words are truncated.

Parameter Tuning in DeepRec

In our experiment, we adopted top-N accuracy to measure overall classification performance, which is defined as the expected label for top N predicted classes where N={1, 3, 5, 10, 15}. DBLP dataset contains a total of $1,892,513$ research papers after preprocessing and is split into three parts training set= 81%, validation set=9%, and test set=10%. For the training Stacked Ensemble Model, we combined CNN and LSTM architectures, which are trained separately. Outputs from both the models are concatenated together and passed through a Dense layer of size $2,048$ with a Dropout rate of 0.2; this output is then batch normalized and passed to a softmax layer.

The model is trained with a categorical cross-entropy loss function and Adam Optimizer. Title and abstract of paper are concatenated as (text) and passed to embedding layers of CNN and LSTM. The model is trained for 10 epochs, and it is validated after every epoch with a validation set of 10% of training data. A batch size of 1, 024 was used during training. Early stopping with a delta of 0.00001 and patience of 2 was applied on validation loss for a call back during training. Once the model is trained, we evaluate our model on test data.

5.4.6 Baseline Methods

To measure the effectiveness of DeepRec we, compare our results with various state-of-the-art methods such as FB, CF, CN, CBF, RWR, PRS, PVR, PRS, and PAVE (Sec. 2.8.1).

In addition to the above state-of-the-art methods, we also validated our proposed model DeepRec against a few more deep learnig based methods, including CNN, LSTM, Bi-LSTM, CNN+Bi-LSTM, RNN+CNN. Among these discussed methods CF and PVR are based on collaborative filtering approach, PAVE and RWR are based on random walk with restart algorithm exploiting co-authorship networks, CN and FB are based on co-authorship network, CBF, PRS are based on content-based filtering method, and LSTM, CNN, RNN, Bi-LSTM are based on deep learning methods.

5.5 Results and Discussion

In this section, the performance of DeepRec against the existing state-of-the-art methods is reported. For clarity and easy understanding, we provide the results and discussion in two steps (offline and online) as given below. During the assessment, best results and the second-best performer are marked by the 'bolf-face' and '+' symbol in each position.

Table 5.4: Accuracy and MRR results for CNN and other approaches

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.0396	0.0784	0.1005	0.1397	0.1685	0.0279
CF	0.0743	0.0994	0.1317	0.1596	0.2016	0.0316
CN	0.0957	0.1256	0.1648	0.2006	0.2325	0.0412
CBF	0.1362	0.1673	0.1988	0.2005	0.2648	0.0414
RWR	0.1718	0.2005	0.2386	0.2719	0.2867	0.0579
PVR	0.1894	0.2037	0.2124	0.2756	0.2948	0.0671
PRS	0.2267	0.2652	0.2884	0.3659^{+}	**0.4732**	0.1059^{+}
PAVE	0.2274^{+}	0.2695^{+}	0.2997^{+}	0.3244	0.3991	0.0878
CNN	**0.2593**	**0.3007**	**0.3906**	**0.4275**	0.4529^{+}	**0.1287**

Best results are highlighted in bold, and 2ND best are marked by ('+')

5.5.1 Offline Evaluation of CNN Model

The complete results of accuracy and MRR are depicted in Table 5.4 at position 3, 6, 9, 12, and 15 respectively. We can see that the CNN model shows a consistent accuracy over all other state-of-the-art methods. More than 25% time (Acc@3=0.2593), it could predict the original venue of the seed paper within top 3 recommendations. Even if the original venues of seed papers are having less number of papers (less than 500 papers), but still more than 45% time (Acc@15=0.4529), CNN model can predict it within top 15 recommendations. As far as accuracy concerns, PSR performs the second-best and best (Acc@15=0.4732) at positions 15 and 12 respectively. FB method exhibits the worst performance with an accuracy of 0.1685 while recommending 15 recommendation.

Table 5.5: Precision results for CNN and other approaches

Methods	P@3	P@6	P@9	P@12	P@15
FB	0.6052	0.5899	0.5753	0.5522	0.5798
CF	0.6125	0.6354	0.6357	0.6466	0.6354
CN	0.6801	0.6681	0.6715	0.6690	0.6592
CBF	0.7125	0.7121	0.7116	0.7016	0.7009
RWR	0.7460	0.7226	0.7159	0.7179	0.7208
PVR	0.8159	0.7848	0.7529	0.7618	0.7601
PRS	0.8079	0.8380+	0.7654	0.7609	0.7639
PAVE	0.8219+	0.8049	0.8395+	**0.8393**	**0.8412**
CNN	**0.8574**	**0.8579**	**0.8564**	0.8271+	0.8376+

Best results are highlighted in bold, and 2ND best are marked by ('+')

During the evaluation of MRR results, we can see that the CNN model performs excellent behavior and shows a MRR result of 0.1287. The proposed approach could predict the original venue at early recommendations as compared to all other methods. In the case of MRR also, the least performance of MRR 0.0279 is shown by the FB method. As far as MRR concerns, PSR performs the second-best among all other state-of-the-art methods.

5.5.2 Online Evaluation of CNN Model

In this section, we examine the performance of the CNN model against other state-of-the-art methods. The evaluation metrics, including precision, nDCG, and average venue quality (H5-Index) are taken into consideration during this evaluation.

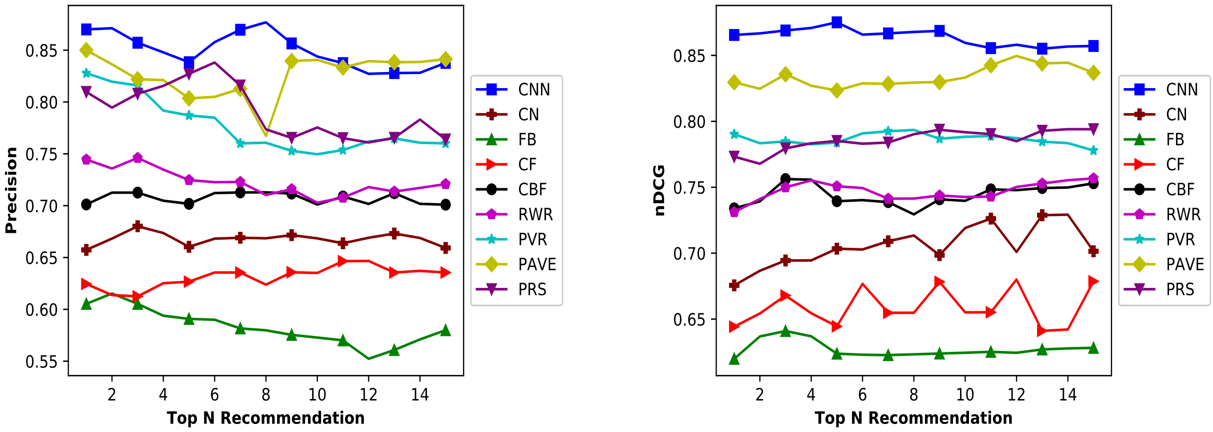
Table 5.6: nDCG results for CNN and other approaches

Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
FB	0.6409	0.6229	0.6238	0.6244	0.6281
CF	0.6678	0.6767	0.6782	0.6800	0.6786
CN	0.6944	0.7028	0.6985	0.7009	0.7014
CBF	0.7562	0.7402	0.7408	0.7478	0.7530
RWR	0.7499	0.7494	0.7437	0.7502	0.7562
PVR	0.7847	0.7908	0.7867	0.7872	0.7778
PRS	0.7794	0.7830	0.7936	0.7849	0.7940
PAVE	0.8356^{+}	0.8288^{+}	0.8298^{+}	0.8446^{+}	0.8368^{+}
CNN	**0.8689**	**0.8657**	**0.8686**	**0.8581**	**0.8571**

Best results are highlighted in bold, and 2ND best are marked by ('+')

Precision@k

The results evaluation of precision@k as shown in Table 5.5 indicates the significance of the CNN model in terms of precision@k over all other standard approaches. It is seen that the CNN model achieves a precision of 0.8768 while recommending the initial 8 venues as shown in Fig. 5.5a. But later on, the precision keeps on decreasing and reaching the value of 0.8281 at position 14.



(a)

(b)

Figure 5.5: (a) Precision analysis for CNN (b) nDCG analysis for CNN

The proposed CNN model exhibits the highest precision of 0.8376 after recommending 15 recommendations. The least value of precision 0.8271 occurs at a position 12. PAVE method shows a higher performance than the CNN model at position 12, 13, 14, and 15, respectively. The method PSR shows the second-best performance at position 6 with a precision of 0.8380. The worst performance with a precision of 0.5798 at position 15 is shown by the FB method among all other standard methods.

nDCG@k

The overall nDCG@k of all methods are shown in Table 5.6. During nDCG@k evaluation, it is observable that the proposed CNN model shows a significant improvement of nDCG over all other state-of-the-art methods. The CNN model performs an upward trend and reaches the highest nDCG of 0.8751 at position 5, and afterward again, it shows a downward trend and finally shows a nDCG of 0.8571 at position 15 as depicted in Fig. 5.5b.

It is seen that PAVE method shows the second-best performance than other state-of-the-art methods. The worst performance with a nDCG of 0.6281 is shown by the FB method after recommending 15 venues among all other standard methods.

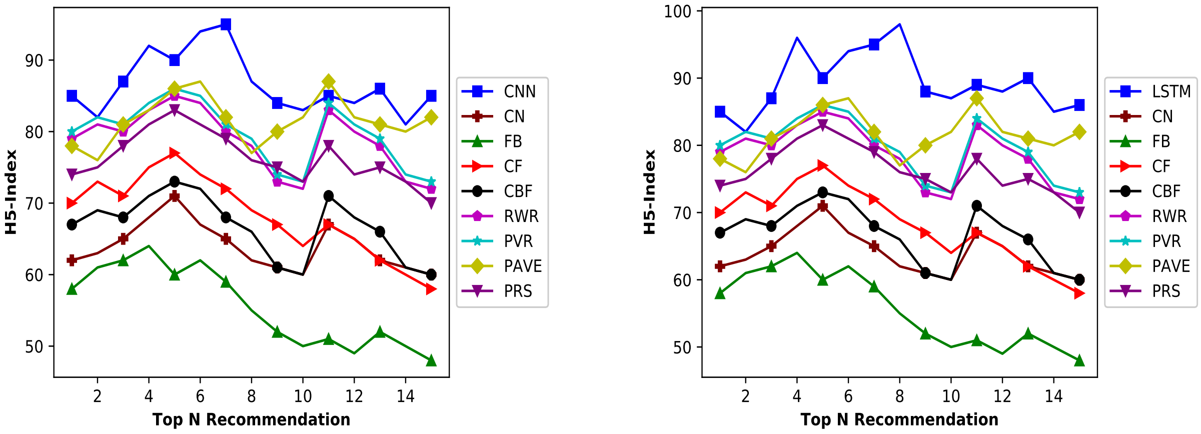
Average venue Quality (H5-Index) Analysis

We considered the dataset prepared for online evaluation (journals having atleast 500 papers) to measure the venue quality. Due to this biasing, chances of capturing better quality journals in terms of H5-Index are relatively higher. The CNN model outperforms other methods in terms of average H5-Index of recommended venues, as displayed in Fig. 5.6a. While evaluating average venue quality, the CNN model performs an upward trend from the beginning and shows an overall average H5-Index of 87.

The top-quality venues recommended by CNN model are at position 7 with the highest H5-Index of 95. Then it shows a downward trend and reaches an H5-Index of value 85 at position 15. The lowest quality of venues recommended by the FB method with an average H5-Index of 56. Similarly, the second-highest quality venues recommended by PAVE model with an average H5-Index of 81.

5.5.3 Offline Evaluation of LSTM Model

During the evaluation of accuracy and MRR, we can see from Table 5.7 that LSTM model shows a consistent accuracy over all other standard approaches. More than 39% time it



(a)

(b)

Figure 5.6: (a) Venue quality of CNN (b) Venue quality of LSTM

Table 5.7: Accuracy and MRR results for LSTM and other approaches

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.0396	0.0784	0.1005	0.1397	0.1685	0.0279
CF	0.0743	0.0994	0.1317	0.1596	0.2016	0.0316
CN	0.0957	0.1256	0.1648	0.2006	0.2325	0.0412
CBF	0.1362	0.1673	0.1988	0.2005	0.2648	0.0414
RWR	0.1718	0.2005	0.2386	0.2719	0.2867	0.0579
PVR	0.1894	0.2037	0.2124	0.2756	0.2948	0.0671
PRS	0.2267	0.2652	0.2884	0.3659^{+}	0.4732^{+}	0.1059^{+}
PAVE	0.2274^{+}	0.2695^{+}	0.2997^{+}	0.3244	0.3991	0.0878
LSTM	**0.3932**	**0.4963**	**0.5865**	**0.6294**	**0.6549**	**0.1593**

Best results are highlighted in bold, and 2ND best are marked by (‘+’)

could predict the original venue of the seed paper within the top 3 recommendations. Initially, the LSTM model shows an accuracy of 0.4963 at position 6. Then slowly it shows an upward trend and exhibits an excellent performance with an accuracy of 0.6549 at position 15.

In the case of MRR evaluation, it is visible that, LSTM model shows an excellent performance over other standard approaches. We have experimented on DBLP dataset and observed that the LSTM model exhibits the overall MRR of 0.1593. The second-best performance is shown by the PRS model with an MRR of 0.1059. The CNN model could predict the original venue at early recommendations as compared to all other methods. In the case of MRR also, the least performance is exhibited by the FB method.

Table 5.8: Precision results for LSTM and other approaches

Methods	P@3	P@6	P@9	P@12	P@15
FB	0.6052	0.5899	0.5753	0.5522	0.5798
CF	0.6125	0.6354	0.6357	0.6466	0.6354
CN	0.6801	0.6681	0.6715	0.6690	0.6592
CBF	0.7125	0.7121	0.7116	0.7016	0.7009
RWR	0.7460	0.7226	0.7159	0.7179	0.7208
PVR	0.8159	0.7848	0.7529	0.7618	0.7601
PRS	0.8079	0.8380^{+}	0.7654	0.7609	0.7639
PAVE	0.8219^{+}	0.8049	0.8395^{+}	0.8393^{+}	0.8412^{+}
LSTM	**0.8803**	**0.8867**	**0.8786**	**0.8654**	**0.8537**

Best results are highlighted in bold, and 2ND best are marked by ('+')

Table 5.9: nDCG results for LSTM and other approaches

Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
FB	0.6409	0.6229	0.6238	0.6244	0.6281
CF	0.6678	0.6767	0.6782	0.6800	0.6786
CN	0.6944	0.7028	0.6985	0.7009	0.7014
CBF	0.7562	0.7402	0.7408	0.7478	0.7530
RWR	0.7499	0.7494	0.7437	0.7502	0.7562
PVR	0.7847	0.7908	0.7867	0.7872	0.7778
PRS	0.7794	0.7830	0.7936	0.7849	0.7940
PAVE	0.8356^{+}	0.8288^{+}	0.8298^{+}	**0.8496**	**0.8368**
LSTM	**0.8879**	**0.8705**	**0.8554**	0.8492^{+}	0.8356^{+}

Best results are highlighted in bold, and 2ND best are marked by ('+')

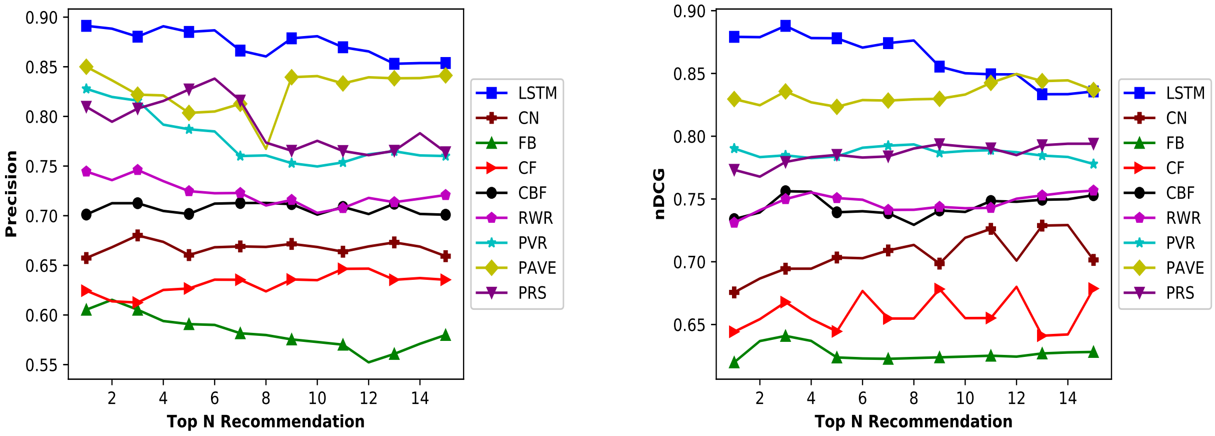
5.5.4 Online Evaluation of LSTM Model

In this section, we examine the performance of the LSTM model against other state-of-the-art methods. The evaluation metrics, including precision, nDCG, and average venue quality (H5-Index) are taken into consideration during this evaluation.

Precision@k

The overall results of precision are shown in Table 5.8. We can see the significance of LSTM model in terms of precision over all other standard approaches. Initially, the proposed LSTM exhibits a precision of 0.8803 at position 3, and after that, it slightly behaves a downward trend and shows a precision of 0.8786 at position 9 as depicted in Fig. 5.7a.

The proposed model LSTM exhibits the highest precision of 0.8537 after recommend-



(a)

(b)

Figure 5.7: (a) Precision of LSTM (b) nDCG of LSTM

Table 5.10: Accuracy and MRR results of DeepRec and other compared approaches

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.0396	0.0784	0.1005	0.1397	0.1685	0.0279
CF	0.0743	0.0994	0.1317	0.1596	0.2016	0.0316
CN	0.0957	0.1256	0.1648	0.2006	0.2325	0.0412
CBF	0.1362	0.1673	0.1988	0.2005	0.2648	0.0414
RWR	0.1718	0.2005	0.2386	0.2719	0.2867	0.0579
PVR	0.1894	0.2037	0.2124	0.2756	0.2948	0.0671
PRS	0.2267	0.2652	0.2884	0.3659^{+}	0.4732^{+}	0.1059
PAVE	0.2274^{+}	0.2695^{+}	0.2997^{+}	0.3244	0.3991	0.0878^{+}
CNN	0.2593	0.3007	0.3906	0.4275^{+}	0.4529	0.1287
LSTM	0.3932	0.4963	0.5865	0.6294^{+}	0.6549	0.1593
Bi-LSTM	0.4076	0.5037	0.5479	0.6173	0.6482	0.1589
CNN+Bi-LSTM	0.4182	0.5096	0.5845	0.6362	0.6547	0.1873
RNN+CNN	0.3887	0.4779	0.4867	0.5343	0.5532	0.1562
DeepRec (Stacking Ensemble)	0.4393*	0.5237*	0.6393*	0.6525*	0.6992*	0.2195*

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

ing 15 recommendations. Similarly, the PAVE method performs the second-best among all other standard approaches except position 6. The method PSR shows an excellent performance with higher precision than PAVE at position 6. The worst performance among all methods is shown by the FB method.

nDCG@k

The overall results of nDCG@k of all methods are shown in Table 5.9. It is clearly seen that the proposed LSTM shows a significant improvement of nDCG over all other state-of-the-art methods. Initially, the LSTM model achieves a nDCG of 0.8789 at position 2. Then the LSTM model performs a downward trend and reaches a nDCG of 0.8356 at position 15 as displayed in Fig. 5.7b.

It is seen that the overall nDCG results of LSTM model are consistent until position 9. The PAVE model shows an excellent performance with higher nDCG than the LSTM model at position 12, 13, 14, and 15, respectively. The LSTM shows the highest nDCG of 0.8879 at position 3. The second-best performance is shown by the PAVE model. The FB model shows the worst performance over all other standard approaches.

Average Venue Quality (H5-Index) Analysis

We investigate the performance of venues quality recommended by LSTM model as compared to other existing approaches. LSTM model outperforms other methods in terms of average H5-Index of recommended venues as displayed in Fig. 5.6b. Overall, the average H5-Index of venues recommended by LSTM model is 89.

The top-quality venues recommended by LSTM model are at position 8 with the highest H5-Index of 98. Then it shows a downward trend and reaches an H5-Index of value 86 at position 15. The lowest quality of venues with an H5-index of 82 is recommended by the LSTM model at position 2.

5.5.5 Offline Evaluation of Ensembled Model: DeepRec

In this section, we examine the performance of the ensembled model DeepRec in terms of evaluation metrics such as accuracy, and MRR. We also conduct paired-samples t-test on overall accuracy and MRR between DeepRec and the second-best performers. Only p values less than 0.05 were considered statistically significant at 5% level of significance (α =0.05). During the assessment of DeepRec, statistically significant results and the second-best performer are marked by the '*' and '+' symbols in each position.

The complete results of accuracy and MRR are depicted in Table 5.10 while evaluating it for the journals having less than 500 papers. It is evident that proposed DeepRec shows a consistent performance over all other standard approaches. More than 43% time, it could predict the original venue of the seed paper within the top 3 recommendations. Initially, the DeepRec model shows an accuracy of 0.4393 at position 3. Then slowly it shows an upward trend and exhibits an excellent performance with an accuracy of 0.6992 at position 15. The PAVE method performs the second-best among all other standard approaches at positions 3, 6 and 9 respectively. We have also investigated the overall The complete results of accuracy and MRR are depicted in Table 5.11 while evaluating it for the journals having more than 500 papers. More than 83% time it could predict the original venue of the seed paper within the top 15 recommendations. The worst performance among all methods is shown by the FB method.

Table 5.11: Accuracy and MRR results of DeepRec and other compared approaches

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.0555	0.0972	0.1250	0.1666	0.1944	0.0338
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	0.1527	0.1805	0.2083	0.2361	0.2916	0.0648
RWR	0.1944	0.2222	0.2500	0.2916	0.3194	0.0775
PVR	0.2083	0.2361	0.2368	0.3194	0.3472	0.0863
PRS	0.2497	0.2877	0.3265^{+}	0.3987^{+}	0.5467^{+}	0.1356^{+}
PAVE	0.2500^{+}	0.2916^{+}	0.3055	0.3611	0.4305	0.0906
CNN	0.4379	0.5264	0.7006	0.7208	0.7273	0.1287
LSTM	0.4563	0.5647	0.6752	0.7342	0.7501	0.1385
Bi-LSTM	0.4547	0.5659	0.6377	0.6984	0.7482	0.1294
CNN+Bi-LSTM	0.4874	0.5651	0.6659	0.7345	0.7842	0.2203
RNN+CNN	0.4231	0.4474	0.4645	0.4981	0.5042	0.2046
DeepRec (Stacking Ensemble)	0.4982*	0.5994*	0.6993*	0.7871*	0.8369*	0.2894*

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

Similarly, during the evaluation of MRR, we can see that the proposed model DeepRec outperforms all other state-of-the-art methods and shows excellent behavior with an MRR result of 0.2195. The proposed approach could predict the original venue at early recommendations as compared to all other methods. The second-best performance is exhibited by the PRS method, whereas the FB method performs the worst among all different standard approaches.

5.5.6 Online Evaluation of Ensembled Model: DeepRec

In this section, we examine the performance of the ensembled model DeepRec against other state-of-the-art methods. The evaluation metrics, including precision, nDCG, and average venue quality (H5-Index) are taken into consideration during this evaluation. We also conduct paired-samples t-test on overall precision and nDCG between DeepRec and the second-best performers. Only p values less than 0.05 were considered statistically significant at 5% level of significance (α =0.05). During the assessment of DeepRec, statistically significant results and the second-best performer are marked by the '*' and '+' symbols in each position.

Table 5.12: Precision results for proposed DeepRec and other compared approaches

Methods	P@3	P@6	P@9	P@12	P@15
FB	0.6052	0.5899	0.5753	0.5522	0.5798
CF	0.6125	0.6354	0.6357	0.6466	0.6354
CN	0.6801	0.6681	0.6715	0.6690	0.6592
CBF	0.7125	0.7121	0.7116	0.7016	0.7009
RWR	0.7460	0.7226	0.7159	0.7179	0.7208
PVR	0.8159	0.7848	0.7529	0.7618	0.7601
PRS	0.8079	0.8380$^+$	0.7654	0.7609	0.7639
PAVE	0.8219$^+$	0.8049	0.8395$^+$	0.8393$^+$	0.8412$^+$
CNN	0.8574	0.8579	0.8564	0.8271	0.8376
LSTM	0.8803	0.8867	0.8786	0.8654	0.8537
Bi-LSTM	0.8748	0.8793	0.8694	0.8563	0.8469
CNN+Bi-LSTM	0.8792	0.8673	0.8685	0.8492	0.8338
RNN+CNN	0.8547	0.8495	0.8437	0.8347	0.8297
DeepRec (Stacking Ensemble)	0.9146*	0.9210*	0.9130*	0.9037*	0.9142*

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

Precision@k

The overall results precision evaluation are shown in Table 5.12. We can see the significance of DeepRec in terms of precision over all other standard approaches. Initially, the proposed DeepRec exhibits a precision of 0.9227 at position 2, and after that, it slightly behaves a downward trend and shows a precision of 0.9040 at position 7 and finally shows a precision of 0.9142 at position 15 as displayed in Fig. 5.8a.

Proposed model DeepRec exhibits the highest precision of 0.9255 at position 1. It

Table 5.13: nDCG results for proposed DeepRec and other approaches

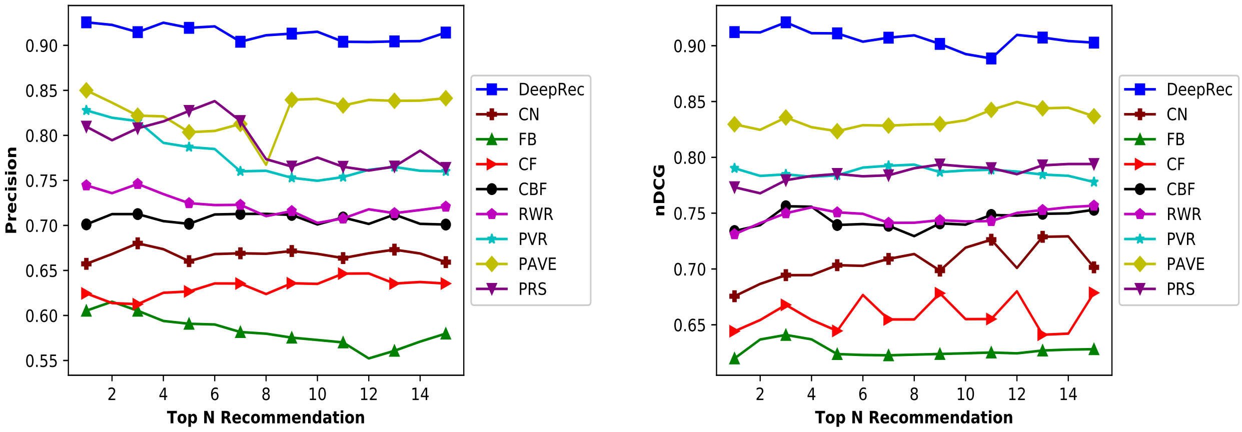
Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
FB	0.6409	0.6229	0.6238	0.6244	0.6281
CF	0.6678	0.6767	0.6782	0.6800	0.6786
CN	0.6944	0.7028	0.6985	0.7009	0.7014
CBF	0.7562	0.7402	0.7408	0.7478	0.7530
RWR	0.7499	0.7494	0.7437	0.7502	0.7562
PVR	0.7847	0.7908	0.7867	0.7872	0.7778
PRS	0.7794	0.7830	0.7936	0.7849	0.7940
PAVE	0.8356^+	0.8288^+	0.8298^+	0.8496^+	0.8368^+
CNN	0.8689	0.8657	0.8686	0.8581	0.8571
LSTM	0.8879	0.8705	0.8554	0.8492	0.8356
Bi-LSTM	0.8793	0.8691	0.8566	0.8398	0.8295
CNN+Bi-LSTM	0.8642	0.8632	0.8584	0.8473	0.8344
RNN+CNN	0.8547	0.8495	0.8437	0.8347	0.8297
DeepRec (Stacking Ensemble)	0.9209*	0.9036*	0.9016*	0.9096*	0.9027*

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

shows a lower precision of 0.9037 at a position 12. Similarly, PAVE method performs the second-best among all other state-of-the-art methods except a few positions 5,6, 7, and 8, respectively. The PRS method exhibits slightly higher precision than PAVE method at those positions. The worst performance among all methods is shown by the FB method.

nDCG@k

The overall results of nDCG@k of all methods are shown in Table 5.13. It is clearly visible that the proposed DeepRec shows a significant improvement of nDCG over all other state-of-the-art methods. During the initial recommendations, the proposed ensemble model DeepRec performs a higher nDCG of 0.9209 at position 3. Then it shows a downward trend and reaches a nDCG of 0.8885 at position 11, and afterward, it shows an upward trend and reaches a nDCG of 0.9027 at position 15. It is clearly shown in Fig. 5.8b that the overall nDCG results of DeepRec are consistent and shows the highest nDCG of 0.9209 at position 3. The PAVE model demonstrates the second-best performance. The FB model shows the worst performance over all other standard approaches.



(a)

(b)

Figure 5.8: (a) Precision of DeepRec (b) nDCG of DeepRec

Average Venue Quality (H5-Index) Analysis

We investigate the performance of venue quality recommended by DeepRec as compared to other existing approaches, as depicted in Fig. 5.9. Overall, the average H5-Index of venues recommended by DeepRec model is 93. The top-quality venues recommended by DeepRec model is at position 4 with the highest H5-Index of 102. Then it shows a downward trend and reaches an H5-Index of value 93 at position 15. The lowest quality of venues with an H5-index of 86 is recommended by the DeepRec model at position 2.

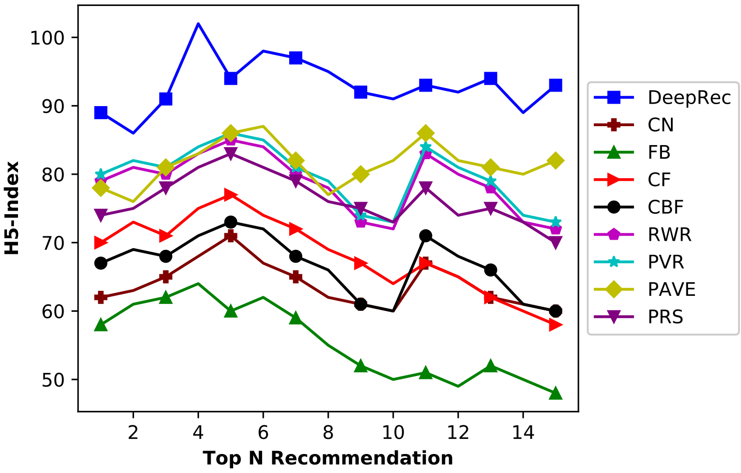


Figure 5.9: Venue quality of DeepRec and other approaches

.

5.5.7 Study of Proposed Approach

The main findings concerning our SQs as introduced in Sec. 5.4.4 are summarized below:

SQ1: How Effective is DeepRec in Comparison to Other State-Of-The-Art Methods?

We investigated the overall results of precision@k, nDCG@k, accuracy, and MRR of proposed ensembled model DeepRec and all other state-of-the-art techniques. It demonstrates the best execution while assessing promising outcomes in higher values of precision@k and nDCG@k separately. Additionally, the performance of accuracy and MRR demonstrates that the proposed approach; DeepRec results are measurably significant over all other state-of-the-art techniques. The outcomes are appeared in Table 5.10, Table 5.11, Table 5.12, and Table 5.13 respectively.

Table 5.14: MRR results for DeepRec over new venue and new researcher

Methods	2<= v_c <8	8<= v_c <15	15<= v_c	2<= p_c <8	8<= p_c <15	15<= p_c
FB	0.0437	0.0591	0.0683	0.0565	0.0677	0.0769
CF	0.0473	0.0637	0.0725	0.0683	0.0746	0.0828
CN	0.0526	0.0773	0.0866	0.0739	0.0877	0.0914
CBF	0.0668	0.0848	0.1132	0.0894	0.0917	0.0995
RWR	0.0793	0.0849	0.0853	0.0854	0.0861	0.0864
PVR	0.0798	0.0879	0.0851	0.0853	0.0893	0.0847
PRS	0.0972	0.0895	0.0949	0.0858	0.0903	0.0885
PAVE	0.0977^{+}	0.1014^{+}	0.1298^{+}	0.1016^{+}	0.1039^{+}	0.1073^{+}
DeepRec	0.2076*	0.2274*	0.2217*	0.2154*	0.2141*	0.2251*

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

SQ2: How is The Quality of Venues Recommended by DeepRec as Compared to State-Of-The-Art Methods?

The venues recommended by DeepRec are of high quality when contrasted with other cutting edge techniques as portrayed in Fig. 5.9. The average H5-index of DeepRec demonstrates a most elevated average estimation of 93 after recommending 15 venues. The most elevated H5-index recommended by DeepRec is 102, and the least is 86, whereas the most noteworthy H5-index suggested by PAVE is 87, and the least is 76. It shows the highest H5-Index of 102 at position 4. At position 2, it shows the lowest H5-Index of 86. PAVE recommends venues having the second best H5-index. The least quality of recommendation performed by the FB model.

SQ3: How Does DeepRec Handle Cold-start Issues and Other Issues Like Data Sparsity, Diversity, and Stability

(i) **Cold-start Issues**: To specifically address "cold-start" issues like a new researcher and new venue, we integrate extracted high-level features from abstract and title with CNN model and LSTM model, respectively into the proposed model DeepRec. We applied CNN in order to extract latent factors from the content information, which then directly integrated into the recommendation process to deal with the cold-start issues. Similarly, we have also used LSTM to extract low dimensional latent factors of high dimensional input (seed paper).

We investigated the performance of DeepRec while assessing for new researchers and new venues associated inputs (seed papers). The examination in Table 5.14 reflects that, regardless of whether the seed paper related to the new researcher and new venue, DeepRec could anticipate the original venue at an early stage of recommendations. It does not require past publication records or co-authorship networks for the recommendations. It considers only the current area of interest along with the title and abstract as inputs to recommend the same. DeepRec works irrespective of researchers' past publication records, rather only focuses on the work at hand. DeepRec does not have data sparsity issues, as mentioned in Table 5.15.

(ii) **Data Sparsity**: To explicitly address data sparsity issue, both significance and relevance parameters are taken into consideration. We have transformed the high dimensional and sparse embedding matrix into a lower-dimensional and dense set using deep learning techniques including CNN and LSTM, respectively. CNN is specifically designed to process temporal, latent contextual aspects of high dimensional and sparse input. Due to this efficiency of extracting hidden contextual features that are relevant makes the deep learning approach highly preferable. We use stacked generalized ensemble leaning of both CNN and LSTM model in order to capture the quality of essentialness, relevance and to extract low dimensional latent factors of high dimensional input aiming to cope with data sparsity issue.

Table 5.15: Cold-start and other issues available

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

For example, in the convolutional layer, 256 filters with window size 3 move on the textual representation to extract the features. As the filters move on, many sequences that capture the syntactic and semantic features are generated. In this work, we have considered the dimension of input data is $300*300$, and the dimension of the first layer output data is $149*1$, and successively, the final layer output is of dimension $34*1$. Hence the convolutional layer is an effective way for dimensionality reduction. As specified in Table 5.15, DeepRec does not have data sparsity issues.

Table 5.16: Diversity and Stability of DeepRec and other approaches

Methods	Diversity (D)	MAS (Stability)
FB	0.238	8.204
CF	0.369	7.865
CN	0.281	8.339
CBF	0.215	4.575
RWR	0.317	6.965
PVR	0.402^{+}	7.165
PRS	0.252	4.212^{+}
PAVE	0.323	6.594
DeepRec	0.425*	2.852*

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

(iii) **Diversity**: To alleviate the problem of diversity issue, we integrated the individual results of both CNN and LSTM model using stacked generalized ensemble leaning. This fusion model is proposed in order to capture the contextual similarity-based relevance features and to extract low dimensional latent factors of high dimensional input aiming to cope with diversity issue.

Diversity is defined in terms of content dissimilarity. We group all papers published at a particular venue and extract their corresponding keywords. We apply the similarity score defined in Eqn. 2.29 (Table 5.16). DeepRec shows the best diversity with a diversity score of 0.425. The second-best performer is the method PVR, which shows a diversity score of 0.402. We have considered the average D-score as a threshold to decide whether a particular method provides diversity or not.

(iv) **Stability**: To deal with the stability issue, we propose a stacked generalized ensemble model DeepRec. At the starting stage, keeping in mind the end goal to fully exploit contextual similarity, CNN and LSTM are applied individually on both abstract and title. In this work, we have provided a comprehensive investigation into the stability of the popular recommendation algorithm, as defined in Eqn. 2.30. As shown in Table 5.16, DeepRec shows the minimum MAS than all other standard approaches. It shows a MAS of 2.852 on DBLP dataset, meaning that on average, every predicted rating will shift by 2.852 after adding new data into the that are identical to the system's current predictions to the training data. We have considered the average MAS-score as a threshold to decide whether a particular method provides stability or not. We investigate the execution of the ensemble model DeepRec after adding another 3% of testing data into the training data. Thus the technique of stacked generalized ensemble model DeepRec is desirable.

5.5.8 Discussion on MAG as Evaluation Dataset

Three different venue recommenders are proposed in the thesis. The first in the line, DISCOVER was mainly for a heterogeneous dataset where recommendations are based on keywords, titles and interactions among different fields of studies. Abstracts of papers were checked at the last stage of recommendation. The other two: CNAVER and DeepReC were for a focused, more homogeneous collection with greater cohesion in terms of citations. Hence, based on the use case, we chose two different datasets: MAG for DISCOVER and DBLP for CNAVER and DeepRec. While MAG suited for DISCOVER well, it could not be used for CNAVER and DeepRec. The following reasons determine our choice of datasets in different works of venue recommendation.

(i) The MAG dataset does not have full-text or abstract of the publications. Abstract matching is done at the last stage in DISCOVER. It is to be noted that the average number of papers involved after main path analysis in CS and BIO domains are around 45-85 and 55-95 respectively to perform abstract similarity. But in CNAVER, abstract matching is performed at the beginning to generate feature description by using LDA model. Similarly, during the training phase of DeepRec, an abstract is needed as an input to both CNN and LSTM models.

(ii) MAG also has very good coverage across different domains. On the other hand, there are certain limitations to completeness. Only 30 million papers out of 127 million have some citation data. The MAG contains 528,682,289 internal citations (citations between the papers in the graph). This means each paper in the graph is cited on average 4.17 times. However, in DBLP-Citation-network dataset, there are 3,079,007 papers and 25,166,994 citations, with the average citations per paper is 8.17 – which is comparatively higher and therefore a better candidate for CNAVER and DeepRec.

(iii) In MAG, a significant portion of the papers are disconnected from the network (neither cite nor are cited by any other papers). There are over 80 million such nodes. In DBLP-Citation-network V10 dataset, citation data is extracted from DBLP, ACM, MAG, and other sources. Due to the basic building block of CNAVER is mainly dependent on the citation network, we consider DBLP-Citation-network V10 dataset over MAG.

5.5.9 Some Insights

The overall performance results obtained and discussed in Sec. 5.5 showcase the efficacy of the proposed DeepRec. The excellent overall precision implies that the models can effectively recommend the relevant venues. However, there are a few limitations to our work. The proposed system DeepRec includes multiple parameters from both CNN and LSTM models along with rigorous experimentation. There were multiple classes to classify, and each class requires a huge amount of data to train both CNN and LSTM models. It eventually leads to an increase in computation costs. Dependence of CNNs on the initial parameter tuning (for a good point) to avoid local optima. Thus, a weakness of CNNs is the considerable amount of work they require to initialize according to the problem at hand. This would require some expert knowledge in the domain. One of the limitations of our model is that it cannot recommend venues that are not present in the dataset or have been removed due to less number of papers of that venue since we fixed the minimum paper count to 500. DeepRec may not recommend suitable venues if there are less number of related papers that exist in the training dataset.

5.6 Conclusions

Academic venue recommendation is an emerging area of research in recommendation systems. The proposed techniques are few in numbers and suffer from various limitations. One of the major issues is that of cold-start having two sub-parts: a new venue and a new researcher. Additionally, there exist problems of sparsity, scalability, diversity, and stability in venue recommender system that are not adequately addressed by existing state-of-the-art methods.

This work proposes DeepRec: A deep learning-based scholarly venue recommender system. The proposed method is explicitly modeled mainly based on a stacked generalized ensemble learning. Our ensemble learning-based model is elaborately designed based on a convolution neural network (CNN), and Long short-term memory (LSTM). DeepRec only requires the title and abstract of a new paper to identify scholarly venues. DeepRec could reasonably address all the specified issues. We conducted an extensive set of experiments on a real dataset: DBLP, and showed that DeepRec consistently outperforms the state-of-the-art methods. It demonstrates substantially higher scores of precision@k, nDCG@k, accuracy, and MRR than other best in class techniques. DeepRec proposes top-notch venues when contrasted with cutting edge techniques as far as H5-index.

However, there is still scope for future investigation and improvement. We intend to explore with different datasets and to broaden it for various controls to enhance precision, accuracy, diversity, novelty, coverage, serendipity, and good fortune. We would like to explore the same with the assistance of heterogeneous bibliographic data to recommend scholarly venues.