A NOVEL RECOMMENDER SYSTEM USING DEEP LEARNING TO ENHANCE THE RATING PREDICTION AND TOP-N RECOMMENDATIONS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By:

YAMINI ARORA

(2K17/ISY/16)

Under the supervision of
Dr. Rahul Katarya
Associate Professor, Department of Computer Science Engineering

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2019

CANDIDATE'S DECLARATION

I, Yamini Arora, Roll No. 2K17/ISY/16 student of MTech Information Systems, hereby declare that the project Dissertation titled "A Novel Recommender System using Deep Learning to enhance the Rating Prediction and Top-N Recommendations" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi   Yamini Arora

Date:

CERTIFICATE

I hereby certify that the Project Dissertation titled "A Novel Recommender System using Deep Learning to enhance the Rating Prediction and Top-N Recommendations" which is submitted by Yamini Arora, Roll No 2K17/ISY/16, Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University.

Place: Delhi **Dr. Rahul Katarya**

Date: **SUPERVISOR**

ACKNOWLEDGMENT

I express my gratitude to my major project guide Dr. Rahul Katarya, Associate Professor, Department of Computer Science Engineering, Delhi Technological University, for the valuable support and guidance he provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Yamini Arora

Roll No. 2K17/ISY/16

M.Tech (Information Systems)

E-mail: arorayamini08@gmail.com

ABSTRACT

Different Recommender System Algorithms such as Content-Based and Collaborative-Based have been developed by researchers and data scientists in order to filter a large amount of information available on the internet and hence, recommend only the relevant and important content based on the personalized interests of users. Information acquired explicitly by collecting users' ratings for an item lead to the problem of data sparsity. Many researchers have been working towards the improvement of rating prediction accuracy by integrating the auxiliary information along with the ratings provided by the users. It has been observed in related works that integrating the textual data along with rating data has brought an improvement in the accuracy of estimating the score given to an item by a user and the ranking of top-n recommendations. However, document modeling approaches are different in different research papers. This Project proposes a unique deep neural network text analysis model that includes newly discovered neural network architecture, Capsule Networks stacked on bi-directional Recurrent Neural Network (Bi-RNN) for developing a robust representation of textual descriptions of items and users. The Deep Neural Network text analysis model is integrated with the Probabilistic Matrix Factorization to generate the recommendations. The proposed Model is called as "CapsMF" since it applies the advanced neural network architecture Capsule Networks (Caps) for document representation and MF represents Matrix factorization that is being enhanced to improve recommendations. The experiment is performed on two real amazon datasets and has shown that the rating prediction accuracy and the recall, as well as the precision of top-n recommendations, have improved in comparison to the basic and hybrid Recommendation System Algorithms. Also, Capsule Networks stacked with Recurrent Neural Networks (RNNs) have outperformed the baseline models that involve single Convolutional Neural Networks (CNN) or CNN combined with Bi-RNN. We have also compared different deep learning algorithms and have shown how different text representations affect the recommendations accuracy.

Keywords: Capsule Networks, Collaborative Filtering, Matrix Factorization, Recommender System, Text Analysis

CONTENTS

List Of Figures

List of Tables

List of Abbreviations

RS: Recommender Systems

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

CN: Capsule Network

MF: Matrix factorization

NLP: Natural Language Processing

DL: Deep Learning

PMF: Probabilistic Matrix Factorization

GRU: Gated Recurrent Unit

AIV: Amazon Instant Video

AA: Apps for Android

DNN: Deep Neural Network

BOW: Bag-of-Words

MAE: Mean Absolute Error

RMSE: Root Mean Squared Error

CHAPTER 1

INTRODUCTION

In this digital era, people use e-commerce websites for shopping and social media sites for news and related information. The problem arises when there are varieties of options available while searching for a particular thing. It is tough for the consumers to find the important content they want according to their interest. The solution for this new digital era issue is Recommender Systems (RS). These are being deployed on e-commerce websites and on the social media platforms that extract the personal information related to the users and recommend them the relevant content and information that they want depending on their interest, activity, and behavior. For example, movies, books, clothes, tweets are being recommended to the people visiting different web sites. Recommender System [1] is an application of Web Mining, a process that discovers interests and behavior from large amounts of data without human interference and hence, ranks the most relevant content or items using the rating prediction of the users visiting a website. Different types of Recommendation System Algorithms and techniques have been developed by the researchers and data scientists that filter the information according to the personalized interests of the users. A recommender system can extract information related to a user in two different ways: explicitly or implicitly.

One can get the information related to the users explicitly by collecting the ratings for an item given by a user and it can be acquired implicitly by monitoring the behavior of users on the different websites related to the product [2]. Their behavior includes songs heard, applications downloaded, websites visited, movies watched, books read. Social information can be compiled by scraping the users' accounts on different social websites, for example, what kind of posts they like, their followers, followees and different activities on social media can tell a lot about the preference of the users and hence, personalized recommendations can be more accurate and precise[3][4].

Based on the information about different items and users, two categories of recommendation algorithms developed are Content-Based, Collaborative-based. Out of these, Matrix Factorization technique based on Collaborative Filtering (CF) gives the best recommendations [5]. The limitation observed in the current system is that Information acquired explicitly by collecting users' ratings for an item lead to the problem of data sparsity because every user does not rate every other movie. The accuracy of rating prediction for an item reduces, thereby not satisfying the needs of the customer. Many researchers have been working towards the improvement of rating prediction accuracy by integrating the auxiliary information along with the ratings provided by the users. It has been observed in related works that integrating the textual data along with rating data has brought an improvement in the rating prediction accuracy. However, document modeling approaches are different in different research papers.

Various topic modeling approaches and dnn models [6] are being used to represent the textual description of items and users for improving the rating prediction accuracy and recall and precision of top-n recommendations. Earlier integrated models involved LDA and SDAE, which combined collaborative filtering (CF) technique with the mentioned textual representation strategy, but these strategies use "bag-of-words"(b-o-w) model to generate the item description numerical matrix [7][8]. The "b-o-w" technique in NLP ignores semantics of order of the words and considers only the frequency of words in the text. Hence, it is less influential in comparison to deep learning-based word embedding and architectures. The deep learning architectures such as CNN and RNN take into account the context of words and sequences of text.

The recent models that integrate matrix factorization with textual data generate a textual representation of documents using deep neural network architectures. They are more effective as they extract semantic information from the text. ConvMF [9] generates the item latent model from the textual description of item documents using CNN. The item description latent model integrated with Matrix Factorization enhanced the rating prediction accuracy than the models that utilized a bag of words technique, they are CTR and CDL and the original matrix factorization technique that consider only ratings for generating the recommendations. DRMF [10], dual regularized matrix factorization technique has utilized the description of both the users' and items' reviews and enhanced the matrix factorization technique. Our work improves the text analysis modeling approach that has improved the rating prediction accuracy. Also, the recall and precision of top-n recommendations have improved, leading to good recommendations.

Capsule Networks has brought advancement in deep learning and has shown an efficient performance over CNN. There are various drawbacks of applying CNN in different applications, CNs deal with the pitfalls observed in CNN and has overpowered traditional neural networks for solving the tasks in different areas of application. CN is majorly applied for classifying images in different categories [11].

CN can be used to solve different applications of NLP and RS. We will further see in related work how CN has been applied for classifying text into different categories. The CN can learn the representation of text very efficiently and then can be merged with RS to improve the quality of recommendations.

This project has successfully executed the following points:

- We have addressed the limitations of CNN based document modeling approach as well as CNN and RNN based text representation approach.
- We have proposed a novel and powerful Deep Neural Network text analysis model combining Capsule Networks and Bi-directional RNN to accomplish the textual representation of items' and users' reviews to improve recommendation performance.
- Experiments results prove that exploiting ratings, items' and users' reviews have improved rating prediction accuracy and precision and recall of top-n recommendations.
- The rating Prediction accuracy on proposed model "CapsMF" has shown improvement over PMF, DRMF-Item, DRMF-User, and DRMF by 26%, 4.7%, 3.5%, and 1.7% in terms of MAE.

Different hybrid Recommendation Models using items' and users' reviews as auxiliary information have been described in the Related Work. The theory related to Capsule Networks and Bi-RNN and Recommendation algorithm, Probabilistic Matrix Factorization is explained in the Methodologies section. The proposed deep neural network text analysis model and the integration of PMF with a novel text analysis model are explained in the Proposed Approach section of the Report. The last section of the report deals with Experiment and Results along with Conclusion and Future Work associated with this approach.

1.1 Recommender Systems

There are different categories of RS that are being deployed in different organizations for recommending different products, movies, news, articles to the customers and people around. Here, we describe the different types of RS used in the market. The RSs are based on the various filtering algorithms applied to filter the important information required. The different filtering algorithms used in RS are:

1.1.1 Content-Based Filtering

Content-based filtering algorithms recommend products to the customers on the basis of their historical data and description of the items. These algorithms analyze the description of the items and the user profile for making the recommendations to the users. A user profile is created on the basis of his /her likes and dislikes on the historical data. These are then further used to learn a classifier so that it can estimate whether the consumer would like or dislike a new item. For example, if an end user has given a high rating to a horror movie in the past, then in a future similar genre of movies will be recommended to the users.

1.1.2 Demographic Filtering

Demographic filtering algorithm filters the information on the basis of similarity between the personal attributes of the users, for example, country, age, sex, etc. These algorithms assume users having similar age, country or religion would have similar kind of preferences.

1.1.3 Collaborative Filtering

Collaborative filtering algorithm collects the information about the users explicitly as well as implicitly. In an explicit way, it collects the ratings from the users about the product or movie and in an implicit way, it collects users' information by monitoring their behaviors such as how many times they have heard a song or seen a product. It applies the k-nearest neighbor algorithm to find the most alike users and then calculate the score of items that have not been seen and hence, rated by user and rank the items on the basis of predicted ratings and recommend top-n items.

1.1.4 Hybrid Filtering

Integration of two different filtering algorithms either CF combined with Content-based filtering or CF combined with demographic filtering is referred to as Hybrid Filtering. It makes use of the advantages of the filtering algorithms to increase the accuracy of recommendations.

The different categories of recommender systems are summarized below in Figure 1.1.

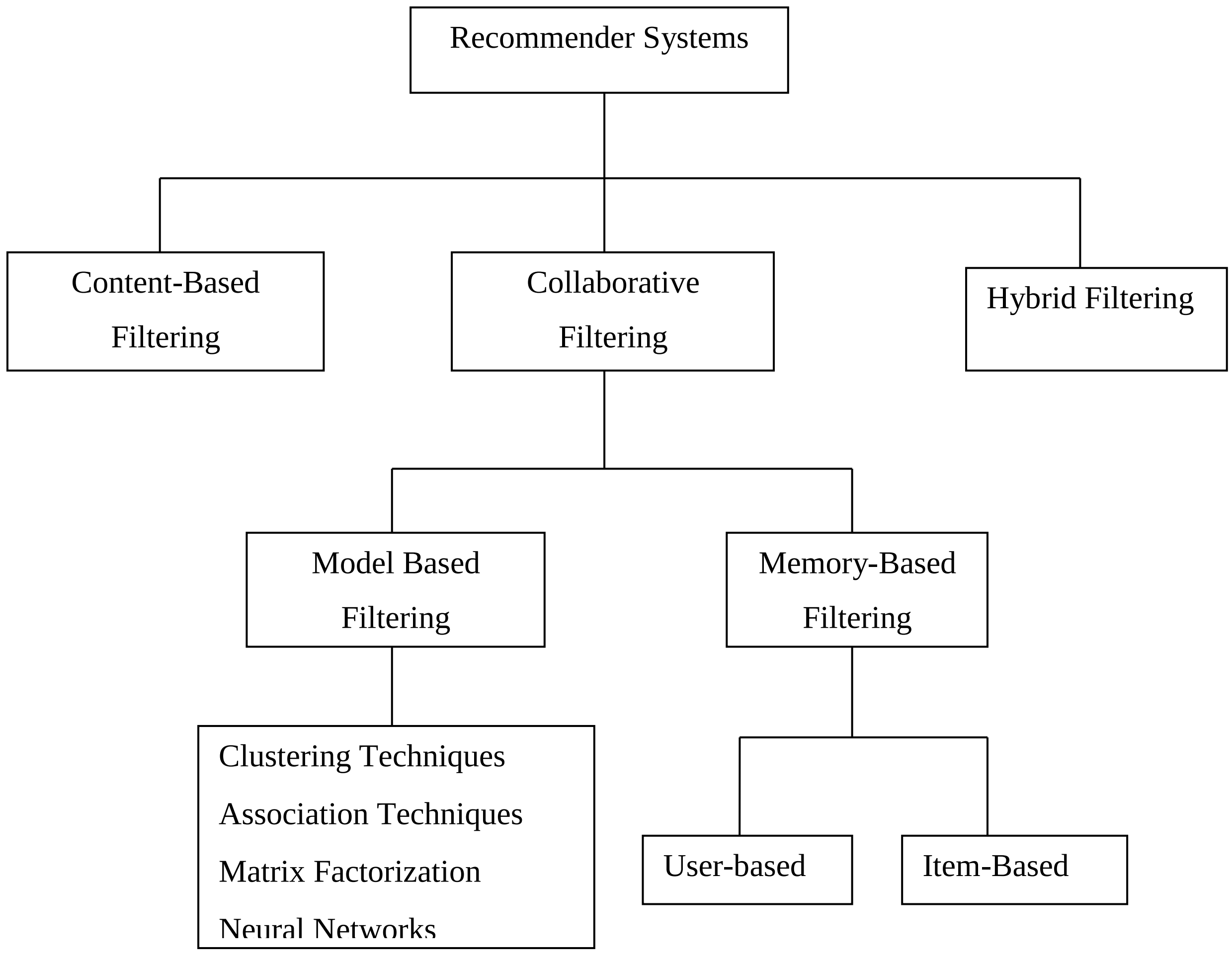


Figure 1.1: Different Types of Recommender Systems

CHAPTER 2

RELATED WORK

2.1 Hybrid Recommendation Systems

Here, we have discussed different models that combine textual data with Recommendation System algorithms providing recommendations to the users. In [12], a brief about the classification of text using CNs has been explained. The different hybrid recommendation systems that have been presented by the researchers that combine textual data and recommendation algorithms are explained further.

CTR, Collaborative Topic Regression [13], incorporates the advantages of collaborative-based, MF algorithm and probabilistic topic modeling approach for recommending research-based articles to the scientists online. Content analysis is done using the Topic modeling approach (LDA) combined with latent factor models using MF for recommending the unseen articles. Their approach worked well as it recommended completely unrated articles to the users that were useful and hence, predictions were done in the right manner. Various efficient algorithms have been proposed in further years that have worked upon these models and have improved them. In HFT [14] model, the authors have developed a statistical model that combines hidden dimensions in the rating matrix with the topics extracted from the text of the review given by users. This methodology helped in predicting users' rating on unseen movies and further recommends them. The models developed by them helped in the discovery of genre and also identified the reviews that are most informative. Their model "HFT", Hidden Factor Topics has addressed the most occurred problem in recommendation tasks, Cold-Start. Their model performed better than the latent factor based recommender system improving accuracy by 5-10% [8]. The model RMR [15], proposed by the authors have combined the content-based and collaborative based filtering techniques to develop a novel integrated RS. Their model "RMR", Rating meets reviews combined the rating model with a topic model to generate recommendations and thereby solving the "cold-start" problem, that is recommending the new products to the users that have not been rating using the description of the item.

Their model also works on the side of the item. For modeling the ratings they have used a mixture of Gaussian, unlike above researchers that have used MF method. They have compared their model with the above described, HFT and CTR and has shown that their model has performed better in terms of rating prediction accuracy. CDL [8], Collaborative Deep Learning brought a change in the topic modeling approach by learning the numeric representation of textual data by applying deep learning techniques on the item description document. In this paper, they have proposed the hierarchal Bayesian Model, which applies deep learning on the textual data and Collaborative filtering approach on rating data in a joint manner. They have used SDAE, Stacked Denoising Autoencoders, the neural network approach combined with the Collaborative Filtering method in order to integrate the rating and content information thereby advancing the quality of recommender systems. Convolutional Matrix Factorization (ConvMF) [9], this model addresses the rating data sparsity by learning the text document using single CNN. The CNN takes into account the surroundings of the word-order that has improved the representation of latent features in the description of the items' documents. Their model integrates PMF and CNN to upgrade the rating prediction task and hence, refine the quality of recommendation of items or movies to the end users. LZhang et al [16] proposed combining CF algorithm with an artificial neural network (ANN) to enhance the scalability and remove the sparsity of recommendation systems. Quadric Polynomial Regression model extracts the latent features, these features then becomes the input to the dnn for predicting the rating scores. The model used to extract the hidden features improves the traditional matrix factorization method [16]. Dual-Regularized Matrix Factorization (DRMF) [10], this is the recent research work in the field of integrating textual data description with PMF. They have exploited the textual description of items and users both and then predicted the rating score. They have also adopted a unique multi-layered neural network model that combines CNN and bi-directional GRU that has given a better latent representation of the text. The content representation regularized the latent models for the items and users in MF.

Different expert recommendation systems have also been designed to recommend the experts in different domains for solving problems using the information retrieval process. It helps in the detection of knowledgeable people expert in their own area. The combination of search engines and NLP can be used to retrieve the experts in this kind of recommendation systems [17]. Different tags and their syntactic patterns associated with Web 2.0 such as audio, video, movies can be exploited for recommendation to solve the cold start problem [18]. A hybrid method has been proposed by the researcher that combines the recommendations of different CF- based approaches for a classification problem. The performance of this approach is analyzed for each use, unlike other approaches that consider the average performance of recommendations across all users. Rating by the users for items has been considered in this research paper [19].

This project is being inspired by the above-related work explained. It brings an improvement in the text analysis model so that the textual representation gets better and the quality of recommendations is enhanced further.

2.2 Text Representation using Capsule Networks

In this project, we have used newly invented neural network architecture, Capsule Networks. We have used it in improving the text analysis model for better representation of views. We have studied different research papers related to the text classification using capsule networks for incorporating it into our project. The research papers gave us insight into how can we implement them and how they can enhance the text representation and further improve the accuracy of the recommendations. This section would further discuss the different implementation of capsule networks for text classification and their performance.

Capsule Networks were proposed by Geoffrey Hinton in 2017, they are a replacement of CNN and called as the better ConvNet. There are several papers published about the application of Capsule Networks in NLP in 2018. Capsule Networks use vector instead of the scalar value. They have added an element of instantiation parameter to each filter. This vector is termed as a capsule.

In [37], the authors have majorly developed capsule network architecture for the classification of text into different categories. They have achieved state-of-the-art performance on single-class text classification. The use of capsules in the CapsNet Architecture allows the transfer of single class classification knowledge to multi-class classification task as well. Text Classification is about reading the passage or textual data and then put the text into different categories. Other similar tasks such as sentiment analysis, toxicity detection can also be done. The Architecture includes the convolutional layer, followed by Primary Capsule Layer then convolutional Capsule Layer, the vectors obtained are then flattened and passed onto the fully connected capsule layer classifying the text into different categories.

Another work has developed an end-to-end Capsule Model for identifying the toxicity and aggression in the comments added by users [38]. Their model has eliminated the need for pre-processing the text. The model performs well on mixed comments that are in English and Hindi both. Various challenges are faced in toxicity detection, such as out-of-vocabulary words, code-mixing of languages and Class Imbalance. Capsule Networks seem to perform better than sequential models at code-mixing. Their architecture involves LSTM for feature extraction, the feature is then passed through the primary capsule layer and then through the convolutional capsule layer. Focal Loss is used to solving the class imbalance problem. The model is applied to different datasets such as Kaggle Toxic Comment Classification dataset, TRAC dataset and has shown good performance over the baseline models.

CNs is also used for user intent detection task. The capsule networks extract and organize the information during supervised intent detection. The representations that are learned can be transferred for solving the task of zero-shot intent detection. These are applied for question answering and dialog systems [39]. In this particular problem, the training of the model is done using the known set of intents; the test is performed on the unseen intents. The model transfers the information about the known intents to the new domain of emerging intents. Semantic Caps Layer and Detection Caps Layer are applied to perform the task. Capsule Networks are now being used for transfer learning as well.

CHAPTER 3

METHODOLOGIES

This section discusses the different methodologies adopted in order to implement a product recommender system.

3.1 Probabilistic Matrix Factorization (PMF)

PMF [20] is collaborative filtering, model-based technique. The task of this algorithm is to infer the preferences of the users based on the scores given by them and what other similar users have liked. In this algorithm, the rating matrix of items and users are given, but it is a sparse matrix because every user does not rate every other movie. The main aim is to fill the missing entries to estimate the score that a user would give to an unseen movie and then recommend those to the user based on predicted rating score.
PMF factorizes the rating matrix into a product of two smaller (low-rank) matrices. Given a rating matrix of M products and N users, and integer rating 1 to 5, the matrix is factorized into two latent feature matrices of users and items.
The two users and items latent feature based matrices is learned from the observed rating matrix. PMF models ratings as drawn from a normal distribution. Mean is determined by user and item via features. It assumes that ratings are normally distributed.

$$\hat{r}_{ui} \sim N(p_u q_i^T, \sigma^2) \tag{3.1}$$

Where r_{ui} represents predicted rating score, N represents normal distribution, p_u represents user-feature vector and q_i represents an item-feature vector.

σ^2 represents variance used to model noise of rating matrix.
User Feature Vector can be represented as:

$$p_{uf} \sim N\,(0, \sigma_p^{\,2}) \tag{3.2}$$

Here, p_{uf} represents user feature vector, N represents the normal distribution, $\sigma_p^{\,2}$ represents variance used to model noise of user latent feature vector matrix.

Item Feature Vector can be represented as:

$$q_{if} \sim N\,(0, \sigma_q^{\,2}) \tag{3.3}$$

Here, q_{if} represents item feature vector, N represents the normal distribution, $\sigma_q^{\,2}$ represents variance used to model noise of item latent feature vector matrix.

In this research, we have compared our model with the basic Probabilistic Matrix Factorization approach. In this proposed integrated model, parameters can be learned using the MAP approach.

3.2 Deep Neural Network for Text Representation

The Deep Neural Network (DNN) Text Analysis Model is being used for generating the text representation of both items and users. A unique and powerful dnn model has been implemented to learn the hidden features of textual reviews. The DNN Text Analysis Model first goes through three Capsule Network (CN) Layers and learns the text representation with these CNs. After learning the semantics and orientation of the text by concatenating three CN Layers, the text is encoded into numbers using Bidirectional-GRU, a variation of RNN. The Framework of DNN Text Analysis Model consists of the Embedding Layer first, and then three CN Layers stacked on a Bi-GRU Layer, and finally the Output Layer that projects the hidden features of the reviews.

3.2.1 Embedding Layer

The input data needs to be encoded in integer; the algorithm does not take strings as input and work on it. Here, the Embedding Layer in Keras is important. It helps in providing a dense representation of words and learns the relationship between the words. In this research, we have used a pre-trained embedding (Glove) [21] to get the dense representation of the reviews of items and users and learn the relationship between the words in the reviews. It helps in extracting the semantic relationship between the words. It encodes the meaning of words into vectors.

3.2.2 Capsule Networks

Capsule Networks [22] are newly developed DNN architecture introduced by Geoffrey Hinton in 2017. These networks brought an improvement in the Convolutional Neural Networks (CNN). The limitations observed in CNN's have been overcome by Capsule Networks. The main advantages of CNs that makes it different from CNN and more efficient are dynamic routing and equivariance. Dynamic routing Algorithm [23] is the main concept behind capsule networks. Analogous to back propagation in Artificial Neural Network, Dynamic Routing between capsules help the capsules in communicating with each other and creating the text representations [24].

3.2.2.1 Routing By Agreement

CN Architecture is developed by the composition of several layers. In the Routing by Agreement procedure, simple entities are represented by the lower level capsules. These basic entities are incorporated to form a complex entity in the higher layers. For example, two simple entities such as nose and lips would form a complex entity, face on the next level. The feedback comes from the upper level of capsules that confirm the predictions made by the capsules at the bottom layer about the formation of a complex entity. If the predictions made by the capsules at a bottom layer regarding the complex entity at higher level matches, the coupling coefficient between the layers of capsules grow. This routing by agreement procedure replaces max-pooling in CNN and saves a lot of information about the different features. This procedure follows a child-parent relationship.

3.2.2.2 Related Mathematics

Given the layers q and q+1 having a and b number of capsules respectively. The task is to calculate the value of activation of the capsules at higher layer q+1 knowing the value of activations at the bottom layer q. Here, u gives the value of activation at bottom layer q. We will calculate the value of v, activation values of capsules at layer q+1 [23]. For a given capsule j at layer q+1, the first step involves the calculation of the prediction vectors of the higher layer using the capsules at a bottom layer.

The prediction vector is calculated by a capsule of layer q for the capsule of layer q+1:

$$\overline{U_{j|i}} = W_{ij} U_i \tag{3.4}$$

Here, Wij represents the weight matrix

The next step is to calculate the Output Vector s_j. It is evaluated using the weighted sum of all the prediction vectors generated by the capsules of lower layer q for the capsule j [14].

$$s_j = \sum_{i=1}^{m} c_{ij} \overline{u_{j|i}} \tag{3.5}$$

The scalar cij represents the coupling coefficient between capsule i (of layer q) and j (of layer q+1). The coupling coefficient is determined using the iterative dynamic routing algorithm that differentiates CN from other traditional neural network architectures.

At the last step, a newly discovered non-linear activation function, Squash function is applied to the output vector v_j of the capsule j:

$$v_j = \text{squash}(s_j) \tag{3.6}$$

It performs the non-linear transformation in the capsule networks keeping the values between zero and one.

3.2.3 Bi-directional Gated Recurrent Unit

We have used Bi-Directional Gated Recurrent Unit stacked on Capsule Networks to enhance the text representation. Gated Recurrent Unit (GRU) solves the different problems observed in RNNs[25] such as long-term dependency and vanishing/exploding gradient problem. These are the most effective neural networks when the context of the input is required. GRU uses two gates, update gate and reset gate in order to maintain the information passed long ago.

Mathematically, Update gate is represented using z_t for the time step t using the following equation,

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \tag{3.7}$$

Where x_t represents input unit, W (z) represents the weight, h_t-1 represents the information from previous input units and U (z) is the weight associated with the previous input units. They are both added and the output is obtained between zero and one due to the sigmoid function applied.

The update gate reduces the vanishing gradient problem as it helps in determining the amount of past information required to be forwarded in the time ahead [26].

The Reset gate also plays an important role to forget the not so required past information. The equation for the reset gate is:

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \qquad (3.8)$$

Where x_t represents input unit, W(r) represents weights associated with the input unit, h_t-1 represents information from previous input units and U(r) represents weights associated with the previous input units. The sigmoid function is applied after summing the results of the input unit and previous input units.

Now, the following formula shows the final output that would be generated using the reset gate that restores the past information and removes the irrelevant information from the past [20]. The new memory content generated using the Reset gate is given as:

$$h_t^{'} = \tanh(W x_t + r_t \odot U h_{t-1}) \qquad (3.9)$$

The h_t' represents the ongoing memory content. The present memory content is generated by multiplying the corresponding weights to the input x_t and multiplying the corresponding weight of previous input units. After that, the element-wise product is calculated between reset gates and weighted previous input units. Then, the non-linear activation function tanh is applied to generate the results. The closing Memory at the present time step t is calculated using the following equation:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t^{'} \qquad (3.10)$$

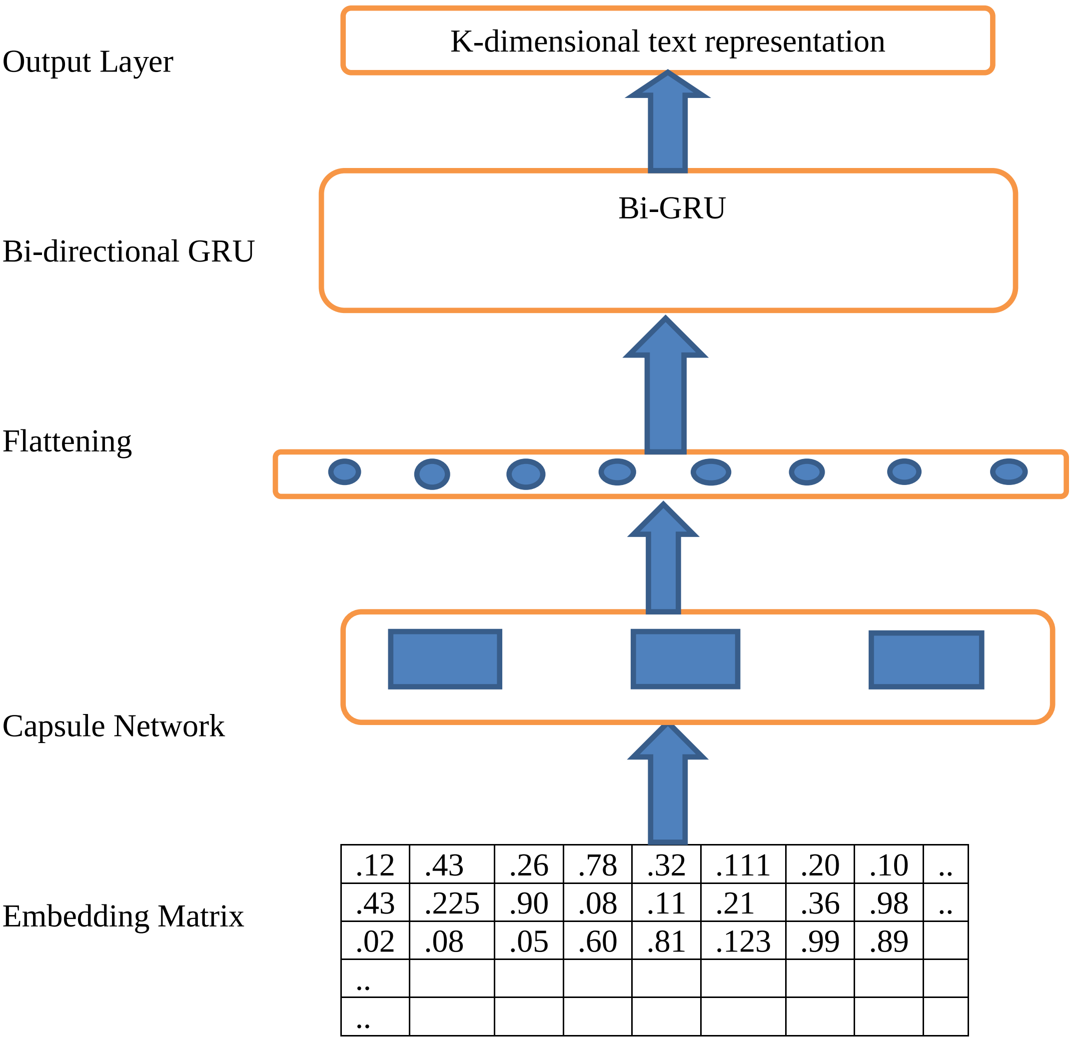
In this equation, h_t represents the closing memory content vector generated, z_t represents the update gate and h'_t represents present memory content.

3.2.4 Output Layer

After going through the embedding layer, Capsule Network Layer and bi-GRU Layers, the features extracted by the last layer are outlined on a k-latent dimensional space by using hyperbolic tanh activation function.

$$\alpha = \tanh(G * h + b) \tag{3.11}$$

Here, G represents the projection matrix obtained in the output Layer, h represents the output obtained from the bi-directional GRU layer and b is the bias. Dropout [27] is also used in the model to prevent the over-fitting in the projection matrix. The main aim behind Dropout is dropping hidden or visible neurons in a neural network model to prevent the over-fitting. Dropout is basically a regularization technique that helps in reducing the interdependence between the nodes in the neural network.



.12	.43	.26	.78	.32	.111	.20	.10	..
.43	.225	.90	.08	.11	.21	.36	.98	..
.02	.08	.05	.60	.81	.123	.99	.89	
..								
..								

Figure 3.1: Deep Neural Network text analysis model generating the representation of users' and items' reviews

The above-mentioned text analysis neural network layers described converting a raw document into a dense distributed representation as shown in Figure 1. Given X, the document of items' reviews and Y, the documents of users' reviews, the dnn text analysis model transforms X to beta and Y to alpha. Alpha and beta are the dense distributed representation of the raw documents of users and items respectively.

Keras functional API is being used to define the neural network model for the purpose of text analysis and getting the projection matrix for the further process. The model parameters are learned by applying the backpropagation algorithm. The input sequence, output, and the corresponding loss function are defined for the model. In the Proposed Approach section, the algorithm is being explained used for implementing the dnn text analysis model based on Keras.

CHAPTER 4

PROPOSED WORK

In this segment, we have described the algorithm implemented for text analysis of reviews of items and users of different products. The matrix obtained from the text analysis is then integrated with the collaborative filtering based PMF method to generate more accurate and precise recommendations.

4.1 Deep Neural Network Text Analysis Model

The Algorithm gives the details about the implementation of Text Analysis Neural Network Model that involves different layers starting from Embedding Layer, Capsule Network Layers and Bi-directional GRU layers that extract features from the description documents and project them into k-dimensional space. It outputs a latent document k-dimension matrix that helps in improving the product recommendations.

The earlier similar models have used convolutional neural networks (CNN) [28] for latent document matrix that had certain limitations. These limitations have been overcome using capsule network layers.

The Algorithm is documented below, showing all the different layers that have been applied and implemented to get the k-dimensional matrix of the user and item reviews document. The document is first converted into a numerical matrix using the embedding layer representing the text as a sequence of word embedding. Then, the document representation is learned through capsule networks and the semantics of the reviews are encoded by going through the bi-directional GRU layers.

Algorithm: Deep neural network (dnn) text analysis model

Input: vocabulary size, e, d, s, gru_dim, k, init_W, n_capsule, n_routings, caps_dim;

Output: DNN text analysis Model

{Embedding Layers}

input_sequence = Input (shape= (d * s,))

seq_emb = Embedding (vocabulary size, e, weights= [init_W / 20], trainable=False) (input_sequence)

{Capsule Layers}

caps_1 = Capsule (num_capsule=n_capsule, dim_capsule = caps_dim,

routings = n_routings, share_weights=True)(seq_emb)

x = Flatten () (caps_1)

caps_2 = Capsule (num_capsule=n_capsule, dim_capsule=caps_dim,

routings = n_routings, share_weights=True) (x_3)

x = Flatten () (caps_2)

caps_3 = Capsule (num_capsule=n_capsule, dim_capsule=caps_dim,

routings=n_routings, share_weights=True)(caps_3)

x = Flatten () (caps_3)

x = concatenate ([caps_1, caps_2, caps_3])

{Bi-directional GRU Layers}

x_gru = Bidirectional (GRU (gru_dim, return_sequences=True)) (x)

x = Flatten () (x_gru)

{Output Layer}

seq_dropout = Dropout (dropout_rate) (x)

out_seq = Dense (output_dimesion, activation='tanh') (seq_dropout)

dnn_text_analysis = Model (input_sequence, out_seq)

dnn_text_analysis (optimizer='rmsprop', loss='mse')

return dnn_text_analysis ;

4.2 CapsMF: Integration of deep neural network text analysis model and PMF

Here, we describe the integration of PMF with the text analysis model and the equations involved in generating user latent feature vector and item latent feature vector. The main aim of the MF technique is to generate user and item hidden feature vectors[29] that particularly describe the choices of the users and they are obtained from both the ratings and the reviews of items and users.

4.2.1 User Latent Vector

User i as a latent vector represented as:

$p_i = €_i + \alpha_i$ (4.1)

Where p_i represents user latent vector, $€ \sim N(0, \sigma_p^2 I)$, α_i represents user document latent vector.

Integrating the above we obtain the user latent model as:

$p_i \sim N(\alpha_i, \sigma_p^2)$, here I is the identity matrix

4.2.2 Item Latent Vector

Item j as a latent vector represented as:

$q_j = €_j + \beta_j$ (4.2)

Where q_j represents item latent vector, $€ \sim N(0, \sigma_q^2 I)$, β_j represents item document latent vector.

Integrating the above we obtain item latent model as:

$q_j \sim N(\beta_j, \sigma_q^2 I)$, here I is the identity matrix

4.2.3 Prediction

There are M users and N items, R represents the rating matrix; then the missing rating that users have not given is predicted using the formula "$P^T Q$", where P represents the user latent matrix of size k X M and Q represents the item latent matrix of size k X N. The equation is given below:

$$\overline{R} = N(P_i^T Q_j, \sigma^2) \tag{4.3}$$

Next is the **Flow Chart** representing the Learning of different parameters in this model.

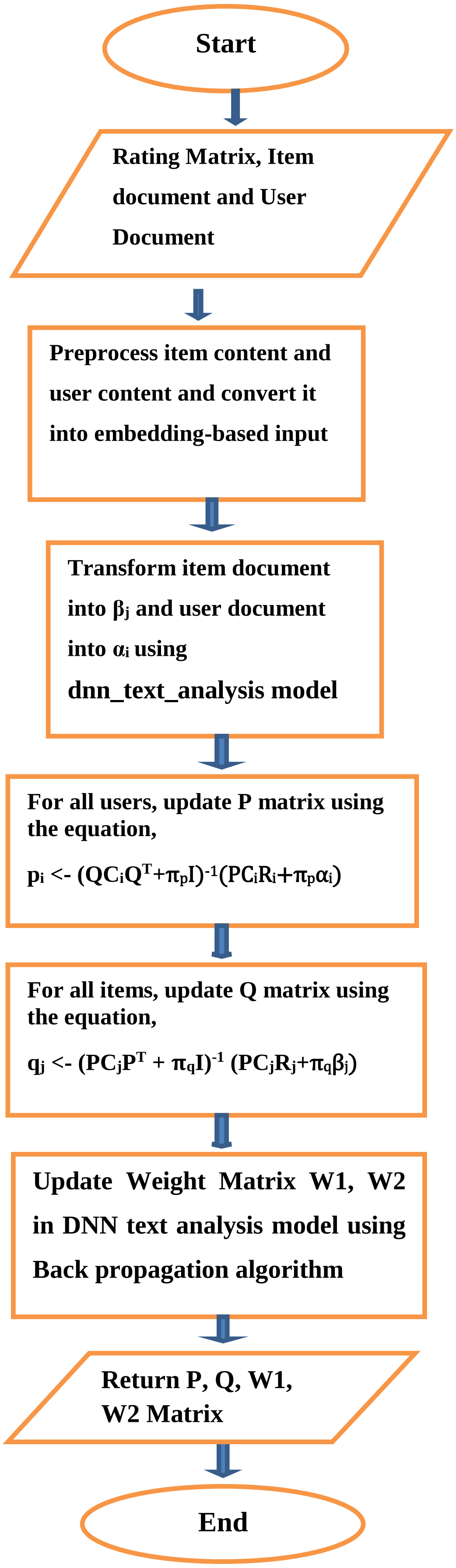


Figure 4.1: Learning Parameters in the proposed Model CapsMF

Figure 4.1 above describes the learning of different parameters used in the model "CapsMF" using Maximum a Posteriori (MAP) approach. First, rating matrix, item document, and user document are given as input to the model. Then, Item Content is pre-processed by removing the stop words and applying lemmatization and convert into embedding based input such that the document matrix can be learned through deep neural network text analysis model. Similarly, User content is pre-processed and converted into embedding based input. After this, the item document matrix and user document matrix are given as input to text analysis model, where the text representation is learned. Item document is transformed into β_j and user document is transformed into α_i using the algorithm described above. After transforming the item reviews document and user reviews document, the user latent matrix and item latent matrix is updated. The Weight Matrices W1 and W2 applied in DNN text analysis model are updated using Back Propagation Algorithm. When the number of iterations is over, return the User Latent Matrix, Item Latent Matrix, and Weight Matrices.

4.3 Why Capsule Networks over CNN?

Capsule Networks are newly discovered Neural Network Architecture by Geoffrey Hinton in 2017. They have been applied in various domains such as Image Classification, NLP, text classification, etc [30]. This is a major breakthrough in the domain of Deep Learning. It has overcome the various drawbacks in the CNN and shown effective results in the classification problems. Therefore, we have incorporated these in our text analysis model replacing the CNN so that the text representation is more effective and powerful. CNN does not take into account the spatial information whereas CN does and that increases its power. The max pooling function used in CNN to extract the highest features is not efficient and results in reduced computational complexity. It has been replaced by Dynamic routing Algorithm that is very efficient in recognizing the spatial relationships and hence improving the classification accuracy. "Routing-by-agreement" proposed by Geoffrey Hinton helps in retaining the relevant information and features of the content and sends the features from lower-level to higher level only if the content is being matched [31].

Replacing the CNN by CN in the deep neural network text analysis model has increased the rating prediction accuracy and hence reducing the root mean squared error (rmse).

4.4 Why Deep Neural Networks for text analysis?

In recent times, the research in deep learning has enhanced. It is being used in different domains such as computer vision, image classification, NLP, Recommender Systems, etc. They have improved the accuracy in different classification and regression problems. They are more complex to understand and take more time in training, but the results generated are more effective and precise. They perform better in extracting semantics from documents and hence, generating an effective text representation of documents [36]. As we saw Capsule Networks perform better than CNN, as they have a dynamic routing algorithm that simultaneously takes feedback from simple entities. Similarly, Bi-directional GRU is good at temporal modeling, that is, it stores information related to past and present. These neural networks are good in feature extraction and dimensionality reduction. Hence, we have proposed a dnn model for text analysis and generating the representation of items' reviews and users' reviews.

CHAPTER 5

DEEP LEARNING ALGORITHMS

In this chapter we have discussed the deep learning algorithms that are being used recently in NLP for text representation. We have applied varied DL algorithms and compared the accuracy of recommendations on varying the text representation algorithms.

5.1 Convolutional Neural Network (CNN)

CNN architecture has different layers for attribute extraction; they are convolutional layer, max pooling layer, activation layer and concatenation layer as shown in Figure 5.1. The convolutional layer applies different size of filters on the window of words to bring out new features from the text. The kernels of different sizes, three, four, and five have been applied on the textual window. The features obtained from each filter are then passed through the max-pooling layer. The max-pooling layer draws out the maximum value from the feature map obtained [36]. Large number of features are obtained after applying the max-pooling are then concatenated together by applying a flatten operation. This converts a 2-d feature matrix into a 1-d feature matrix. The activation function hyperbolic tangent function is applied to incorporate non-linearity.

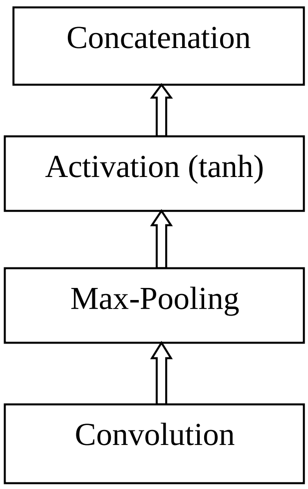


Figure 5.1: CNN Layers

5.2 Long Short Term Memory (LSTM)

LSTM is a modification of RNN which is commonly used in NLP tasks such as Sentence Classification. The simple RNN suffers from many drawbacks; Vanishing Gradient is one of them. The RNN is unable to retain long term memories. Hence, a variant of RNN, LSTM is introduced [26]. The carry, which acts as a conveyor belt is introduced in the RNN architecture. It is fed into the architecture. The carry is computed using the input, output and the previous carry. It is very much effective in learning long term memory. These networks are successful in learning the data for long periods of time.

5.3 Auto-Encoders

Auto-encoders are another special kind of unsupervised neural network architectures. In this neural network, the input given and output obtained is almost the same. They learn the low-level representation of the input data in an unsupervised manner. The representation generated is then again transformed back to original data. The auto-encoders consist of three important components, Encoding, Latent-view, and Decoding. The encoding architecture takes the input and passes it through the reducing number of neurons to generate the low-level representation. The Latent View representation reduces the input to the lowest form and then preserves the information. It is then given to the decoding architecture, which is the same as the encoding part of the architecture but has the increasing number of neurons that generates almost similar input [31].

CHAPTER 6

EXPERIMENT AND RESULTS

We have conducted the experiments on the two amazon datasets, one is “Amazon Instant Video (AIV)” and the other is “Apps for Android (AA)”. Both the datasets are taken from Amazon Product Reviews site (http://jmcauley.ucsd.edu/data/amazon/). This site contains reviews of different kind of products as well as metadata regarding the product. We have considered only the reviews and ratings of the products. We have taken 5-core datasets, in these 5-core datasets, the users and items have five reviews each. Table 6.1 gives a description of the datasets that have been used for implementation. The Rating is given from one to five to an item by a user.

Table 6. 1: Description of the datasets

Datasets	Number of Users	Number of Items	Total Reviews/Ratings	Rating Scale
AIV(Amazon Instant Video)	5,130	1,685	37,126	1-5
AA (Apps for Android)	87,721	13,209	752,937	1-5

We have conducted the experiments on Google Colaboratory Notebook using GPU since the dataset is too large and deep learning models take time in training. We have calculated the RMSE and MAE for comparing the accuracy of different rating prediction models. We have also calculated the precision and recall for assessing the accuracy of recommendations.

6.1 Different Evaluation Metrics

The formulas used for different evaluation metrics is described here:

$$\text{MAE} = \Sigma_{i,j} \mid r_{ij} - \overline{r_{ij}} \mid / N \qquad (6.1)$$

Where r_{ij} represents the actual rating, r'_{ij} represents the predicted rating and N represents the count of samples.

RMSE = square root $(\sum_{i,j} (r_{ij} - r'_{ij})^2) / N)$ (6.2)

Precision and Recall are calculated to assess the accuracy of top-N recommendations, we have calculated these metrics for top-300 recommendations given to the users.

Precision @ N = (# of recommended items @N that are relevant} / (# of recommended items @ N) (6.3)

Recall @ N = (# of recommended items @N that are relevant) / (total #of relevant items) (6.4)

6.2 Parameter Settings

Different values that have been set for π_p and π_q are given in Table 6.2 and Table 6.3.

Table 6. 2: Parameter Settings of π_p and π_q for rating prediction

	AIV		AA	
Models	Π_p	Π_q	Π_p	Π_q
PMF	0.1	0.01	0.1	0.01
ConvMF	10	1	10	10
DRMF-Item	1	10	1	100
DRMF-User	10	0.1	100	0.1
DRMF	100	10	10	100
CapsMF	100	10	10	100

Where $\Pi p = \sigma^2 / \sigma_p^2$ and $\Pi q = \sigma^2 / \sigma_q^2$

Also, parameters related to Capsule Layer are set accordingly for "CapsMF":

$n_c = 50$; $n_r = 5$; $n_d = 100$

Here, n_c = number of capsules, n_r = number of routings and n_d = Capsule Dimension

Table 6. 3: Parameter Settings of Π_p and Π_q for ranking of items

	AIV		AA	
Models	Π_p	Π_q	Π_p	Π_q
PMF	1	5	5	10
ConvMF	0.5	5	10	1
DRMF-Item	0.5	5	10	1
DRMF-User	5	0.5	10	1
DRMF	5	0.5	10	1
CapsMF	5	0.5	10	1

Where $\Pi p = \sigma^2 / \sigma_p^2$ and $\Pi q = \sigma^2 / \sigma_q^2$

Also, parameters related to Capsule Layer are set accordingly for "CapsMF":

$n_c = 50$; $n_r = 5$; $n_d = 100$

Here, n_c = number of capsules, n_r = number of routings and n_d = Capsule Dimension

N = 300 for top-N recommendation

6.3 Results

The different values of evaluation metrics obtained on implementing the different models are shown in this section. The baseline models are PMF, ConvMF, DRMF-Item, DRMF-User, and DRMF as explained earlier in the related work. For the good results, we have averaged the results of five-folds in order to estimate the prediction and recommendation performance accurately.

Below Table 6.4 shows the results of rating prediction performance of different models on AIV dataset in term of MAE and RMSE.

Table 6. 4: Rating Prediction Performance of different models on **AIV Dataset**

Models	MAE	RMSE
PMF	0.9627	1.2088
ConvMF	0.7821	1.0378
DRMF-Item	0.7419	1.0101
DRMF-User	0.7325	0.9866
DRMF	0.7190	0.9661
CapsMF	**0.7064**	**0.9593**

Table 6.5 shows the results obtained when rating prediction is implemented using different Models on AA dataset. Rating Prediction performance of different models is shown on the basis of MAE and RMSE respectively.

Table 6. 5: Rating Prediction Performance of different models on **AA Dataset**

Models	MAE	RMSE
PMF	1.1829	1.4773
DRMF-Item	1.0664	1.3763
ConvMF	1.00683	1.2912
DRMF-User	0.9638	1.2617
DRMF	0.9071	1.1930
CapsMF	**0.8878**	**1.157**

The best values are shown in bold.

According to the accuracy comparisons shown in the tables above, it can be concluded that CapsMF, the model described has shown improvement in terms of MAE and RMSE on the two datasets: AIV and AA.

6.4 Graphical Analysis of Results

Here, we have shown the ranking performance of different Models on the two datasets AIV and AA graphically in term of Precision and Recall.

Below Figure 6.1 and Figure 6.2 graphically represent the top-300 recommendation performance of different Models on AIV dataset in term of precision and recall respectively.

AIV

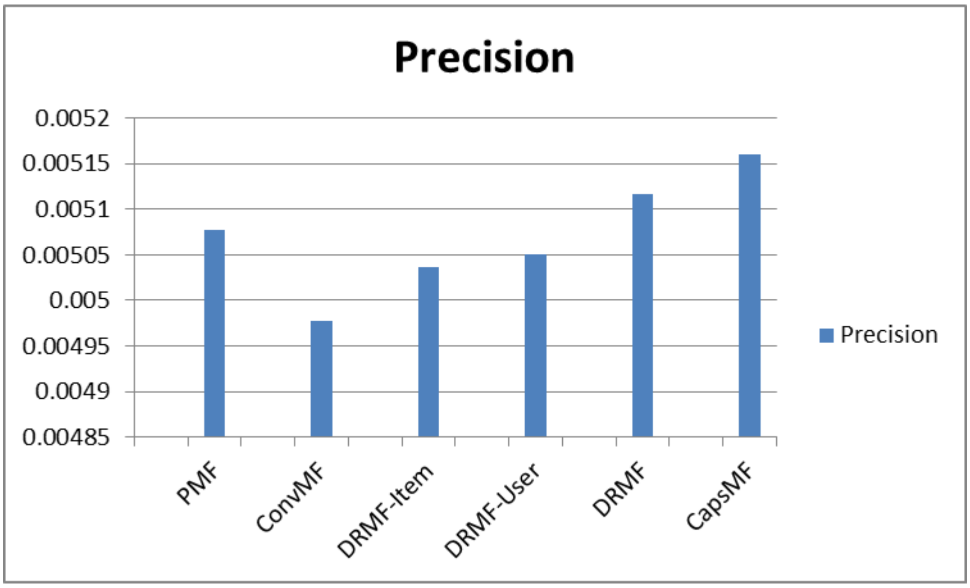


Figure 6.1: Top-300 ranking performance in terms of Precision

AIV

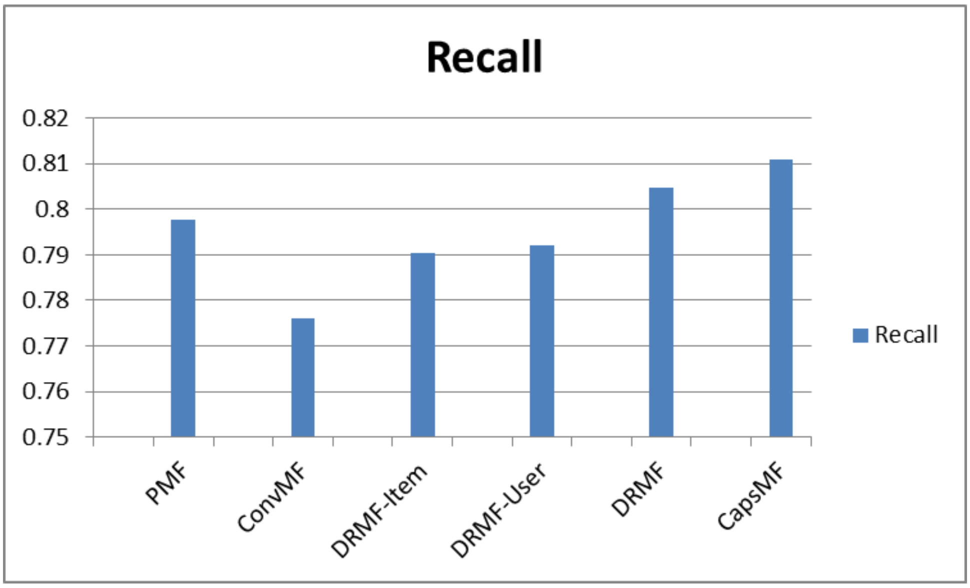


Figure 6.2: Top-300 ranking performance in terms of Recall

Figure 6.3 and Figure 6.4 show the results obtained while implementing the task of Top-N recommendations on AIV dataset in term of precision and recall respectively. Here, the value of N varies from 50 and ends at 300.
It is noteworthy that there is not much gain in top-n recommendation performance.

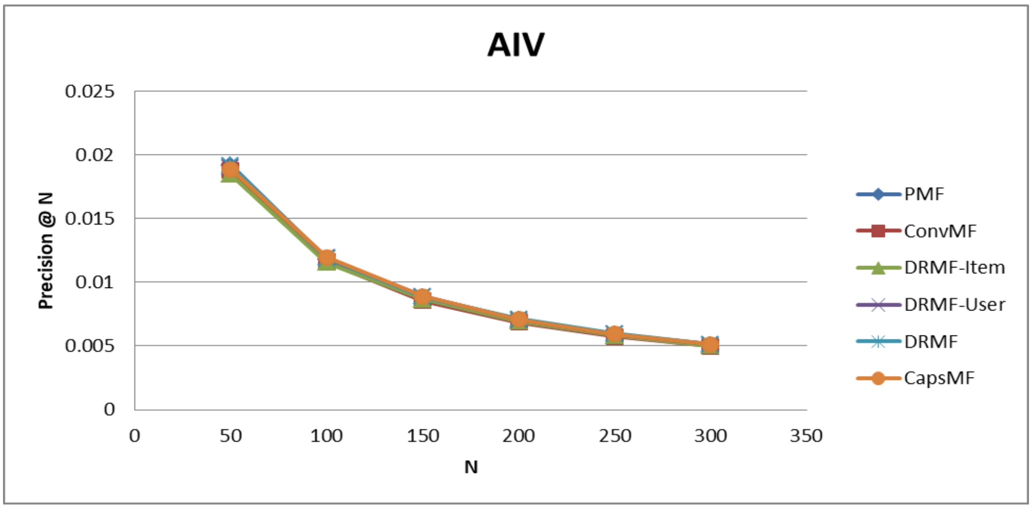


Figure 6.3: Top – N recommendation in term of Precision

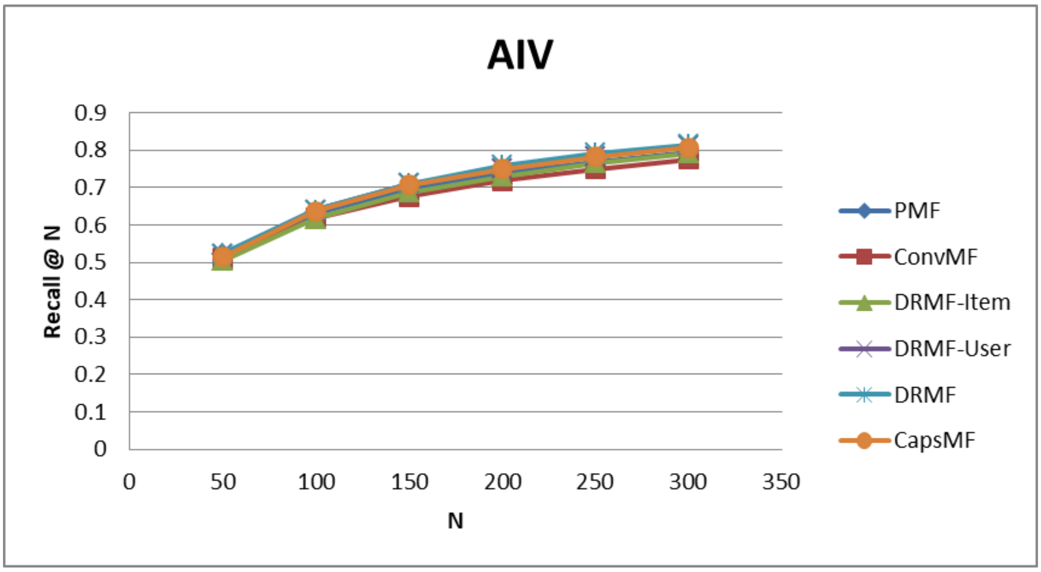


Figure 6.4: Top-N Ranking Performance in term of Recall

We can observe in the charts above that the proposed approach “CapsMF” has shown better recommendation performance on AIV dataset. Both RMSE and MAE have smaller values than the baseline Models implemented. All the Models have been implemented on Google Colaboratory Notebook in Python using GPU for getting the results faster. Higher values of Precision and Recall prove that recommendation performance has improved over the baseline Models described in the related work.

Following are the graphical representations of the results obtained on "Apps for Android (AA)" dataset. Figure 6.5 shows the ranking performance of top-300 recommendations in term of Precision graphically on the AA dataset.

AA

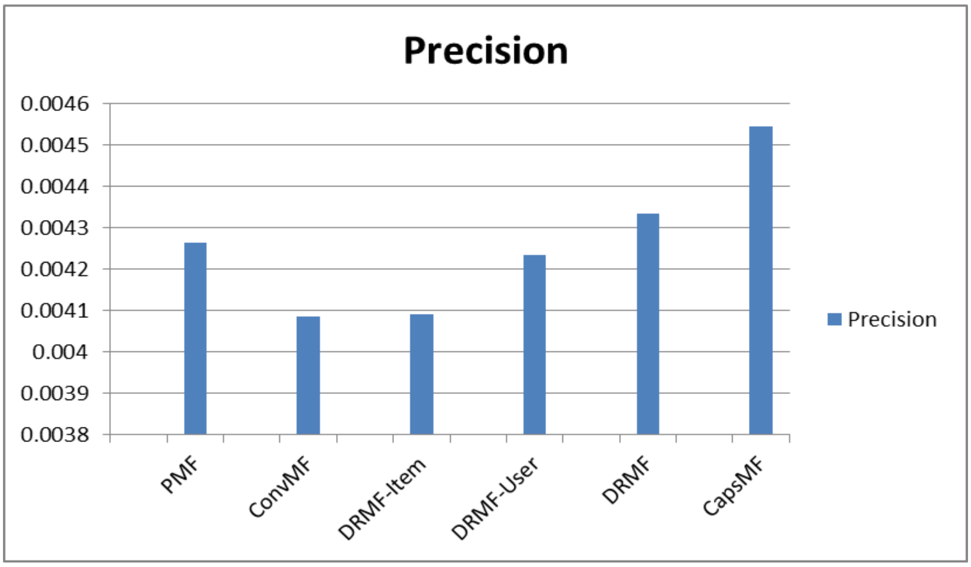


Figure 6.5: Top-300 recommendation performance in terms of Precision

Figure 6.6 and Figure 6.7 graphically represents the ranking performance of items in term of Precision and Recall on AA dataset. The value of N in Top-N recommendation performance varies from 50 to 300.

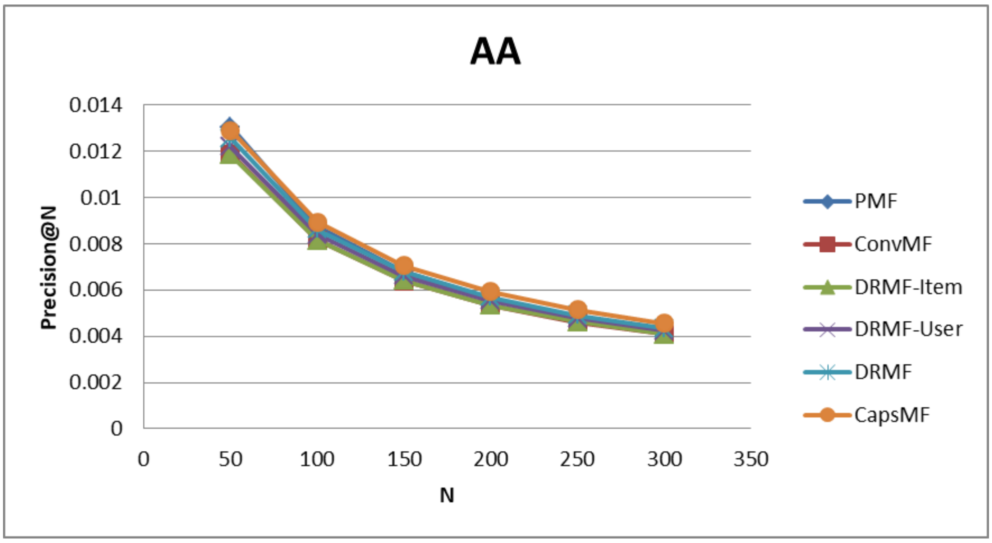


Figure 6.6: Top-N Recommendation Performance in term of Precision

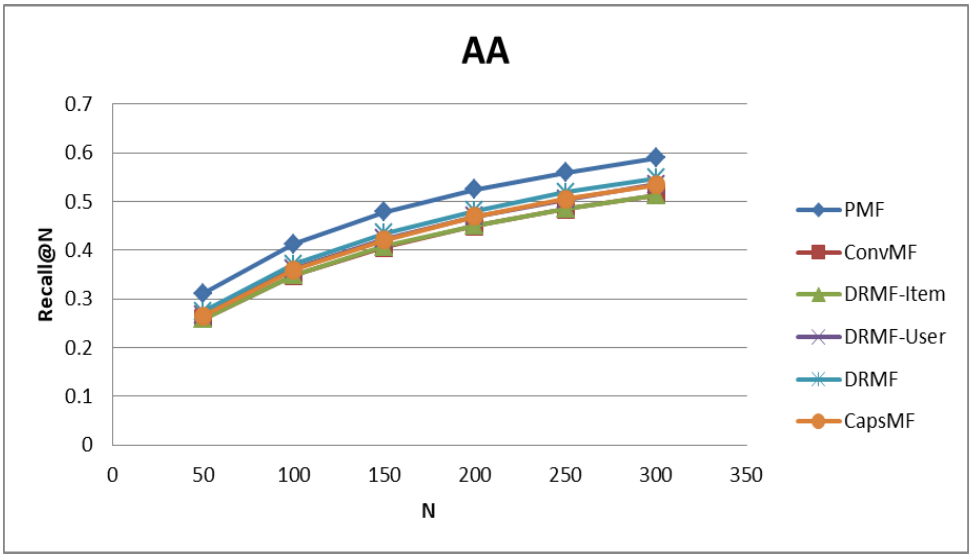


Figure 6.7: Top-N Ranking in term of Recall

Similarly, when the proposed approach "CapsMF" is implemented on Apps for Android (AA) dataset has shown improvement over the baseline Models on two different tasks, one of rating prediction and another, the ranking performance of items that is being recommended to the users.

6.5 Performance Comparisons

Above we have shown the results in the tables and analysis of the different models on rating prediction and top-n recommendations using column charts. As we can observe that CapsMF, the proposed deep neural network model for text analysis has shown better performance when integrated with MF Technique. For the task of rating prediction on AIV dataset, CapsMF has shown improvement over PMF, DRMF-Item, DRMF-Use, and DRMF by 26%, 4.7%, 3.5%, and 1.7% respectively in term of MAE. In term of RMSE, CapsMF has improved over PMF, DRMF-Item, DRMF-User, and DRMF by 20.6%, 5.02%, 2.76%, and 0.7% respectively.

For the task of top-n recommendation on AIV dataset, CapsMF has shown significant improvement when compared with the other models of similar type in terms of both precision and recall. We have chosen n = 300. It has improved by 1.6% when compared with PMF and 2.3%, 2%, and 0.8% improvement w.r.t DRMF-Item, DRMF-User and DRMF in term of Precision. Similar significant improvements have been observed in terms of Recall @ 300 on AIV Dataset.

When the task of rating prediction and top-n recommendation are implemented on "Apps for Android (AA)" dataset, significant improvements have been recorded. CapsMF has shown better rating prediction accuracy over the baseline models. It has outperformed the baseline models PMF, DRMF-Item, DRMF-User, and DRMF by 25%, 16.7%, 7.8%, and 2.12% respectively in term of MAE. Similarly, RMSE values have also been reduced when calculated on the models by 21.6% when compared with PMF, 15.9%, 8.29%, 3.01% when compared with DRMF-Item, DRMF-Use, and DRMF respectively. We can observe that the proposed model CapsMF has shown better results on a larger dataset, "Apps for Android (AA)".

6.6 Comparison of Deep Learning Algorithms

In this section we have graphically shown the results obtained on varying the deep learning algorithms for text representation. We have shown how the recommendation accuracy gets affected if we use different combinations of neural network architectures for representing the text in the numerical matrix. We have used the AIV dataset to show the variation. On the basis of results obtained in Figure 6.8, we can conclude that the combination of Capsule Layer with GRU has shown good accuracy in rating prediction.

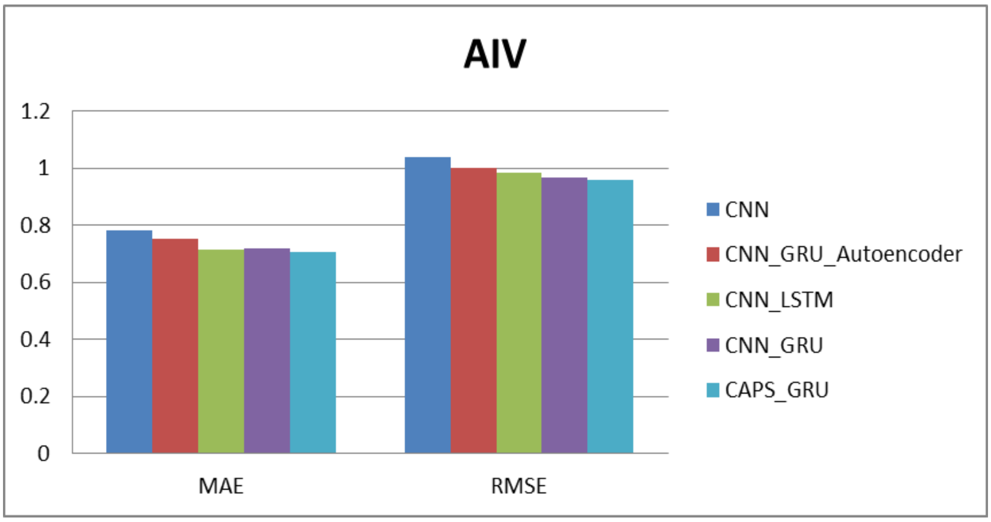


Figure 6.8 : Comparison of Deep Learning Algorithms on Rating Prediction

Similarly, in Figure 6.9 and Figure 6.10, we have compared different DL agorithms in term of Precision and Recall.We can observe that the textual representation using Capsule Layer and GRU has shown good performance.

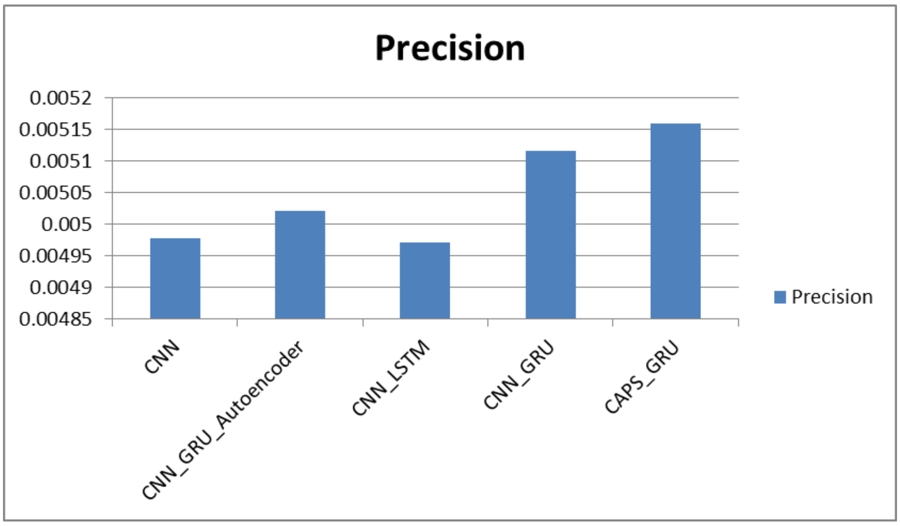


Figure 6.9: Comparison Of DL algorithms in term of Precision

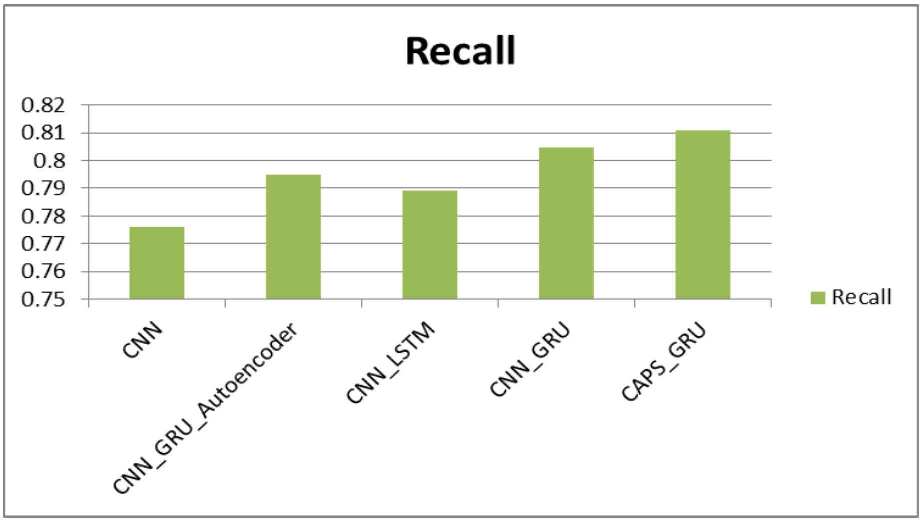


Figure 6.10: Comparison of DL Algorithms in term of Recall

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this research paper, we have addressed the limitations of CNN based document modeling approach as well as CNN and RNN based text representation approach. We have proposed a novel and powerful dnn text analysis model stacking Capsule Networks and Bi-directional RNN to generate the textual representation of users' and items' reviews integrated with PMF to improve recommendation performance. Experiments results have proved that exploiting ratings, items' and users' reviews have improved rating prediction accuracy and precision and recall of top-n recommendations. It can also be concluded that the proposed model, "CapsMF", performs better on larger datasets as compared to smaller datasets. Both the users and items reviews have been considered to build the separate deep neural network for text analysis, hence, it has resulted in better recommendations to the users and also solves the cold-start problem in recommender systems.

There are still certain limitations that come with Capsule Networks; one that has been observed while performing the experiments is capsule networks takes times in training. The advantage is it takes fewer data points in training as compared to CNN.

In future works, we would refine the deep neural network for text analysis with new deep learning technologies getting discovered for analysis of text for getting a better representation of the text. We would apply the proposed model in different recommendation datasets and scenarios such as social recommendations [32], joint-recommendation, group recommendation [33], etc. Other contextual information related to users and items such as images, tweets [34][35], metadata about products can be integrated with collaborative based filtering technique to enhance the personalized recommendations.

REFERENCES

[1] M. Razghandi and S. A. H. Golpaygani, “A Context-Aware and User Behavior-Based Recommender System with Regarding Social Network Analysis,” *2017 IEEE 14th Int. Conf. E-bus. Eng.*, pp. 208–213, 2017.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Knowledge-Based Systems Recommender systems survey,” *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.

[3] M. Liphoto, C. Du and S. Ngwira, "A survey on recommender systems," 2016 International Conference on Advances in Computing and Communication Engineering (ICACCE), Durban, 2016, pp. 276-280. doi: 10.1109/ICACCE.2016.807376..

[4] X. Kong and S. Member, “A Survey of Collaborative Filtering-Based Recommender Systems : From Traditional Models to Hybrid Models Based on Social Networks,” *IEEE Access*, vol. 6, pp. 64301–64320, 2018.

[5] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion Models for recommendation systems," in Big Data Mining and Analytics, vol. 1, no. 4, pp. 308-323(IEEE), December 2018. doi:10.26599/BDMA.2018.9020008

[6] Zhang, Shuai and Yao, Lina and Sun, Aixin and Tay, Yi, “Deep Learning Based Recommender System: A Survey and New Perspectives,” ACM Comput. Surv., February 2019, vol. 52, pages 5:1--5:38, doi: 10.1145/3285029.

[7] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, “Probabilistic author-topic models for information discovery,” *Proc. 2004 ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’04*, no. 1990, p. 306, 2004.

[8] H. Wang and N. Wang, “Collaborative Deep Learning for Recommender Systems,” Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1235-1244, doi:10.1145/2783258.2783273, 2015.

[9] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu., "Convolutional Matrix Factorization for Document Context-Aware Recommendation," Proceedings of the 10th ACM

Conference on Recommender Systems, page 233--240. New York, NY, USA, ACM, (2016).

[10] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, “Knowledge-Based Systems Dual-regularized matrix factorization with deep neural networks for recommender systems R,” *Knowledge-Based Syst.*, vol. 145, pp. 46–58, 2018.

[11] Jaiswal, Ayush, AbdAlmageed, Wael and Natarajan, Premkumar, "CapsuleGAN: Generative Adversarial Capsule Network," cite arxiv:1802.06167 , pp. 1–10, 2018.

[12] R. Katarya and Y. Arora, "Study on Text Classification using Capsule Networks," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 501-505, doi: 10.1109/ICACCS.2019.8728394

[13] C. Wang and D. M. Blei, “Collaborative Topic Modeling for Recommending Scientific Articles,” Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 448-456, New York, NY, USA, ACM 2011.

[14] J. Mcauley, “Hidden Factors and Hidden Topics : Understanding Rating Dimensions with Review Text.” Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 165--172, ACM, 2013.

[15] Guang Ling, Michael R. Lyu, and Irwin King, "Ratings meet reviews, a combined approach to recommend," In Proceedings of the 8th ACM Conference on Recommender systems (RecSys '14). ACM, New York, NY, USA, 105-112. DOI: https://doi.org/10.1145/2645710.2645728, 2014.

[16] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A Recommendation Model Based on Deep Neural Network," in IEEE Access, vol. 6, pp. 9454-9463, 2018. doi: 10.1109/ACCESS.2018.2789866

[17] N. N. Khasmakhi, M. A. Balafar, and M. R. F. Derakhshi, “Engineering Applications of Artificial Intelligence The state-of-the-art in expert recommendation systems” *Eng. Appl. Artif. Intell.*, vol. 82, no. August 2018, pp. 126–147, 2019.

[18] F. M. Belém, A. G. Heringer, J. M. Almeida, and M. A. Gonçalves, “Exploiting syntactic and neighborhood attributes to address cold start in tag recommendation,” *Inf. Process. Manag.*, vol. 56, no. 3, pp. 771–790, 2019.

[19] D. Rojo and P. Valdiviezo-díaz, “Hybrid Collaborative Filtering Based on Users

Rating Behavior,” IEEE Access, vol. 6, 2018.

[20] Ruslan Salakhutdinov and Andriy Mnih, " Probabilistic Matrix Factorization," In Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07), Curran Associates Inc., USA, 1257-1264, 2007.

[21] J J. Pennington, R. Socher, and C. D. Manning, “GloVe : Global Vectors for Word Representation,” Empirical Models in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[22] S. Sabour, C. V Nov, and G. E. Hinton, “Dynamic Routing Between Capsules,” arXiv:1710.09829, Nips, 2017, Advances in Neural Information Processing Systems 30, pp- 3856--3866.

[23] G. Hinton, S. Sabour, and N. Frosst, “MATRIX CAPSULES WITH EM ROUTING,” International Conference on Learning Representations (ICLR), pp. 1–15, 2018.

[24] Srivastava, Saurabh and Khurana, Prerna and Tewari, Vartika, "Identifying Aggression and Toxicity in Comments using Capsule Network", Proceedings of the First Workshop on Trolling, Aggression, and Cyberbullying (TRAC-2018), pp. 98—105, 2018.

[25] B. Van Merri and C. S. Fellow, “Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation,” In Proceedings of the 2014 Conference on Empirical Models in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 1724–1734, October 2014.

[26] J. Chung, “Gated Recurrent Neural Networks on Sequence Modeling," arXiv : 1412. 3555v1 [cs . NE], pp. 1–9, December 2014.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. 15, 1 (January 2014), 1929-1958.

[28] Lei Zheng, Vahid Noroozi, and Philip S. Yu, "Joint Deep Modeling of Users and Items Using Reviews for Recommendation," In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17). ACM, New York, NY, USA, 425-434, 2017. DOI: https://doi.org/10.1145/3018661.3018665.

[29] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer 42, 8 (August 2009), 30-37.

DOI=10.1109/MC.2009.263 http://dx.doi.org/10.1109/MC.2009.263

[30] Y. Wang, A. Sun, and J. Han, “Sentiment Analysis by Capsules,” WWW '18 Proceedings of the 2018 World Wide Web Conference, Pages 1165-1174 vol. 2, 2018.

[31] Hinton G.E., Krizhevsky A., Wang S.D. (2011) "Transforming Auto-Encoders," In: Honkela T., Duch W., Girolami M., Kaski S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2011. Lecture Notes in Computer Science, vol 6791. Springer, Berlin, Heidelberg

[32] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, “Collaborative personalized tweet recommendation,” *Proc. 35th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR ’12*, no. 800, p. 661, 2012.

[33] W. Wang, G. Zhang, and J. Lu, “Member contribution-based group recommender system,” *Decis. Support Syst.*, vol. 87, pp. 80–93, 2016.

[34] S. Khater, H. G. Elmongui, and D. Graˇ, “Tweets You Like : Personalized Tweets Recommendation based on Dynamic Users Interests,” IEEE, pp. 1–10, 2014.

[35] R. Katarya and Y. Arora, "A Survey of Recommendation Systems in Twitter," 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 2018, pp. 1-5. doi: 10.1109/CIACT.2018.8480264

[36] Johnson, Rie and Zhang, Tong, “Effective Use of Word Order for Text Categorization with Convolutional Neural Networks”, in Proceedings of the 2015 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, pages = 103—112, doi = "10.3115/v1/N15-1011"

[37] Wei Zhao and Jianbo Ye and Min Yang and Zeyang Lei and Suofei Zhang and Zhou Zhao, “Investigating Capsule Networks with Dynamic Routing for Text Classification”, CoRR, abs/1804.00538, 2018

[38] Srivastava, Saurabh and Khurana, Prerna and Tewari, Vartika, "Identifying Aggression and Toxicity in Comments using Capsule Network", Proceedings ofthe First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 98—105, 2018

[39] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu, “Zero-shot User Intent Detection via Capsule Neural Networks”, journals/corr/abs-1809-00385, 2018

LIST OF PUBLICATIONS OF THE CANDIDATE'S WORK

[1] R. Katarya and Y. Arora, "Study on Text Classification using Capsule Networks," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 501-505, doi: 10.1109/ICACCS.2019.8728394

[2] R. Katarya and Y. Arora, "A Survey of Recommendation Systems in Twitter," 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 2018, pp. 1-5. doi: 10.1109/CIACT.2018.848026