Journal of Big Data

RESEARCH Open Access

Web crawling based context aware recommender system using optimized deep recurrent neural network

Venugopal Boppana[1*] and P. Sandhya[2]

*Correspondence:
venugopalboppana@outlook.com
[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai, India
Full list of author information is available at the end of the article

Abstract

Recommendation systems are obtaining more attention in various application fields especially e-commerce, social networks and tourism etc. The top items are recommended based on the ability of recommender system which predict the future preference out of the available items. Because of the internet, the people in the current society has too many options that's why the recommendation system is very essential. The recommendation is achieved by the particular users who predict the ratings for numerous items and recommend those items to other users. Majorly, content and collaborative filtering techniques are employed in typical recommendation systems to find user preferences and provide final recommendations. But, these systems commonly lacks to take growing user preferences in various contextual factors. Context aware recommendation systems consider various contextual parameters into account and attempt to catch user preferences appropriately. The majority of the work in the recommender system domain focuses on increasing the recommendation accuracy by employing several proposed approaches where the main motive remains to maximize the accuracy of recommendations while ignoring other design objectives, such as a user's an item's context. Therefore, in this paper an effective deep learning based context aware recommendation model is proposed which can be act as an efficient recommender system by showing minimum error during recommendation. Initially, the dataset is pre-processed using Natural Language Tool Kit (NLTK) in Python platform. After pre-processing, the TF–IDF and word embedding model is used for every preprocessed reviews to extract the features and contextual information. The extracted feature is considered as an input of density based clustering to group the negative, neutral and positive sentiments of user reviews. Finally, deep recurrent neural Network (DRNN) is employed to get the most preferable user from every cluster. The recurrent neural network model parameter values are initialized through the fitness computation of Bald Eagle Search (BES) algorithm. The proposed model is implemented using NYC Restaurant Rich Dataset using Python programming platform and performance is evaluated based on the metrics of accuracy, precision, recall and compared with existing models. The proposed recommendation model achieves 99.6% accuracy which is comparatively higher than other machine learning models.

Keywords: Context aware recommendation, Web crawling, User preference vector, Similarity measure, Deep recurrent neural network

Introduction

Nowadays, recommendation systems (RSs) are more eminent and they are used in different web application areas. The recommendation system is a kind of software tool that provides opinions to the users based on their needs also it is known as an information filtering system. The opinions are like what to purchase, which song to heed, which book to recite, etc. [1]. The information overloading is a difficult problem on the internet because of the explosive growth in the count of existing data and the count of visitors to visit a web page frequently. The common examples for the application areas of recommendation systems are recommending books in Amazon and recommending movies in Netflix [2, 3]. The internet store gets personalized for every customers through recommendation of books from the popular website Amazon.com. Several consumers or consumer groups benefit through different personalized ideas because many of the recommendations are personalized also it delivered a ranked list of items. Based on that ranking, RSs predict apt products and facilities for the users [4].

The most prominent strategies of the recommendation systems are the Content and Collaborative filtering method, Hybrid recommendation, Knowledge-based filtering, Demographic method and, Model-based technique. Some researchers use both the combination of these methods for recommendation systems [5]. In the content-based filtering, the fundamental process relies on consumer descriptions and their needed item. Then a profile is managed representing the items by means of signifying it to the target user who previously adored the same [6]. The Collaborative filtering method is the most renowned method also widely used in products, services and travel recommendations [7]. This is also a common method for designing the recommendation system. It uses a massive volume of data collected from the behaviour of the user in an earlier time and predicts which item the users like most [8].

Hybrid recommendation approach is the combination of two or more recommendation method which is used to enhance the quality of recommendations to overcome the restrictions of outdated recommendation methods and the best example for this method is Netflix [9]. Knowledge-based filtering recommended the items built on either suggestion related to user preferences or particular domain information regarding how items connect to user preferences [10]. In the demographic recommender system, it offers recommendations based on a demographic profile of the user like gender, age, nationality etc. [11]. The model-based approach is a kind of collaborated filtering technique that involves constructing a model relevant to dataset rankings. It is known by extracting the data from the dataset and utilize that as a model to provide recommendations also it possibly deliver the benefits of scalability and speed [7].

Most of the recommender systems overlook the succeeding information and mainly concentrated on the content information. Still, the successive data offers more evidence about the behaviour of the user [12–15]. After many years, web service becomes the standard technology for sharing software, information, and computing resources on numerous amount of web pages. It is a process of recognizing useful services and recommending those services to end users [16]. In web services, the web page recommender is essential for websites. The knowledge representation and integrating the web are the challenging problems to make effective web page recommendation. So many of the recommender system uses web usage and domain knowledge through the semantics [17].

The proposed recommendation model make a decision for visiting a restaurants based on user reviews and contextual information. The convention recommender system gives attractive and relevant recommendation for an active consumer using content based filtering algorithm, collaborative algorithm and hybrid filtering algorithm on the basis of predicted ratings. The recent researches employed the sentiment analysis to predict the user preference based on reviews [18]. The combination of sentiment analysis and aspect categorization also employed for hotel Recommendation System based on online review [19]. At any time, prediction of unseen data is possible if the model is established by the supervised learning algorithm on available data history like user reviews and user item rating.

To achieve this, we suggested a context aware recommender system in this work. This method is more prevalent in recent years and they play an imperative part in intelligent choice making systems. This method provides the information related to most visited places such as restaurants. The goal of this technique is to effectively predict user intention and make recommendation accordingly. Moreover, investigate the behaviour of already visited users and create recommendations based on new user preferences. In this study, we have proposed a context aware recommendation system considering user preference and contextual information for predicting favourite restaurants. The major goals of present paper are given below:

1. To build an effective clustering algorithm to analysing the user review sentiments based on word embedding models.
2. To propose a deep recurrent neural network model for accurate recommendation prediction by the aid of word embedding with contextual features.
3. To improve a deep recurrent neural network model using bald eagle search algorithm for optimal selection of hyper parameters.

The remainder of this paper is structured as follows: the recent related works with problem definition is provided in "Related works" section. "Proposed methodology for graph based recommendation model" section presents the details of proposed context aware recommendation model. "Results and discussion" section illustrates the simulation and the performance of our proposed model, and lastly, "Conclusion" section provides the conclusion of the paper.

Related works

Some of the recent related studies related to recommendation models is discussed below:

A deep learning modified neural network algorithm was proposed by Sasikala and Mary Immaculate Sheela [20] for sentiments analysis of online product review. For forthcoming forecast of online products, a method of Improved Adaptive Neuro-Fuzzy Inference System (IANFS) was presented. Initially, the values of dataset are divided as collaboration based (CLB), Content-based (CB) and Grade-based (GB). Next, by utilizing DLMNN every consequence goes through review analysis (RA) as positive, neutral and negative. For future predictions, the IANFIS performs a weighting factor.

Based on hybrid recommendation algorithm a novel implementation of a product recommendation system was presented by Revathy [21]. Based on the original structure to deliver a visual data organization is the main advantages of this method. To search the products anytime and anyplace this method delivers a simple method. Sentiments, review and ratings are evaluated and characterized as negative and positive sentiments. To avoid fake reviews, the MAC based filtering method can be used. Supermarket can help for get new customers, easy transactions and easy buying. Hybrid References is one of the main system module which help to mitigate the issues of content based recommendations and traditional collaborative filtering models.

By integrating online product Song et al. [22] implemented a prospect theory-based method to rank the products. As indicated by the customers' essential product necessities of alternative products are gotten. Then, depends on multiple criteria the objective values and online scores of alternative products are collected and fused when the standardization method was completed. At last, according to the alternative products they get the alternative products ranking depending upon numerous models. By integrating the objective and subjective information using the richer products information was a new idea in a product information.

Based on contextual information an electronic product recommender system from sentiment analysis was presented by Osman et al. [23]. To make the items prediction the recommendation algorithms mostly depends upon the users rating. For recommender system they present a sentiment based model using contextual information. In electronic product recommendation by using outcomes of RMSE and MAE measures their sentiment-based contextual information model provides improved performance.

Based on the automatic features an ensemble detection method was presented by Hao et al. [24]. At first, the users' behaviors are analysed to collaboratively discover the shilling profiles from multiple views like ratings, user graph and item popularity. Secondly, the stacked denoising auto encoders are used based on the data preprocessed from several views to automatically extract the consumer features with various corruption rates. Moreover, based on principal component analysis the features mined from various views are effectually joined. At last, the weak classifiers are generated according to the features extracted with various corruption rates and then combined to detect attacks.

Wu et al. [25] presented a context aware recommender system using graph convolution machine (GCM). The GCM comprises encoder, decoder and graph convolutional layers, the encoder part creates an embedding vectors based on users, items and contexts. Further, the embedding vector fed into the graph convolutional layers to refine user and item embedding's. The decoder part output the prediction score by taking the embedding's of user, item, and context interactions. Moreover, deep neural network based Neural Collaborative Filtering model is proposed by He et al. [26] for recommendation system. Cheng et al. [27] developed an Aspect–Aware recommendation system based on Latent Factor Model using rating and reviews. The rating score is predicted by an aspect importance, which is relied on the features of targeted items and preferences of targeted user's. Table 1 shows the comparative analysis of above discussed related works.

Table 1 Comparative analysis of related works

References	Technique	Domain	Dataset	Metrics	Context-awareness
Sasikala and Mary Immaculate Sheela [20]	Deep learning modified neural network (DLMNN), Adaptive Neuro-Fuzzy Inferences System (IANFIS)	Ecommerce	Own extracted product online reviews	Precision, Recall, F1-score, accuracy	No
Revathy [21]	Sentiment analysis	Ecommerce	Own extracted product reviews	Precision, Recall	No
Song et al. [22]	Prospect theory and multi criteria decision making	Ecommerce	Own extracted online data	Ranking	No
Osman et al. [23]	Sentiment analysis	Electronic	Amazon data	RMSE and MAE	Yes
Hao et al. [24]	Stacked denoising auto encoders	Ecommerce	Movie lens, Netflix, and Amazon data	Precision, Recall	No
Wu et al. [25]	Graph convolution machine (GCM)	Ecommerce	Yelp and Amazon data	Hit ratio (HR) and normalized discounted cumulative gain (NDCG)	Yes
He et al. [26]	Neural collaborative filtering*	Ecommerce	Movie lens and Pinterst	Hit ratio (HR) and normalized discounted cumulative gain (NDCG)	No
Cheng et al. [27]	Aspect-aware latent factor model (ALFM)	Ecommerce	Amazon and Yelp data	RMSE	No

Core finding of problem and motivation

Recommender systems customized to influence contextual information during the course of final recommendation is referred as context-aware recommender systems. The accessible past studies explores different ways to make recommendations and suggests various methods with respect to users' requirements. The most of the studies in the recommender system field given attention on maximizing the accuracy of recommendation by utilizing different methodologies where the key goal focuses to increase the recommendation accuracy while neglecting additional major requirements, such as the context information related to user's an item's context. The principal problem for a recommender system is to provide fruitful recommendations by utilizing user-item contextual information.

This research paper presents a context aware recommender model by employing by word embedding based contextual feature extraction with effective sentiment clustering and deep learning techniques. It helps to facilitate the recommendation of restaurants with the aid of user centric feedback reviews from online platform. We extract the textual information's keywords form raw web data, which reflects the ideal way for extracting the contextual informative features. It creates information representation model, via word embedding methods. Finally we develop an offline knowledge building and recommendation prediction model using deep machine learning techniques.

Proposed methodology for graph based recommendation model

Recommendation system is a data refining model, which offers users with data, which he/she may be fascinated in. Context aware recommendation systems have fairly a determined technique in analysing users' behaviour in the internet and harvest endorsements based on their favourites. It helps to facilitate to recommend good services via online media. Initially, we adopt Complex Event Processing (CEP) module in proposed recommender system to process multiple streams of continuous data, and identify meaningful attributes. The most commonly used TF–IDF and word embedding model is employed to extract the feature from user reviews along with contextual information. The density based clustering algorithm follows the similarity measures such as Dice's coefficient, Jaro–Winkler distance, Damerau–Levenshtein distance, Cosine similarity and Tversky index for grouping the sentiments user reviews. Finally, a deep recurrent neural network (DRNN) model is developed to select the user preference vector for final recommendation. The process flow of system model is displayed in Fig. 1.

In this paper, a model has been introduced that yields recommendations to consumers considering the contextual details such as working time and location of restaurants given by the consumer apart from reviews. Initially, the dataset is obtained for performing the further process. The dataset is pre-processed by NLTK tool in Python platform. Based on the dataset, we create the weight vector matrix based on TF–IDF model. Further, the weight vector matrix is inputted to DRNN to predict the possible recommendation based on user preferences. In DRNN training phase, pre-visited user feedback review

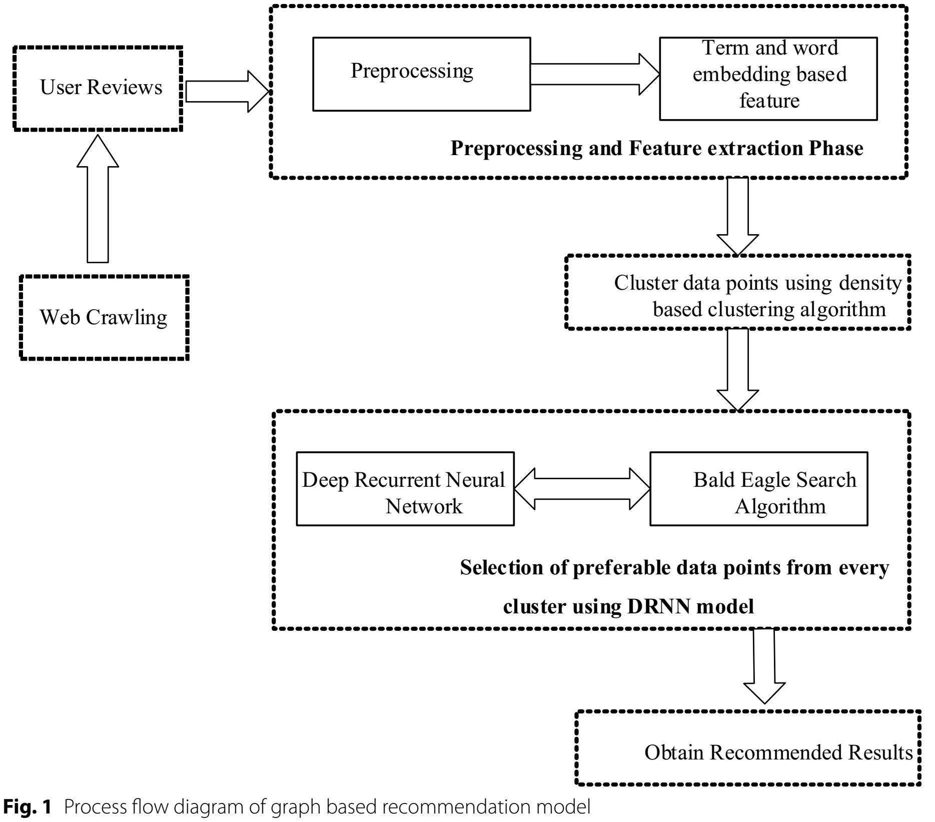
User Reviews
Preprocessing
Term and word embedding based feature
Preprocessing and Feature extraction Phase
Web Crawling
Cluster data points using density based clustering algorithm
Deep Recurrent Neural Network
Bald Eagle Search Algorithm
Selection of preferable data points from every cluster using DRNN model
Obtain Recommended Results

Fig. 1 Process flow diagram of graph based recommendation model

vector score is computed to make the possible recommendation. After the training process, the new user preference vector is generated during the testing phase for final recommendation.

Web crawling

The major intention of web crawler is to extract the information from web sources in customized way. The usage of web crawler is not problematic when obtaining the reviews from the web sources. It is used to save the web sheets also link them in to the local sources because which are the central phase of search engines. In an integrated setting, gathering and analysing the whole contents of web is the common goal of web crawlers. In this work, Beautiful Soup based web Crawling technique is utilized for scraping the data's for websites [28].

Complex event processing (CEP)

On the basis of event pattern or event procedure rule, the CEP is said to be a real-time data process methodology. It is used to retrieve the high level knowledge from the large extent of data. It also utilized to analyse the trends, track the torrents of data from multiple sources, and patterns. In this work, PySiddhi tool is utilized to get the events from data streams, senses complex conditions represented through a Streaming SQL language, and generates possible actions [29].

Pre-processing

Several websites are used to collect the data about web pages. Initially, the pre-processing is done to the datasets. The unwanted noise in the dataset is eliminated by this process also it have direct influence over the calculation of the output. The following difficulties are processed by the pre-processing step: (i) punctuation removal, (ii) removal of stop words like prepositions, articles, etc. and (iii) alteration of upper case letters to lower case letters.

Stop words removal

The English words which doesn't give much meaning to the sentence is termed as stop words. These meaningless words are removed without losing the meaning of the sentence. The examples for stop words are like he, the, have, of, etc. Normalization is described by removing the repeated words. Some words are repeated many time in the web pages or articles. These type of letters not matched with all dictionary words and these are very complex to deal with. For illustration, "super" can be inscribed as superrr, sooper, or suuuppper. The proposed technique declines such occurrences through pre-processing. The row is altered when a letter appears above two times.

Stemming

The procedure of stemming is defined by transforming all the words in a text to their stem or root. The morphological attaches from the words are eliminated by this process. The word is formed by stem of a word in which it is known as root. For example, words "stems", "stemmed", "stemmer", and "stemming" take a mutual stem that is "stem". The different forms of a words are recognized by this stemming and combine those words form together. Without stemming process, various forms of a single word is considered as different words. The terms are converted to their stem with the application of certain heuristics to remove the suffixes.

During stemming process suffix words like 'ed', 'ing', 'ly', 'ment' were eliminated from every user reviews.

Example for stemming word removal on user review:

Input: Amazing food, live entertainment. Great place!
Output: Amaz food, live entertain. Great place!

The outcome gained on stemming phase is visibly represented in above example.

Feature extraction

In recommendation system, feature extraction process plays an important role to reduce the dimensionality of the input data to ensure the prediction accuracy and enhance the time efficiency. In this recommendation model, related features are extracted from the terms returned through the pre-processing phase utilizing a normalized TF–IDF, and word embedding method.

Term based feature extraction

In recommendation system, feature extraction process plays an important role to reduce the dimensionality of the input data to ensure the prediction accuracy and enhance the time efficiency. In this recommendation model, related features are extracted from the terms returned through the pre-processing phase utilizing a normalized TF–IDF method. Normalized TF–IDF is a vector space method that simply extracts the weight of terms numerically. Here, TF–IDF is integrated due to its highly precise performance when compared to other statistical methods. It extracts the features from the high level features by discarding the low level features. This merit makes us to integrate TF–IDF in this proposed framework. Because, the taken input is web related reviews which may contain large number of unwanted data which may reduce the proposed performance. To avoid that, we have used TF–IDF for feature extraction [33]. For every term i, the weight is calculated as follows:

$$W_i = \frac{\left(TF_i \times \log\left(\frac{N}{n_i}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(TF_i \times \log\left(\frac{N}{n_i}\right)\right)^2}}, \tag{1}$$

where n_i is the number of reviews comprising term i and N is the total number of reviews. TF represents the number of occurrence of each term in a review, while IDF denotes the length normalization. The pseudo code for the feature extraction process is illustrated in Algorithm 1. The related-term matrix features acquired at this stage are fed into the fuzzy based clustering.

```
Algorithm 1: TF-IDF based Feature Extraction
Input
    T: the distinct terms in all reviews
    R: the reviews of the training set
Output: weight Matrix
Step:
for each term $t_i \in T$ do
    for each review $r_j \in R$ do
        $w_{ij} = frequecy\ of\ trem\ t_i\ in\ review\ r_j$
    end for
end for
```

Feature extraction based on word embedding

One of the most popular representation of document vocabulary is known as word embedding. The word2vect is utilized for creating the word embedding. The procedure of the word is not understood by a machine this is the reason for using word embedding. The numerical or binary value can be understand by machine. So, to process a language the word2vec must be needed. Each machine can change the tokenize word to a vector after applying word embedding.

One of the most popular technique by using shallow neural network to learn word embedding is known as word2vec. Two methods are utilized in word2vec for word embedding such as continuous bag of word (CBOW) and skip-gram [30]. CBOW is utilized for word context to predict the target word. A word is utilized by skip gram to predict the target value.

CBOW The hidden layer is eliminated and projection shared the same position which gotten by all words. This way of procedure is known as bag of word. On the other side its continuously distributed words thus it named as CBOW.

The mathematical expression for CBOW method is,

$$Q = N \times D + D \times \log_2 (v). \tag{2}$$

Skip-Gram Based on other words the skip-gram method tries to find the words in a same sentence. We utilize the present word with hidden layer projection as an input that may calculates the words within the range.

The formula for skip-gram method is,

$$Q = C \times \left(D + D \times \log_2 (v)\right). \tag{3}$$

where word maximum distance is represented as C.

Context information extraction based on word embedding

The contextual information of user tip is extracted to access the hidden bases of knowledge leads to find out the rich context. The process of contextual feature extraction comprises both bag of words and word embedding model to extract the simple expressions from the collections of user tips and identifying the valuable expressions.

In the word embedding model, the threshold value based comparison is made to extract the context from user tips. Mainly, the similarities among contexts also taken to strengthen the relations among them. For a user tip $T_1 = \{c_1 = \{e_1, e_2, e_3\},\ c_2 = \{e_2, e_4\}\}$, where c_1 and c_2 are the contexts for the expressions e_1 and e_2, correspondingly. If the value of similarity $Val_{S_{1,2}} = \frac{L(c_1 \cap c_2)}{L(c_2)}$ among the contexts c_1 and c_2 is higher than or equal to a threshold value T_v, then the context c_2 is append into the context list of the tip T_1. Likewise, If the value of similarity $Val_{S_{2,1}}$ among the contexts c_2 and c_1 is higher than T_v, then the context c_2 is append into the context list of the tip T_1. The setting of threshold value T_v is user oriented to carrying out the technique.

Density based clustering

In the clustering phase, all the features extracted from user reviews are gathered into various groups based on the similarities. Clustering is termed as an unsupervised learning process that determines the hidden patterns in the instance data; also, it is used to identify the meaningful features for accurate recommendation. Moreover, in statistical machine learning and data mining, the process of clustering is considered as one of the important tasks. In this work, Density based Clustering (DBC) Algorithm is used to label the selected features into positive, neutral and negative reviews. DBC algorithm which used to measure the values of clustering points and also it used to measure the radius of density at minimum number of points. Since it denote the distance function as $d(p, q)$ and it parameters mentioned as *Eps*—neighbourhood points and min *pts*—minimum number of points. The input feature set has measured at two dimension and the dimension are measured as two point which shown in Eq. (4);

$$d(p,q) = \left[\frac{(t_p - t_q)^2}{t_{scale}^2} + \frac{(h_p - h_q)^2}{h_{scale}^2} \right]^{1/2}, \tag{4}$$

where delta time has represents as '*t*', Elevation has represented as '*h*' and t_{scale} and h_{scale} used for normalization method so that can able to compare the dataset points over '*t*' and '*h*' axis. Therefore, the $d(p, q)$ function are unit less thus the heuristic of effective way to determine as two parameters. Here the *Eps* value determine the points at all time so it can modify only min *pts* because that can able to estimate the density of the average points from search space.

The distance between the cluster centroid and the data point is evaluated to update the cluster centroid which is performed using the different similarity calculation techniques such as Dice coefficient, Damerau–Levenshtein distance, Tversky index, Cosine similarity, and Jaro–Winkler distance [31] are defined below;

- Dice's coefficient

This similarity measure used to estimate the similarity of user reviews, and the mathematical formulation for similarity estimation, which is shown in Eq. 5

$$QS = \frac{2|A \cap B|}{|A| + |B|}, \tag{5}$$

where $|A|$ *and* $|B|$ represents the sum of expressions available in reviews. QS represents quotient of similarity.

- Cosine similarity

  In this similarity method, two vectors of an inner product space is calculated to determine cosine of the angle. Cosine of 0° value is 1, for some other angle it is < 1. Majorly, in positive sign cosine similarity is utilized, its outcomes proficiently limited in (0, 1). Cosine similarity [*cos(θ)*] can be computed using the subsequent Eqs. 6 and 7,

$$Similarity = \cos\theta = \left\{ \frac{Dot\ product\ (K, L)}{\|K\| * \|L\|} \right., \tag{6}$$

$$\cos\theta = \frac{\sum_{i=1}^{n} K_i L_i}{\sqrt{\sum_{i=1}^{n} K_i^2 \sqrt{\sum_{i=1}^{n} L_i^2}}}, \tag{7}$$

where K_i and L_i are constituent of vector K and L.

- Tversky index

  It is an asymmetric similarity measure which is obtained by generalizing the dice coefficient and Jaccard index.

  For set, X and Y, the Tversky index lies between 0 and 1.

$$S(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|}, \tag{8}$$

In case, if $\alpha = \beta = 0.5$, it simplifies dice coefficient, and $\alpha = \beta = 1$ it simplifies Jaccard index.

- Jaro–Winkler distance

  This string metric that measures the edit distance among two different sequences are referred as Jaro–Winkler distance.

  For two strings $s1$ and $s2$, the Jaro–Winkler distance is defined in Eq. (9),

$$sim_j = \left\{ \begin{array}{cc} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) & otherwise \end{array} \right., \tag{9}$$

where $|si|$ represents string length, transposition number is denoted as t and the number of matching characters are denoted as m.

- Damerau–Levenshtein distance

  It is also considered as string metric which estimate the edit distance among two different sequences. Informally, the less number of operations (like, substitution, transposition, deletion, and insertion) that are required to convert one word to another is obtained by Damerau–Levenshtein distance.

The distance exist between 2 strings a and b is defined by Damerau–Levenshtein distance using the function $d_{a,b}\left(i,j\right)$. The value obtained using this function is the distance between the string a having i symbol prefix and the string b of j symbol prefix.

Recommendation prediction using deep recurrent neural network

In the DRNN based model, the user tags along with tips are needed to get the list of restaurant venues to be recommended. In this paper, the popular restaurant venues in Foursquare Dataset is recommended by using the deep recurrent neural network. The check-in, tip and tag data are significant for recommending a restaurant.

The connection between the neurons in recurrent neural network (RNN) forms a directed circle. The RNN keep track of its internal hidden state over the recurrent connections which is differ from the feed forward network. The text and speech are the different types of task and the behaviour of RNN is suitable for processing these particular tasks. The usage of sequential information is the idea behinds this RNN. The RNN have the capability of predict the next word in the sentence given every single word before that word.

The recurrent neural network architecture is illustrated in Fig. 2.

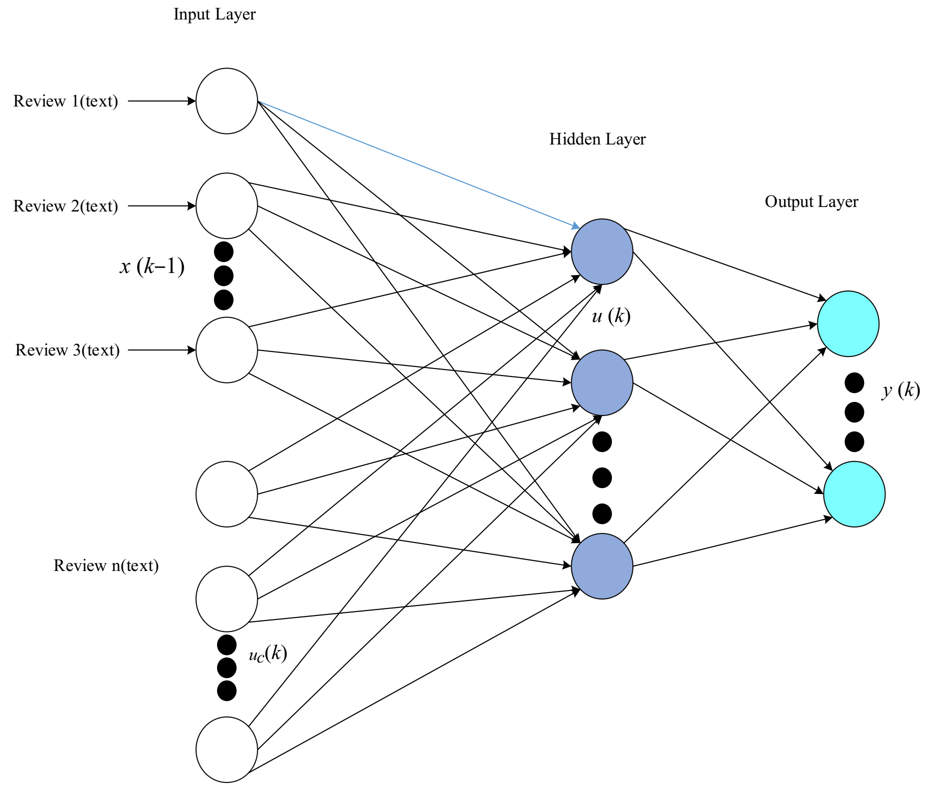

Fig. 2 Recurrent neural network architecture of venue recommendation based on user tags and tips

DRNN based recommendation prediction

The feature extracted from TF–IDF and word2vec is adopted as an input of the DRNN in the proposed model. Due to the different loss function, the BES considered as the popular choice for hyper parameter tuning. The proposed RNN model has three different layer i.e. input, output and hidden layer which is shown in Fig. 3. All the inputs and outputs are independent of each other in traditional neural network. The RNN recursive formula is shown below.

$$h_t = \tanh\left(W_h h_{t-1} + W_x x_t\right), \tag{10}$$

$$y_t = W_y h_t, \tag{11}$$

where weighting matrix is represented as W_h, input vector is represented as x_t, hidden layer vector is represented as h_t and output vector is represented as y_t. Long term dependencies issue is presented in RNN, the issues of weight matrix and long interval time keeps to multiply recurrently with earlier outcomes. This may cause exploding gradient and vanishing gradient issues. To avoid this issue, Long Short Term Memory (LSTM) is used which will enhance the performance.

The structure of LSTM units is used in RNN layer. The RNN with LSTM units is known as LSTM network. The variance among the traditional RNN and LSTM in which every neuron in LSTM is a memory cell. Every neuron includes three gates such as input, forget and output gates.

From the cell, the forget gate $f(t)$ defines which data will be unwanted. At the previous units $t-1$ by entering the output h_{t-1} and add the input x_t with current time t in to sigmoid function $s(t)$.

$$f(t) = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{12}$$

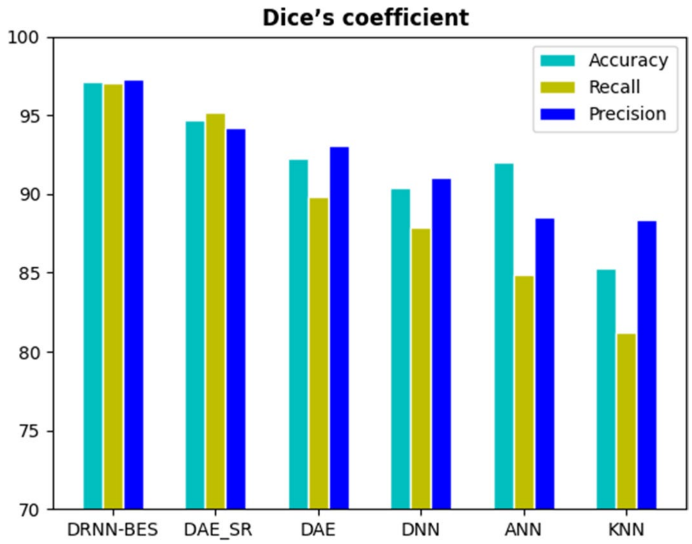

Fig. 3 Performance analysis based on Dice's similarity coefficient

$$S(t) = \frac{1}{1 + e^{-t}}, \tag{13}$$

where the weight matrix is represented as W and bias vector is represented as b.

The gate of input explains which new data to recollect in cell state.

$$i(t) = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{14}$$

$$\tilde{C}(t) = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{15}$$

To get the updated information the values of $i(t)$ and $\tilde{C}(t)$ is multiplied by the sigmoid function that we really want to add the cell state.

$$C(t) = f(t) \times C(t-1) + i(t) \times \tilde{C}(t). \tag{16}$$

The output gate explains which data will be output in the cell state. The cell state is first triggered in the tanh layer before being multiplied by $o(t)$. At time t the multiplication result is the output data $h(t)$ in the block of LSTM.

$$o(t) = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{17}$$

$$h(t) = o(t) \times \tanh(C_t). \tag{18}$$

The available data was categorized into three non-overlapping sets for the purpose of training, testing and validation. The size of training data varies depending on the scenario. At first, we want to train the LSTM model in foursquare location recommendation datasets. To choose the best parameters as well as the performance in the proposed model the validation is employed. Finally the same dataset is utilized to the testing purpose to verify the performance and accuracy. The weight and bias value of all the three gates can be updated by BES [32] optimization. To validate the co-sequences of every phase of hunting is the main behaviour of bald eagle. The hunting behaviour of BES can be classified in to three stage i.e. select, search and swooping stage.

$$Fitness\, f(t) = \max \sum \frac{W_i(t)}{W_{best}}. \tag{19}$$

In the select phase, the BES find and pick the best area as best bias within the chosen search space.

$$b_{new,i} = b_{best} + \alpha * r(b_{mean} - b_i), \tag{20}$$

where random number is denoted by r.

In the search stage, the best position as best weight value for the swoop is mathematically calculated by

$$W_{i,new} = W_i + y(i) * (W_i - W_{i+1}) + x(i) * (W_i - W_{mean}). \tag{21}$$

In the swooping stage, the bald eagle swings from best weight in the search space and best bias in the best area. Both these are calculated and mathematically illustrated as below

$$W, \; b_{i,new} = rand * W, \; b_{best} + x1(i) * (b_i - c1 * P_{mean}) + y1(i) * (W_i - c2 * W_{best}). \tag{22}$$

Based on the above Eq. (23), the weight and bias value will be updated in RNN.

```
Input: Clustering Data
Output: Most preferable cluster form every user
Step 1: Initialize: Input, Output and Hidden Vector
Step 2: Update Weight and Bias by BES
        \\ set the fitness value
        \\ Pick the best Bias using (20)
        \\ Pick the best weight using (21)
        \\ Both Bias and weight are calculated using (22)
        \\ Check the fitness using (19), if the condition is satisfied or not
Step 3: Calculate hidden layer by (10)
        \\ Long term dependencies issues is generated
                \\ Exploding gradient and vanishing gradient
Step 4: Enhance the performance by LSTM
        \\ Calculate input gate, forget gate and output gate
        \\ Calculate the forget gate using (12)
        \\ calculate the input gate using (15)
        \\ Calculate the output gate using (17)
        \\ Update the hidden using (18)
        \\ If satisfied it will stop, or else it will continue
        \\ else
Stop
```

Model training

In this study, an end to end type of LSTM model is employed to explore the process of recommendation prediction. The related model parameter setting is listed in Table 2.

The learning rate is initially set to 0.002, the BES optimization is employed to adjust the hyper parameters during model training. The batch size is set to 8, and the state and hyper parameters in the proposed model are marginally adjusted on the testing process for correct prediction.

Table 2 Parameter setting of proposed model

Parameter	Value
Number of Bi-LSTM layers	3
Input layer size	40
Hidden layer size	80
Output layer size	20
Learning rate	0.002
Batch size	8
Decay factor	0.2
Decay period	5
Optimizer	BES

Results and discussion

In simulation analysis, the outcome of proposed DRNN-BES is compared with the existing KNN based collaborative filtering model and ANN, DNN, DAE, DAE-SR based recommendation models. For validating the performance of proposed model, the data sources are obtained from the Foursquare NYC Restaurant Rich Dataset [34]. The data set comprises 3112 users and 3298 venues with 27,149 check-ins and 10,377 tips. For the purpose of performance evaluation, the dataset comprising 10,377 tips is partitioned into a ratio of 70:20:10 for training, validation, testing respectively i.e., 7264 reviews for training, 2075 for validation, 1038 for testing. The proposed strategy is estimated by means of accuracy, recall and precision. The proposed DRNN based recommendation model simulated on Python programming platform. The performance analysis is made based on the similarity measure, sentiment with context information and varying the training size of data which are presented in upcoming sub sections.

Performance measures

The following are the performance measures that are used in the simulation for the performance analysis. Accuracy, Precision and Recall and are the performance parameters utilised in the experimental results.

- *Precision value* It is indicated for regained document. It is estimated through the division of total count of related documents to total count of resultant documents.

$$Precision = \frac{TP}{TP + FP}. \tag{23}$$

- *Recall value* Related documents associated with the request.

$$Recall = \frac{TP}{TP + FN}. \tag{24}$$

- *Accuracy value* Essential related documents for classification is given by accuracy. The accuracy performance is always in better performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{25}$$

(TP—true positive, TN—true negative, FP—false positive, FN—false negative).

Performance analysis based on similarity measures

The performance analysis of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Dice's coefficient Jaro–Winkler distance, Damerau–Levenshtein distance, Cosine similarity and Tversky index employed in density peak clustering algorithm are compared and explained in this section.

The Table 3 gives the performance value of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Dice's coefficient and Fig. 3 shows the graphical representation of analysis. For KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES classifier precision value are 85.4, 88.5, 91, 89.6, 94.2 and 97.3% recall value are 80.2, 84.9, 87.9,

Table 3 Dice's coefficient based performance comparison

Performance metrics	KNN	ANN	DNN	DAE	DAE-SR	DRNN-BES
Precision	85.4	88.5	91	89.6	94.2	97.3
Recall	80.2	84.9	87.9	90.6	95.2	97.8
Accuracy	85.3	90	90.4	92	94.7	97.3

90.6, 95.2 and 97.8%. The accuracy value for DNN, ANN, KNN, DAE-SR and proposed DRNN-BES classifier are 85.3, 90, 90.4, 92, 94.7 and 97.3%. Compared with the existing techniques our proposed approach has better performance.

The Table 4 and Fig. 4 depicts the performance value and investigation of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Damerau–Levenshtein distance based similarity. For KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES classifier precision value are 76.2, 79.2, 82.5, 83.8, 90.3 and 94.3%, recall value are 79.2, 80.4, 86.4, 88.2, 89.3 and 93.6% also accuracy value are 78.2, 82.4, 84.4, 86.4, 90.7 and 94%. It is obviously agreed, compared with other classifier the proposed approach has better performance.

The Table 5 gives the performance value of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Tversky index and Fig. 5 shows the graphical representation of analysis. For KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES classifier

Table 4 Damerau–Levenshtein distance based performance comparison

Performance metrics	KNN	ANN	DNN	DAE	DAE-SR	DRNN-BES
Precision	76.2	79.2	82.5	83.8	90.3	94.3
Recall	79.2	80.4	86.4	88.2	89.3	93.6
Accuracy	78.2	82.4	84.4	86.4	90.7	94

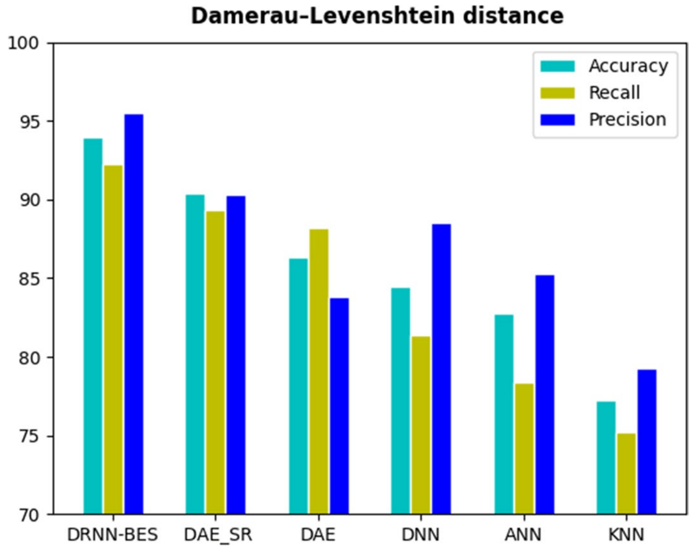

Fig. 4 Performance analysis based on Damerau–Levenshtein distance

Table 5 Tversky index based performance comparison

Performance metrics	KNN	ANN	DNN	DAE	DAE-SR	DRNN-BES
Precision	90.9	93.5	96.8	97.2	97	99.5
Recall	92.8	94.2	95.7	92.8	98.8	99.3
Accuracy	93.8	95.9	97.6	98.1	99	99.4

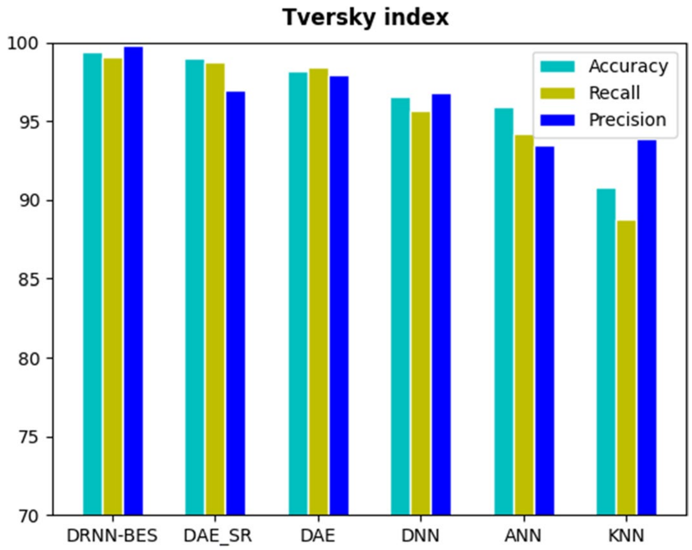

Fig. 5 Performance analysis based on Tversky index

Table 6 Cosine similarity based performance comparison

Performance metrics	KNN	ANN	DNN	DAE	DAE-SR	DRNN-BES
Precision	90	92	94	94	98.2	98.6
Recall	85	88	92	93	98.5	98.9
Accuracy	89	93	96	95	98.3	98.7

precision value are 90.9, 93.5, 96.8, 97.2, 97 and 99.5% recall value are 92.8, 94.2, 95.7, 92.8, 98.8 and 99.3. The accuracy value for DNN, ANN, KNN, DAE-SR and proposed DRNN-BES classifier are 93.8, 95.9, 97.6, 98.1, 99 and 99.4%. Compared with the existing techniques our proposed approach has better performance.

The Table 6 and Fig. 6 depicts the performance value and investigation of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Cosine similarity. For KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES classifier precision value are 90, 92, 94, 94, 98.2 and 98.6%, recall value are 85, 88, 92, 93, 98.5 and 98.9% also accuracy value are 89, 93, 96, 95, 98.3 and 98.7%. It is obviously agreed, compared with other classifier the proposed approach has better performance.

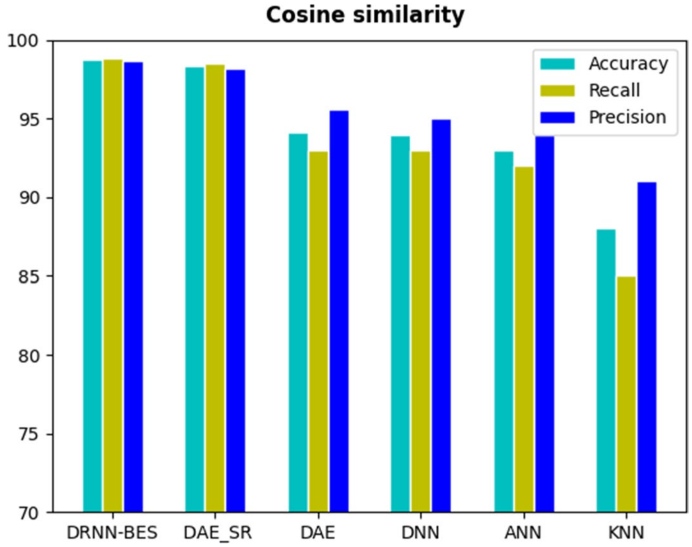

Fig. 6 Performance analysis based on cosine similarity

Table 7 Jaro–Winkler distance based performance comparison

Performance metrics	KNN	ANN	DNN	DAE	DAE-SR	DRNN-BES
Precision	79.2	80.2	88.4	87.3	91.4	95.7
Recall	80.3	83.6	85.9	88.2	90.3	95.8
Accuracy	81.2	85.5	87.2	90.3	93.6	95.8

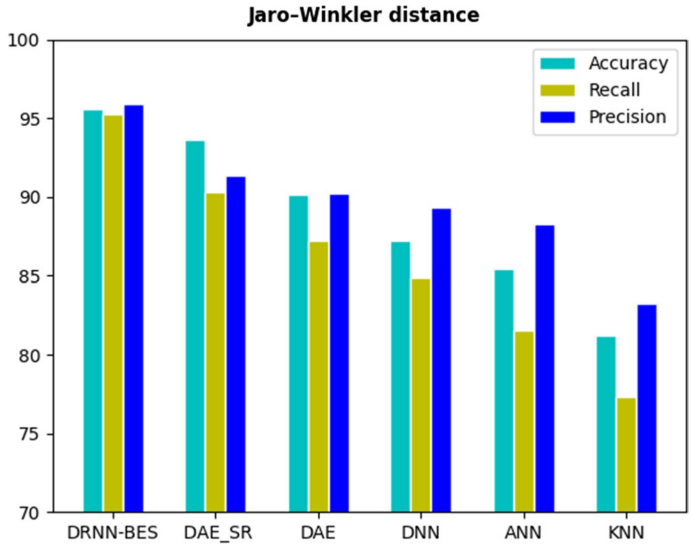

Fig. 7 Performance analysis based on Jaro–Winkler distance

Table 8 Evaluation of accuracy using various feature representations

Size of training data	Feature representation	DNN	DAE-SR	DAE	DRNN
Complete (100%)	TF–IDF	92.5	93.2	94.6	95.2
	TF–IDF + CBOW	95.2	96.8	97.5	98.3
	TF–IDF + CBOW + Skip-gram	96	97.3	98.4	99.5
80%	TF–IDF	91.3	92.7	93	94.3
	TF–IDF + CBOW	92.8	94	95.8	97.6
	TF–IDF + CBOW + Skip-gram	95.2	96.2	97.2	98.2
60%	TF–IDF	90.2	91.3	92.1	93.2
	TF–IDF + CBOW	92.3	93	94.2	95.8
	TF–IDF + CBOW + Skip-gram	93	94.3	95.4	97.2
40%	TF–IDF	90.2	90.8	91.4	92
	TF–IDF + CBOW	91.3	92.2	93.6	94.8
	TF–IDF + CBOW + Skip-gram	93.5	94.6	95.8	96

Table 9 Evaluation of accuracy using various sentiment with context information

Sentiment	Context	DNN	DAE-SR	DAE	DRNN
Negative	Without context information	90.5	92.2	93.6	94.2
	With context information	91.2	92.8	94.5	95.3
Neutral	Without context information	93.3	93.7	95	96.3
	With context information	94.8	94	96.8	97.8
Positive	Without context information	95.2	96.3	97.1	98.2
	With context information	96.3	97.4	98.5	99.6

The Table 7 gives the performance value of KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES for Tversky index and Fig. 7 shows the graphical representation of analysis. For KNN, ANN, DNN, DAE, DAE-SR and proposed DRNN-BES classifier precision value are 79.2, 80.2, 88.4, 87.3, 91.4 and 95.7% recall value are 80.3, 83.6, 85.9, 88.2, 90.3 and 95.8%. The accuracy value for DNN, ANN, KNN, DAE-SR and proposed DRNN-BES classifier are 81.2, 85.5, 87.2, 90.3, 93.6 and 95.8%. Compared with the existing techniques our proposed approach has better performance. From the above results, Tversky index based performance are is high when compared tom the other similarity.

Performance evaluation based on sentiment contextual information and training data size

In context aware recommendation models, training data size gains the notable importance. Simulations have been conducted with different training data size. The efficiency of proposed recommendation model gets increases while increasing the size of training data. Table 8 displays the performance of proposed DRNN along with DAE-SR and DAE using different feature representations on changing the size training data. Here firstly, a complete training data (100% of the total data) is utilized for model training by different feature representations. Additionally, 20% trimming is performed to each of the training data files and recurrent the same simulations. The evaluation obtained from the table values shows that the proposed DRNN model achieves higher accuracy up to 99.5% compared to DAE-SR and DAE based model. Table 9 shows the accuracy performance comparison using various sentiment with context information. From the analysis of table values conclude that the accuracy performance is improved while adding the contextual information. Moreover, the comparison is made with respect to training, validation, testing accuracies among the proposed DRNN model with other models is displayed in Table 10.

As in our proposed model, the Cosine similarity is employed to determine the contextual similarity of terms which leads to provide fair recommendation. Moreover, the proposed DRNN has hybrid with BES, an optimization algorithm. This hybrid architecture has improved the overall performance of proposed architecture. Normally, different optimization algorithms are now available but we have selected this algorithm to hybrid with BES this is because the proposed BES has attained efficient solution identification that other optimization algorithms. Due to this reason, we have hybrid BES with DRNN and attained efficient result than other existing algorithms. Existing model does not include any optimization algorithms for optimal parameter selection, but in our work we have combined BES with DRNN to develop efficient recommendation system.

ANOVA test for statistical validation of proposed model

The DRNN based proposed recommendation model is qualitatively validated by accuracy and recall measures which are opposed each other. The statistical implication of DRNN model is examined using familiar statistical validation technique named as analysis of variance (ANOVA) model is employed. The result of the ANOVA test is evaluated with DAE and DAE-SR based recommendation model to reveal the statistical implication of DRNN model using the input metrics of accuracy and recall. Generally, the null hypothesis (*Hnull*) of an ANOVA test denotes that the mean value of two or more methods for the designated set of values are identical and so disapprove the null hypothesis assumption. The f-statistic measure is employed to provide the outcome of an ANOVA test. After fulfilling the condition of two assumptions discussed below, the *Hnull* becomes disapproved.

i. The p-value < level of importance.
ii. Value of *F-statistic* value > value of *F-critical.*

Similarly, the alternative hypothesis represented as *Halt* defined in Eqn. 26 to pledge the null hypothesis *Hnull.*

$$Hnull: \mu DRNN = \mu DAE - SR \; = \mu DAE, \tag{26}$$

$$Halt: \mu DRNN \neq \mu DAE - SR \neq \; \mu DAE. \tag{27}$$

Table 10 Evaluation of accuracy with respect to training, validation and testing

Model	Training	Validation	Testing
DNN	94.5	90.2	96
DAE-SR	96.2	92.8	97.3
DAE	97	94.3	98.4
DRNN	99.6	95.2	99.5

Table 11 Summary of input/output values based on accuracy

(a) Summary of input values based on accuracy

Methods	Trials (N)	Mean	Std. Dev.	Std. Error
DRNN	5	0.34	0.1294	0.0579
DAE-SR	5	3	0.7906	0.3536
DAE	5	3.18	0.2864	0.1281

(b) Summary of output values based on accuracy

Source	Degree of freedom	Sum of squares	Mean square
Among groups	2	25.2893	12.6447
Within groups	12	2.8953	0.2413
Total	14	28.1846	

Table 12 Summary of input/output values based on Recall

(a) Summary of input values based on Recall

Models	Trials (N)	Mean	Std. Dev.	Std. Error
DRNN	5	97.05	0.7984	0.3571
DAE-SR	5	94.5	0.3953	0.1768
DAE	5	92.6	0.5184	0.2318

(b) Summary of output values based on Recall

Source	Degree of freedom	Sum of squares	Mean square
Among groups	2	49.8583	24.9292
Within groups	12	4.2498	0.3541
Total	14	54.1081	

Especially, five trails has been taken from all models by employing number of iterations to conduct an ANOVA test. Additionally, other measures such as level of importance α = 0.05 and confidence interval (CI) range = 95% are considered. Tables 11(a), (b), and 12(a), (b) have display the input selected for executing the ANOVA test based on accuracy and recall metrics to examine the output value in the form of f-ratio and p-value. The valuation of the test outcomes shown in Tables 11(b), 12(b), it can be confirmed that the alteration in the mean value of error has statistically valid, therefore the null hypothesis H_{null} is disapproved and approved the alternative hypothesis H_{alt}. Additionally, in the ANOVA test for accuracy, the value of f-ratio is 52.4082. The p-value is 0. Therefore, the ANOVA test outcome at $p < 0.05$ is valid.

The f-ratio is 52.4082. The p-value is 0. The outcome at $p < 0.05$ is valid.

The f-ratio is 70.392. The p-value is 0. The outcome is not-valid at $p < 0.05$.

Conclusion

In this paper, an effective context aware recommendation model is proposed. Initially, the consumer feedback comments are extracted from the online amenities via web crawling technique. In the beginning of the recommendation system, pre-processing is carried out to remove the irrelevant words from the user reviews. After pre-processing, TF–IDF vector model is employed to extract relevant features numerically from the feedback user tips. Further, word embedding model is employed to extract the contextual information from user tips. Then, the density based clustering algorithm is executed to group similar sentiments of user tips. Finally, the deep recurrent neural network model is employed to select the possible user preference vectors from clusters. The comparative analysis performed based on similarity measures, training data size and sentiment based contextual information using this metrics the metrics of accuracy, precision, recall. Our proposed model achieves accuracy up to 99.6 with the inclusion of contextual information and outperforms compared to other deep learning model. In future, the aspect based opinions need to be considered to achieve fair recommendation with different domain datasets.

Abbreviations

NLTK: Natural Language Tool Kit; TF–IDF: Term Frequency–Inverse Document Frequency; RS: Recommendation systems; IANFS: Improved Adaptive Neuro-Fuzzy Inference System; CLB: Collaboration based; CB: Content-based; GB: Grade-based; DLMNN: Deep learning modified neural network; RA: Review analysis; CEP: Complex Event Processing; RNN: Recurrent neural network; LSTM: Long Short Term Memory; KNN: K-nearest neighbours; ANN: Artificial neural networks; DNN: Deep neural network; DAE: Denoising Auto encoder; DAE-SR: Denoising Auto encoder-Super Resolution; TP: True positive; TN: True negative; FP: False positive; FN: False negative.

Acknowledgements

Not applicable.

Authors' contributions

VB has found the proposed algorithms and obtained the datasets for the research and explored different methods discussed. SP contributed to the modification of study objectives and framework. Their rich experience was instrumental in improving our work. All authors contributed to the editing and proofreading. Both authors read and approved the final manuscript.

Funding

Authors did not receive any funding for this study.

Availability of data and materials

We have collected NYC Restaurant Rich datasets which is available publicly.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai, India. [2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai, India.

Received: 15 June 2021 Accepted: 31 October 2021
Published online: 20 November 2021

References

1. Thorat PB, Goudar RM, Barve S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. Int J Comput Appl. 2015;110(4):31–6.
2. Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: principles, methods and evaluation. Egypt Inform J. 2015;16(3):261–73.
3. Tarus JK, Niu Z, Yousif A. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. Future Gener Comput Syst. 2017;72:37–48.

4. Ricci F, Rokach L, Shapira B. Recommender systems: introduction and challenges. In: Recommender systems handbook. New York: Springer; 2015. p. 1–34.
5. Chen L, Chen G, Wang F. Recommender systems based on user reviews: the state of the art. User Model User Adapt Interact. 2015;25(2):99–154.
6. De Gemmis M, Lops P, Musto C, Narducci F, Semeraro G. Semantics-aware content-based recommender systems. In: Recommender systems handbook. New York: Springer; 2015. p. 119–59.
7. Jiang S, Qian X, Shen J, Fu Y, Mei T. Author topic model-based collaborative filtering for personalized POI recommendations. IEEE Trans Multimed. 2015;17(6):907–18.
8. Wei J, He J, Chen K, Zhou Y, Tang Z. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst Appl. 2017;69:29–39.
9. Al-Hassan M, Lu H, Lu J. A semantic enhanced hybrid recommendation approach: a case study of e-Government tourism service recommendation system. Decis Support Syst. 2015;72:97–109.
10. Colombo-Mendoza LO, Valencia-García R, Rodríguez-González A, Alor-Hernández G, Samper-Zapater JJ. RecomMetz: a context-aware knowledge-based mobile recommender system for movie showtimes. Expert Syst Appl. 2015;42(3):1202–22.
11. Al-Shamri MY. User profiling approaches for demographic recommender systems. Knowl Based Syst. 2016;100:175–87.
12. Amara S, Subramanian RR. Collaborating personalized recommender system and content-based recommender system using TextCorpus. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). IEEE. p. 105–9.
13. Choudhury SS, Mohanty SN, Jagadev AK. Multimodal trust based recommender system with machine learning approaches for movie recommendation. Int J Inf Technol. 2021;13(2):475–82.
14. Wang D, Liang Y, Xu D, Feng X, Guan R. A content-based recommender system for computer science publications. Knowl Based Syst. 2018;1(157):1–9.
15. He R, Li Q, Ai B, Geng YL, Molisch AF, Kristem V, Zhong Z, Yu J. A kernel-power-density-based algorithm for channel multipath components clustering. IEEE Trans Wirel Commun. 2017;16(11):7138–51.
16. Yao L, Sheng QZ, Ngu AH, Yu J, Segev A. Unified collaborative and content-based web service recommendation. IEEE Trans Serv Comput. 2015;8(3):453–66.
17. Hu Y, Peng Q, Hu X, Yang R. Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. IEEE Trans Serv Comput. 2015;8(5):782–94.
18. Asani E, Vahdat-Nejad H, Sadri J. Restaurant recommender system based on sentiment analysis. Mach Learn Appl. 2021;6:100114.
19. Ray B, Garain A, Sarkar R. An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. Appl Soft Comput. 2021;98:106935.
20. Sasikala P, Mary Immaculate Sheela L. Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS. J Big Data. 2020;7:1–20.
21. Revathy R. A hybrid approach for product reviews using sentiment analysis. Adalya J. 2020;9(2):340–3.
22. Song Y, Li G, Ergu D. Recommending products by fusing online product scores and objective information based on prospect theory. IEEE Access. 2020;8:58995–9006.
23. Osman NA, Noah SAM, Darwich M. Contextual sentiment based recommender system to provide recommendation in the electronic products domain. Int J Mach Learn Comput. 2019;9(4):425–31.
24. Hao Y, Zhang F, Wang J, Zhao Q, Cao J. Detecting shilling attacks with automatic features from multiple views. Secur Commun Netw. 2019. https://doi.org/10.1155/2019/6523183.
25. Wu J, He X, Wang X, Wang Q, Chen W, Lian J, Xie X. Graph convolution machine for context-aware recommender system. arXiv preprint arXiv:2001.11402. 2020.
26. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS. Neural collaborative filtering. In: Proceedings of the 26th international conference on World Wide Web. 2017. p. 173–82.
27. Cheng Z, Ding Y, Zhu L, Kankanhalli M. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In: Proceedings of the 2018 World Wide Web conference. 2018. p. 639–48.
28. Zheng C, He G, Peng Z. A study of web information extraction technology based on beautiful soup. J Comput. 2015;10(6):381–7.
29. https://siddhi-io.github.io/PySiddhi/.
30. Deepak G, Shwetha BN, Pushpa CN, Thriveni J, Venugopal KR. A hybridized semantic trust-based framework for personalized web page recommendation. Int J Comput Appl. 2018;14:1–1.
31. Wu C, Wu J, Luo C, Wu Q, Liu C, Wu Y, Yang F. Recommendation algorithm based on user score probability and project type. EURASIP J Wirel Commun Netw. 2019;2019(1):80.
32. Alsattar HA, Zaidan AA, Zaidan BB. Novel meta-heuristic bald eagle search optimisation algorithm. Artif Intell Rev. 2020;53(3):2237–64.
33. Garg S. Drug recommendation system based on sentiment analysis of drug reviews using machine learning. In: 2021 11th international conference on cloud computing, data science & engineering (confluence). IEEE; 2021. p. 175–181.
34. https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

Publisher's Note