Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 4589060, 19 pages
https://doi.org/10.1155/2019/4589060

Research Article

A Neural Network-Inspired Approach for Improved and True Movie Recommendations

Muhammad Ibrahim,[1] Imran Sarwar Bajwa ,[1] Riaz Ul-Amin,[2] and Bakhtiar Kasi[2]

[1]*Department of Computer Science & IT, Islamia University of Bahawalpur, Bahawalpur, Pakistan*
[2]*Faculty of Information and Communication Technologies, BUITEMS, Quetta, Pakistan*

Correspondence should be addressed to Imran Sarwar Bajwa; imran.sarwar@iub.edu.pk

Received 15 April 2019; Revised 28 May 2019; Accepted 17 June 2019; Published 4 August 2019

Guest Editor: Ricardo Soto

In the last decade, sentiment analysis, opinion mining, and subjectivity of microblogs in social media have attracted a great deal of attention of researchers. Movie recommendation systems are the tools, which provide valuable services to the users. The data available online are growing gradually because the online activities of users or viewers are increasing day by day. Because of this, big data, analytics, and computational issues have raised. Therefore, we have to improve recommendations services upon the traditional one to make the recommendation system significant and efficient. This article presents the solution for these issues by producing the significant and efficient recommendation services using multivariates (ratings, votes, Twitter likes, and reviews) of movies from multiple external resources which are fetched by the web bot and managed by the Apache Hadoop framework in a distributed manner. Reviews are analyzed by a deep semantic analyzer based on the recurrent neural network (RNN/LSTM attention) with user movie attention (UMA) to produce the emotion. The proposed recommender evaluates multivariates and produces a more significant movie recommendation list according to the taste of the user on a mobile app in an efficient way.

1. Introduction

"Recommendation systems" are services that use Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques to provide the empirical solutions of the recommendations for various application frameworks and services [1]. Recommendation systems enables mobile apps and web applications to make the perception intelligently about the selection of different items, movies [2], hotels [3], food [4], tourism [5], books [6], TV shows [7], YouTube videos [8], health [9], etc. Community trends polarize towards music, movies, or videos. For music or movies or videos, a huge amount of stream is available online, but which one of them will be watched is still a rising question. Music or movie recommendation systems still have challenges like the playlist, magnitude, security, privacy, recommendation, and session. Therefore, MRSs become a domain of music information retrieval (MIR) [10–13]. Now, the society has changed, and community trends highly depend on mobile app usage. Several products are enriched by the usage of a mobile app. So mobile app recommendation systems are essential for suitable selection of recommended items [14–16]. Most of the recommender systems are univariate and use ratings and reviews or tweets [17], and other few are bivariate (sentiment score and likes) [18–20]. This work is state of the art and uses the multivariate matrix, which makes the decision using a dynamic approach for suggesting the movie according to the relative taste of the users. The term "multivariate" means involving many variables like a qualitative variable (semantic score) and quantitative variables (Twitter likes, rating, and votes) of movies from three movie sites for significant recommendation [21]. Our work is on extremity grouping of movie reviews, where an opinionated report is labeled with semantic emotions of the microblog text or reviews and emotions [22] using a semantic parser based on the recurrent neural network (RNN/LSTM) [23, 24]. A drawback is that change of a user's review about a movie may affect the user's preference. The nature of reviews influenced by the choice of words uses multilingual dictionaries. Some

recommendation systems use linked movie databases, including Trovacinema, Google Places, and Netflix, and Wikipedia provides linked data and ontologies for descriptions about the movie [25–27]. Using the shallow machine learning models for solving the NLP problems is handcrafted and time-consuming. Nowadays, word embedding, neural-based models achieve success and popularity by producing a better result as compared to traditional machine learning logistic regression, SVM, and KNN.

Artificial neural networks are the mathematical models that are inspired by human neural networks. They have three simple layers: input, output, and hidden layers, or sometimes only two layers: input and output layers. The input layer is connected to the hidden layer via a lean weight. The hidden layer output combines via the activation function $h = \phi(w_i \cdot x_i)$. In the ANN, like the biological neural network, neurons are the nodes, while synopses are the edges. Each artificial neuron has an activation function in the ANN. There are several activation functions like sigmoid which ranges from 0 to 1, hyperbolic function which ranges from −1 to 1, and softmax function whose output in categorical distribution and ReLu function is a feedforward neural network. The ANN is not an algorithm; it is a framework for several machine learning algorithms to solve a complex work. Therefore, we can say that it is a collection of neurons or networks of neurons (https://en.wikipedia.org/wiki/Artificial_neural_network). The recurrent neural network (RNN/LSTM) processes the sequence semantically, which is the basic structure of deep neural networks. Several NLP tasks are performed by RNNs/LSTM attention. In this work, we used the hierarchical neural network (HNN) based on LSTM attention, which impaled the global user and movie information via word and sentence-level attention for document representation. The user's reviews and movie features at the word and sentence level are taken for semantic analysis of reviews, which play a major role in the process of true recommendations. Global user information represents the personal behavior and the movie feature represents a movie genre or a movie profile or linked data which are useful for semantic extraction of movie reviews [28]. In natural language (word sequence), each word or sentence is related to another one and requires to be understood semantically. A huge amount of data are available online on web contents (ratings, reviews, likes, votes, smiley, images, and stars) that can be fetched by a web bot or web agent or crawler, which are all same terms used interchangeably. Web content (ratings, reviews, likes, votes, smiley, images, and stars) is useful for recommendation services. These contents are evaluated and make the perception about users, and items make the recommendation for others [29, 30]. The hot issues of big data like computational complexity are managed by using Map-Reduce and Apache Mahout in NoSQL [31, 32] distributed environment which reduce computation complexity by clustering and horizontal scaling instead of empowered single machine [33]. Because user frequency and data volume gradually increase, it is difficult to manage these huge data by a single machine. Sparsity can be reduced by factorization [34]. Movie recommendation systems provide services to users using content-based filtering algorithms [35], collaborative filtering [36], and some combined forms to make a hybrid filtering algorithm [37]. We used implicate rating to handle the cold start problem [38], an implication managed by the server The multivariate movie recommender provides the services to users to watch the movies according to their profile or history (previously watched or rated). Therefore, there is a need to improve recommendation systems for significant recommendation services. We developed a pilot version for these problems, which consists of a mobile app, a web scraper, and a multivariate recommender to provide the significant services for movie recommendation in an efficient way.

This work is arranged as follows: related works are discussed in Section 2, the recurrent multivariate movie recommendation system model is explained in Section 3, recurrent multivariate movie recommendation system implementation is given in Section 4, experiments and results are discussed in Section 5, and evaluation of the system is done in Section 6. The conclusion of this paper is presented in Section 7 and future work with more parameters in Section 8.

2. Literature Review

Sentiment analysis deals with the user's comments, reviews, likeness, ratings, etc. to retrieve the sentiment and opinions of users. The microblog text sentiment analysis is based on the NLP methodology to retrieve suitable YouTube videos and movies and campaigns for smoking cessation, pharmacovigilance, politics of elections, advertisement of pizza, journalistic inquiry, and influenza prevention for public health [39–45]. The CNN and RNN are two major categories of deep neural networks (DNNs). Sequential and hierarchal structures deal with the RNN and CNN, respectively. Both the CNN and RNN can be supervised, semisupervised, and unsupervised. The deep learning algorithm also involves in propagation and weight update activities. RNNs are based on multiple layers: input, hidden, and output layers, while CNNs have input, hidden, and pooling layers. The CNN is efficient for pattern recognition in hierarchal data classification. However, the RNN deals with linear data to be semantically analyzed and classified in NLP; in the CNN, the window size is limited, so the RNN is very useful if reviews from the microblog are very large [46, 47]. Recommendation frameworks were presented as agents of the second class, being characterized as frameworks that "... enable individuals to settle on decisions dependent on the conclusions of other individuals." [48]. Early data-sharing frameworks had a place with the primary class and depended on text-based classification or separation, which works by choosing important things as per many literary catchphrases [49]. Recommender frameworks propose "things important to clients dependent on their unequivocal and verifiable inclinations, the inclinations of different clients, and client and thing traits." [50]. The recommendation system is finding the right product according to the taste of the customer by filtering the fact through the likeness value [51]. Suggestions utilize the assessments of a community of clients to help

people in that community all the more adequately distinguish the content of enthusiasm from a possibly overpowering set of decisions [52]. Recommendation by demographics which groups the users as per the traits of their personnel file, besides, creates proposals dependent on classes of the statistic. A premature precedent is a generalization-based Grundy system, which has been made to bolster book searching in a library [53]. The recommendation is reliant on the computation of utility of each item for a user' utility capacity (http://www.eqo.info). Recommendation by knowledge proposes things dependent on legitimate inductions about a user's inclinations. A learning portrayal or a rule about how a thing meets a specific client requirement is important (http://www.findme.com.ph). By applying preference-based collaborative filtering, a recommender system intend to foresee majority of estimation of likeness, where a few users may provide inconspicuous views as well [54]. There are two types of architecture for the recommendation systems: One is centralized and situated at a specific location [55]. Another one is geographically distributed and situated at different locations [56]. There are three types of recommendation modes by which the system will be initiated: The first one is the push mode in which suggestions are pushed to the user while he is not associating with the system by email [48]. The second one is the pull mode in which suggestions are generated but are displayed to the user just when he permits or unequivocally asks for it [57]. Push and pull modes are the active mode in which the recommender is initiated. The third one is the passive mode in which suggestions are generated as a feature of the customary framework activity, for instance, an item suggestion with reference to a user's preference [58]. A user's preference of items can be determined by using the linear adaptive function multiattribute utility theory (MAUT) [59]. Cosine similarity determined by cosine vector comparability is one of the well-known measurements of insight since it notionally considers just the edge of two vectors without the size. The collaboration between the search item and the other item that is rated by users can be measured by the angle of their vectors; if the angle is 90°, then the value of cosine similarity is zero, which means the item is irrelevant. If the angle between cosine vectors is nearly about zero, then the value of cosine similarity is one, which means the product is relevant (https://en.wikipedia.org/wiki/Cosine_similarity) [60]. There are three major classes of collaborative filtering: (1) collaborative filtering (CF) in which users and items' profile data are required to make a decision for recommendation [61], (2) content-based filtering on the description of the content of items and user preference information (explicate or implicate) for recommendation [62], and (3) combining various filtering techniques to handle scalability, sparsity, and cold start problem and other big data issues of the recommendation system to get better outcomes [63].

3. Multivariate Movie Recommendation Model

The multivariate approach is (see Figure 1) based on three modules: mobile app, multivariate recommender, and web scraper. Users can get the recommendation services through a mobile application. The mobile app module provides the information such as the user's query, profile, and history to the recommender module. The recommendation is made for both registered and unregistered users of the mobile app. The recommendation module is based on the deep learning NLP module and computation module. The NLP module preprocesses the fetched qualitative data (user's reviews) of microblogs using a tokenizer, stemmer, and POStagger and then semantically analyzes the reviews and extracts the semantic emotions about movies. Semantic parser work is based on the deep machine learning algorithm recurrent neural network (RNN/LSTM attention) with user movie attention (UMA). Semantic emotion is classified into five major classes: (i) Highly Favorable, (ii) Favorable, (iii) Averagely Favorable, (iv) Unfavorable, and (v) Highly Unfavorable, on the bases of their relative semantic scores. While the computation module normalized the quantitative data (Twitter likes, votes, and ratings), normalized scores and semantic emotional scores were evaluated to generate the recommended movie list. The recommended movie list consists of five medals and their popularity such as Platinum: "Highly Popular," Gold: "Popular," Silver: "Averagely Popular," Bronze: "Unpopular," and Copper: "Highly Unpopular." The recommended movie list is generated according to users' taste and preference. A web scraper fetched data (reviews, Twitter likes, votes, and ratings) from external data source sites (CinemaBlend, Moviefone, Rotten Tomatoes, and Twitter) and stored them in the NoSQL database for computation. Users' feedback about a movie and app is useful for generating the recommended list and evaluation of system reliability.

3.1. NLP Module. NLP has the capability to understand natural language. Users share their opinions and reviews from the microblog that help in making a decision. Positivity, negativity, and neutrality are extracted by opinion mining, whereas emotions are extracted by semantic analysis. In our work, the NLP module determines the semantic emotion of the movie's reviews by the LSTM-attention machine learning algorithm. This semantics is one of the parameters in multivariates used to make a recommendation. This methodology for semantics is depicted as follows:

(i) The module fetches the reviews from microblogs related to movies such as CinemaBlend, Moviefone, and Rotten Tomatoes

(ii) The module preprocesses the microblog text or reviews using a sentence splitter, tokenizer, and stemmer/lemmatizer

(iii) The module determines the sense of the word to strength the sentiment using SenticNet

(iv) Semantic parsing based on attention is done to construct a parse tree to identify the syntactic tree as the emotion of the sentence

(v) RNN/LSTM-user movie attention (UMA) machine learning algorithm is used to classify the reviews

3.2. Preprocessing. It is estimated that more than 80% of data are unstructured and not in an organized manner.

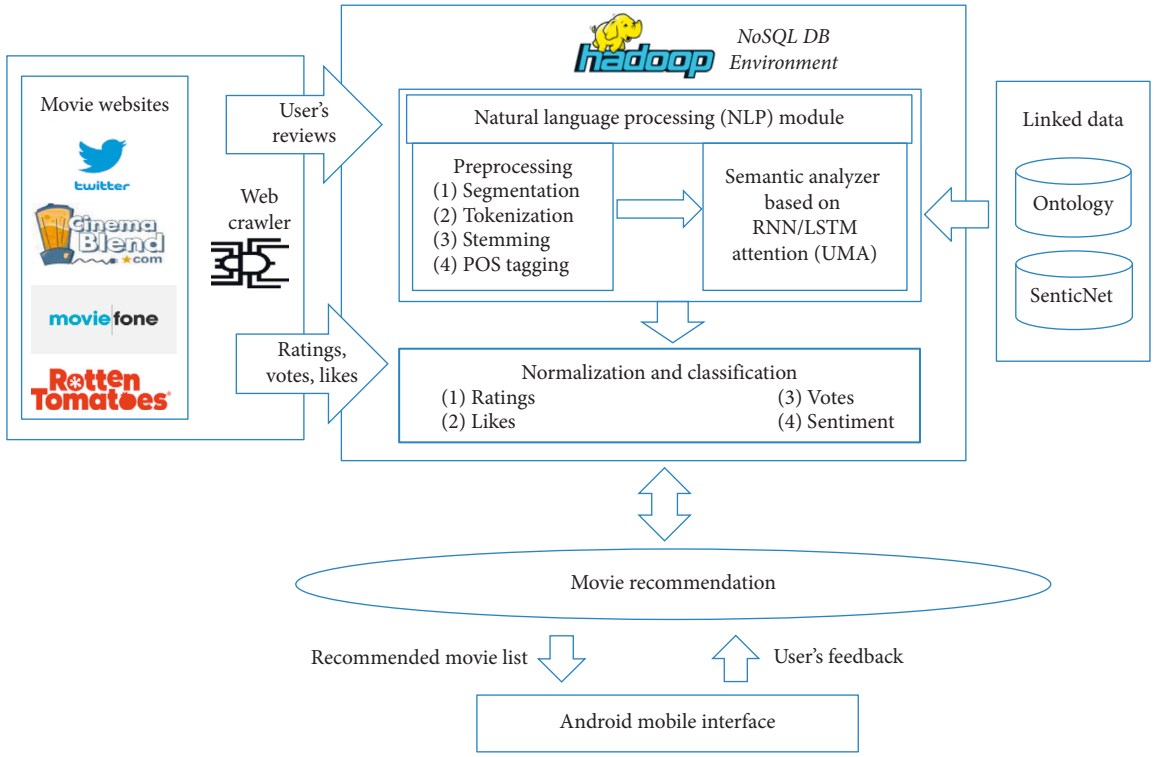

Figure 1: Architecture of the multivariate movie recommendation system.

Preprocessing of text is cleaning or normalization of text/reviews. Stemming or lemmatization and tokenization are done to reduce the sparsity and shrink the feature space. Semantic analysis has to face some challenges such as short text, misspelling, grammatical mistake, slang, unusual terms, tags, white spaces, noise, and emoji. Text is a sequence of words, while word is a meaningful sequence of characters. However, the question is how to find out the boundaries of words. Words are identified by spaces or punctuation in English. However, a compound word is a set of words which have no spaces in German, for example, ("childhood memories description of an unforgettable event") ⟶ ("Kindheitserinnerungen Beschreibung eines unvergesslichen Ereignisses"), while there are no spaces at all in Japanese like this ("childhoodmemoriesdescriptionofanunforgettableevent").

3.2.1. Tokenization. The process of splitting the text stream into units is called tokenization. Units refer to tokens. For example, "This movie is so riddled" is a character string which is tokenized as [This] [movie] [is] [so] [riddled]. Splitting the input sequence into tokens has some problems. Splitting by white space has a problem that different tokens are tokenized into similar words, while the same words may have similar meanings (https://NLTK.Tokenize.WhiteSpaceTokenizer). Splitting by punctuation in which some punctuation are not meaningful is like "An apostrophe problem" (https://NLTK.Tokenize.WordPunctTokenizer). Splitting comes up with the set of rules that generate a more meaning full result (https://NLTK.Tokenize.TreeBankWordTokenizer).

3.2.2. Stemming (Lemmatization). The stemmer stemmed the words like the Porter stemmer, which stemmed the English words "looked" as "look" with a morphological production rule, for example, [("SSES ⟶ SS"): ("Caresses ⟶ caress")], [("IES ⟶ I"): ("Ponies ⟶ Poni")], [("SS ⟶ SS"): ("Caress ⟶ Caress")], and [("S ⟶ S"): ("Cats ⟶ Cat")], but due to stemming of nonwords, the same plural word can be stemmed to singular and irregular forms. These are produced like (Wolves ⟶ wolv), (Feet ⟶ Feet). The WordNet database is looked up for lemmas to solve this type of problem. It solves some specific problems but not all, like (Wolves ⟶ wolf) and (Feet ⟶ Foot) (https://NLTK.Stem.WordNetlemmatizer).

3.2.3. POS-Tag Generation. POS tags are determined for all the tokens by Treebank POStagger. Treebank Project 1 represents 36 POS tags (http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html). For example, the POStag of string "Unwatchable I made it through 20 minutes I think" is [Unwatchable/VB] [I/PRP] [made/VBD] [it/PRP] [through/IN] [20/CD] [minutes/NNS] [I/PRP] [think/VBP].

3.2.4. Word Sense Disambiguation (WSD). WSD is the issue of deciding the "sense" of a word. A lexicon controls a word and its conceivable faculties. Bar-Hillel, 1960, presented the example ["Little John was looking for his toy box. Finally, he found it. The box was in the pen. John was very happy."]. In the previous string, a word "pen" has different senses according to WordNet. "Pen" word defines an "ink flow from a point to write"; here, pen is defined as an "arena of

cattle" and as a "bird's family." In the assessment of the movie's reviews, SenticNet is utilized to indicate their degrees of polarity, antagonism, and impartiality. The SenticNet score of the terms and its recurrence are determined to get the general supposition of the reviews (https://sentic.net) [64].

3.3. Parsing. In NLP, parsing is the process of determining the structure of a sentence by analyzing its essential words based on an underlying syntax. The Stanford parser is used to construct the parse tree that determines the syntactic structure relative to grammar (language). Parsing can refer to various things. Shallow parsing or chunking is the process of grouping the words into noun phrases (NP). Stuff can also be grouped into VP (verb phrases) and PP (prepositional phrases) using grammar like (S $\longrightarrow$ NP | VP), (NP $\longrightarrow$ Det-Noun), (NP $\longrightarrow$ ProperNoun), and (VP $\longrightarrow$ Verb | NP). In contrast, dependency parsing determines the dependencies between the words and their type. For example, spaCy + displaCy for parsing and rendering is used to produce a more semantic result.

3.3.1. RNN/LSTM. Neural networks are represented by RNN/LSTM cells [65]. Typically, in Birdseye, RNN/LSTM is a chain of several copies of the same static network, as shown in Figure 2. From input, the sequence of copies of networks is working in a single timestep. In addition, networks are linked with each other via their hidden states h. So we can say that every copy network has its own inputs as the copy network is unfolded or unrolled. Let the sequence be represented as x_1, x_2, $x_3 \ldots x_n$ and each timestep be represented as $x_t \in x_1 \ldots x_n$. At timestep t, h_t is a hidden layer and f is used to calculate the hidden state: $h_t = f(h_{t-1}, x_t)$. A word is represented by a timestep in the long sequence. For example, the given string is represented as a sequence in the mathematical form: "it is a good movie" $\longrightarrow$ ["it," "is," "a," "good," "movie"]. And the timestep ($t = 0, 1, 2, \ldots$) for the string "it" is represented as x_0, "is" as x_1, "a" as x_2, "good" as x_3, and "movie" as x_4. If $t = 1$, then x_t = "is" $\longrightarrow$ "current timestep to event" and x_{t-1} = "it" $\longrightarrow$ "previous time stamp to event":

$$
\begin{aligned}
X &= \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}, \\
\text{input gate at time } t\text{:}\quad i_t &= \sigma\left(W_i \cdot X + b_i\right), \\
\text{forget gate at time } t\text{:}\quad f_t &= \sigma\left(W_f \cdot X + b_f\right), \\
\text{candidate state at time } t\text{:}\quad \widetilde{C}_t &= \tanh\left(W_c \cdot X + b_c\right), \\
\text{final memory cell:}\quad C_t &= f_t * c_{t-1} + i_t * \widetilde{C}_t, \\
\text{output gate:}\quad o_t &= \sigma\left(W_o \cdot X + b_o\right), \\
h_t &= o_t * \tanh\left(C_t\right).
\end{aligned}
\tag{1}
$$

At the $i_t s$ input gate, the decision on which information should be remembered or rid of is made by the sigmoid function σ. It produces a 0 or 1 value: 0 means forget, while 1 means remember in the cell state. Sigmoid function at the input gate takes a decision on which value should be updated, and the new candidate value information is represented by tanh function $\widetilde{C}_t$. Output gate sigmoid function decides which part of information should be produced, and then tanh function produces the value between 1 and −1.

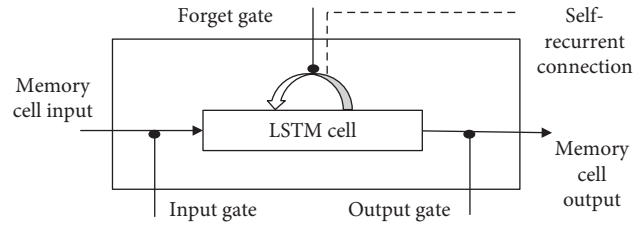

Figure 2: LSTM cell architecture.

The sequential semantic information is preserved in the recurrent neural network's hidden states. In the hidden state (h_t), the semantic information of the input sequence is preserved. When a new input is experienced and again delivered to be the subsequent input, then semantic information is altered. Passing the information from one to another network helping to find out the correlation among the words from the sequence is represented as a long-term dependency.

3.3.2. LSTM-Based Sequence Labeling. Predicates from a given input sequence are marked, and the label arguments corresponding to every predicate are identified. For example, in the given sentence "I watched the movie," the predicate (watched) is marked, and labels corresponding to the predicate are "I," "the," and "movie" as an agent, null, and theme, respectively. Multiple predicates may present in a sentence, and different labels may be marked to the same word for every predicate. Concatenating pretrained ones (Word2vec) generates vectors of every word. The 1-bit flag represents the predicate in the specific training unit to confirm that the network deals with every predicate separately and serves it into the LSTM layer to the word context. With the predicate, any one word is labeled to take the dot product of its hidden state. A softmax function is applied over it. The probability of a sentence is calculated as follows:

$$
\begin{aligned}
P(X) &= \prod_{k=1}^{n} P\left(x_i \mid x_1, x_2, x_3 \ldots x_{t-1}\right), \\
X_{l,r} &= \mathrm{ReLU}\left(x_l \cdot x_r\right).
\end{aligned}
\tag{2}
$$

Here, the role label r is calculated by the weight matrix parameter using ReLU function and predicate lemma and the role depicted by taking the dot product of vectors to embedding.

3.3.3. Neural Sentiment Classification (NSC). Document-level sentiment classification is measured by neural sentiment classification (NSC) based on hierarchical LSTM attention with user movie attention (UMA) (see Figure 3) that is represented by the user's global information and movie features [28]. Let a review $d \in D$ with sentences, each sentence $(s_1, s_2, \ldots, s_n)$ of a particular review $s_i \in d$, a user $u \in U$, and a movie $m \in M$ review corpus (users and their movie set). Moreover, l_i is the length of the i-th sentence,

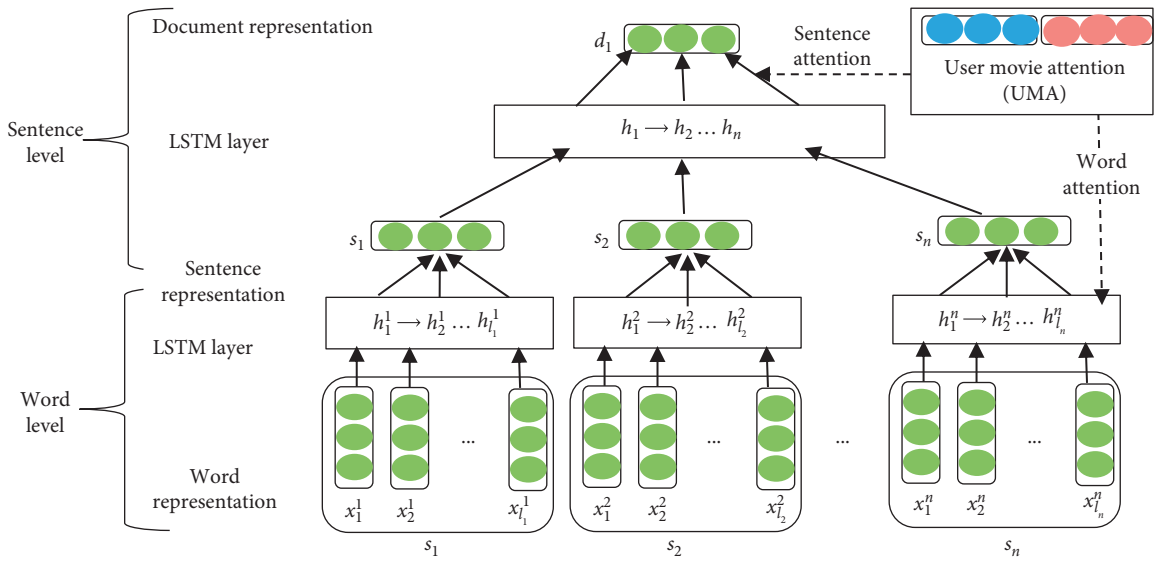

Figure 3: LSTM-attention (UMA) architecture.

while s_i consists of l_i words as $x_1^i, x_2^i, \ldots, x_{l_i}^i$. Predicting the semantic rating of documents is done according to their text information. Firstly, in word-level low-dimensional semantic space, each word x_j^i is mapped to its embedding $x_j^i \in \mathbb{R}^d$ in a sentence. Every step has a given input word x_j^i, the current cell state c_j^i, and the hidden state h_j^i that may be updated with the preceding cell state c_{j-1}^i. Then, the hidden state h_{j-1}^i is represented. The document representation architecture is presented as follows:

$$\begin{aligned}\begin{bmatrix} i_j^i \\ f_j^i \\ o_j^i \end{bmatrix} &= \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix}\left(W \cdot \left[h_{j-1}^i, x_j^i\right] + b\right),\\ \widehat{c}_j^i &= \tanh\left(W \cdot \left[h_{j-1}^i, x_j^i\right] + b\right),\\ c_j^i &= f_j^i \odot c_j^i + i_j^i \odot \widehat{c}_j^i,\\ h_j^i &= o_j^i \odot \tanh\left(c_j^i\right).\end{aligned} \tag{3}$$

Sigmoid activation function and gate activation functions are represented as σ and i, f, and o, respectively, while elementwise multiplication is represented as $\odot$. Training parameters needed for training are represented as x and b. The feed hidden states $[h_1^i, h_2^i, \ldots, h_{l_i}^i]$ are represented to a mediocre pooling layer to acquire the representation of the s_i sentence. Sentences are embedded at the sentence level $(s_1, s_2, \ldots, s_n)$ into the LSTM; after that, document representation d is acquired via a mediocre pooling layer in a similar way as follows:

$$\begin{aligned}\begin{bmatrix} i_i \\ f_i \\ o_i \end{bmatrix} &= \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix}\left(W \cdot \left[h_{i-1}, s_i\right] + b\right),\\ \widetilde{C}_i &= \tanh\left(W \cdot \left[h_{i-1}, s_i\right] + b\right),\\ C_i &= f_i \odot c_{i-1} + i_i \odot \widetilde{C}_i,\\ h_i &= o_i \odot \tanh\left(C_i\right).\end{aligned} \tag{4}$$

Here, training parameters needed for training are represented as s and b. The feed hidden states $[h_1, h_2, \ldots, h_n]$ are represented to a mediocre pooling layer to acquire the d_i document representation.

3.3.4. User Movie Attention (UMA). At various levels, a necessary component is extracted by using user movie attention (UMA) for sentiment classification. UMA is applied at the word level to construct a sentence and sentence level to generate a document. Obviously, sentence meaning may not be represented by all words for several users and movies. In spite of feeding hidden states at the word level to an average pooling layer, user movie attention (UMA) is used to extract user/movie relative words, which are essential to sentence meaning. Informative words are aggregated to produce the representation of the sentence. Formally, weighted hidden states generate the enhanced sentence as follows:

$$\begin{aligned} s_i &= \sum_{j=1}^{l_i} a_j^i h_j^i,\\ d_i &= \sum_{i=1}^{n} a_i h_i.\end{aligned} \tag{5}$$

Importance of the jth word is measured by a_j^i for the current user and movie. Each user u and movie m are embedded continuous and real-valued vectors $u \varepsilon \mathbb{R}^{d_u}$ and $m \varepsilon \mathbb{R}^{d_m}$, while user and movie embedding is represented as d_u and d_m dimensions, respectively. Moreover, for every hidden state, the attention weight a_j^i is presented as follows:

$$a_j^i = \frac{\exp\left(e\left(h_j^i, u, m\right)\right)}{\sum_{k=1}^{l_i} \exp\left(e\left(h_k^i, u, m\right)\right)}. \tag{6}$$

For the sentence level,

$$a_i = \frac{\exp\left(e\left(h_i, u, m\right)\right)}{\sum_{i=1}^{n} \exp\left(e\left(h_i, u, m\right)\right)}. \tag{7}$$

Importance of words for sentence representation as well as document representation is presented by e score function as follows:

$$e\left(h_j^i, u, m\right) = v^{\mathrm{T}} \tanh\left(W_h h_{ij} + W_u u + W_m m + b\right). \tag{8}$$

For the sentence level,

$$e\left(h_i, u, m\right) = v^{\mathrm{T}} \tanh\left(W_h h_i + W_u u + W_m m + b\right), \tag{9}$$

where v is a weight vector and v^{T} represents its transpose, while W_h, W_u, and W_m are weight matrices. Meaning of every document varies for different users and movies by the sentence, which provides the hints. So in the sentence level, usage of attention a with the u user and m movie vector at the word level to select informative sentences to generate document representation d is presented as follows:

$$d = \sum_{i=1}^{n} \beta_i h_i. \tag{10}$$

In the sentence level, the β_i weight of the h_i hidden state is measured similar to word attention. The higher level representation of document d is generated by hierarchical extraction from words and sentences in the document. So, for sentiment classification of the document, it is used as features. tanh activation function is used at the nonlinear layer for current document representation in the target space of C classes:

$$\widehat{d} = \tanh\left(W_c d + b_c\right). \tag{11}$$

tanh activation function is used at an absolute layer to get sentiment distribution of the document:

$$d_c = \frac{\exp\left(\widehat{d}_c\right)}{\sum_{k=1}^{C} \exp\left(\widehat{d}_k\right)}. \tag{12}$$

Sentiment classes and prediction probability of sentiment class C are represented as C and p_c, respectively. During the training, loss function for optimization is measured by error cross-entropy between the distribution of Gold sentiment and distribution of our model sentiment as follows:

$$L = \sum_{d \varepsilon D} \sum_{c=1}^{C} p_c^g\left(d\right) \cdot \log\left(p_c\left(d\right)\right). \tag{13}$$

Here, Gold probability of sentiment class C and training document are represented as p_c^g and d, respectively, while reality-based truth is one and others are zero.

Some nomenclatures used in our mathematical model are presented in Table 1.

Table 2 presents the emotion class.

3.4. Computation and Classification. Sentiment analysis determines the emotions of reviews. Firstly, the aggregated sentiment score of each document from each site for the j-th movie is computed. Then, the qualitative score and then aggregated quality scores for Twitter likes are computed to get the final score for the recommendation of movies and generate the popularity class relative to the final score:

$$\begin{aligned}
C^{j,1} &= \sum_{\substack{i=0 \\ k=1}}^{n} dc_i^{j,k}, \\
C^{j,2} &= \sum_{\substack{i=0 \\ k=2}}^{n} dc_i^{j,k}, \\
C^{j,3} &= \sum_{\substack{i=0 \\ k=3}}^{n} dc_i^{j,k}, \\
q^{j,1} &= \left[\left(R^{j,1}\right) + \left(V^{j,1}\right) + \left(C^{j,1}\right)\right], \\
q^{j,2} &= \left[\left(R^{j,2}\right) + \left(V^{j,2}\right) + \left(C^{j,2}\right)\right], \\
q^{j,3} &= \left[\left(R^{j,3}\right) + \left(V^{j,3}\right) + \left(C^{j,3}\right)\right], \\
Q^j &= \left[\left(q^{j,1}\right) + \left(q^{j,2}\right) + \left(q^{j,3}\right)\right], \\
Q^j &= \sum_{k=1}^{3} q^{j,k},
\end{aligned} \tag{14}$$

j-th movie final multivariate emotional score E

$$= \log\left(\gamma\left[\left(Q^j\right) + (\mathrm{TL})\right]\right),$$

where $\gamma = 0.5$, and

j-th movie popularity score P

$$= \left[\frac{f_{m_j} - \min\left(f_{m_j}\right)}{\max\left(f_{m_j}\right) - \min\left(f_{m_j}\right)} * 10\right]. \tag{15}$$

This mathematical formulation is used to determine the final popularity score using the multivariate model. The emotional value is stretched to 10 scales, by which the popularity status is determined. Every movie is labeled with a medal according to the popularity score, and the algorithm that identifies the medal by using fuzzy logic on behalf of the popularity score to find the popularity of the movies is depicted as follows:

(1) IF multivariate value ≥08, THEN: Platinum: "Highly Popular"

(2) ELSE IF multivariate value ≥06, THEN: Gold: "Popular"

(3) ELSE IF multivariate value ≥04, THEN: Silver: "Average Popular"

(4) ELSE IF multivariate value ≥02, THEN: Bronze: "Unpopular"

(5) ELSE Copper: "Highly Unpopular"

Table 1: Nomenclatures and description.

Nomenclature	Description
d	Document/review
s	Sentence
x	Word
D	Review corpus
m	Movie
l	Length of a sentence
h	Hidden state
S	Total movie sites
b	Biases
TL	Twitter likes
t	Timestep
$C^{j,1}$	j-th movie sentiment at site S_1
$C^{j,2}$	j-th movie sentiment at site S_2
$C^{j,3}$	j-th movie sentiment at site S_3
$R^{j,1}$	j-th movie rating at site S_1
$R^{j,2}$	j-th movie rating at site S_2
$R^{j,3}$	j-th movie rating at site S_3
Q^j	j-th movie total quantitative score
RecS	Final recommendation score
AWAS	Aggregated weighted average sentiment
Multivariate	Multivariate final score
i	Input gate
o	Output gate
f	Forget gate
σ	Activation function
b	Biases
v	Weight vector
v^{T}	Vector transpose
t	Timestep
$\odot$	Multiplication
h_t	Hidden state at t timestep
h_{t-1}	Hidden state at t–1 (previous) timestep
W	Weight matrix for input to hidden layers at t timestep
$\varnothing$	tanh is an activation function
x_j^i	Input at timestep (t)
$V^{j,1}$	j-th movie votes at site S_1
$V^{j,2}$	j-th movie votes at site S_2
$V^{j,3}$	j-th movie votes at site S_3
L	Loss
AS	Aggregated sentiment
WAS	Weighted average sentiment

Table 2: Emotion status.

Semantic score	Emotional class
0.5< and ≤1.00	Highly Favorable
0.00< and ≤0.5	Favorable
−0.5< and ≤0.00	Average Favorable
−1.00< and ≤−0.50	Unfavorable
≤−1.00	Highly Unfavourable

The ranges of the popularity scores and their respective medals and degree of popularity are given in Table 3.

The category represented by a movie genre to classify the movie according to its features, movie recommendation services suggests top 10 popular movies with their category according to the user request and profile history.

Table 3: Popularity scores and their respective medals and popularity status.

Popularity score	Medal rank	Status
0.8–1.0	Platinum	Highly Popular
0.6–0.79	Gold	Popular
0.4–0.59	Silver	Average Popular
0.2–0.39	Bronze	Unpopular
0.0–0.19	Copper	Highly Unpopular

4. Multivariate Movie Recommendation System Implementation

4.1. System Component Interaction. User android application is front end of the system (see Figure 4) by which users can get the web services from the system, and back end is the movie recommendation system in the NoSQL environment with Apache Mahout and Hadoop, which provide the web services to the users as well as a web scraper by which the system fetched the data. Web scraper fetched the data from the external data source on the bases of matching lexicons of the query and movie content.

4.2. NoSQL Environment Implementation

4.2.1. Hadoop Architecture. It is a framework with four fundamental components: (1) HDFS splits the file into many small files and stores them on three servers for fault tolerance constraints as replicas in a distributed file system manner. (2) Map Reduce programming standard is for handling and manipulating big data. (3) Common/Core holds the reference library and services to backing up Hadoop. (4) YARN performs management, computation, and scheduling of resources and tasks.

4.2.2. Apache Mahout. Implementation of collaborative filtering, clustering, and classification is done by Apache Mahout. In the NoSQL environment, Apache Mahout interfaces implement the Hadoop framework and evaluate the performance similarities and neighborhood measures. A multivariate web scraper is implemented and big data are generated.

4.3. Web Scraper. Our web scraper is a scripting program, which surfs the W3, fetches data from different movie websites to extracts the reviews, votes, ratings, and Twitter likes, and stores them in the repository. In addition, it manages and handles scrape data in a NoSQL environment using Hadoop and Apache Mahout. The web scraper (web bot) receives the URLs and matches them with keywords (Meta tags) of the web page. If the keywords are matched, then the web pages are downloaded; otherwise, the irrelevant pages are discarded.

4.4. NLP Tools. Stanford CoreNLP technology tools are used to process the natural language like English. They give the words, relative parts of speech, and identification of

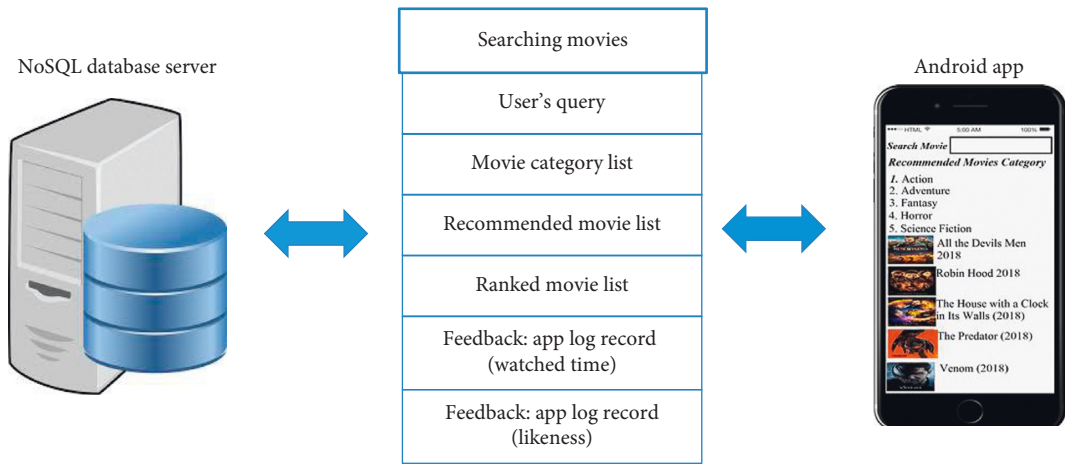

Figure 4: Interaction of components of multivariate movie recommendation.

sentiments. The Stanford CoreNLP framework is the integration of many of Stanford's NLP tools, like POStagger, parser, sentiment analyzer, named "entity recognizer," and pattern learning and information extracting tools from https://stanfordnlp.github.io/CoreNLP.

4.5. Mobile Application Usage

4.5.1. Unregistered Users. Unregistered users can request the movie by the search query to our recommendation system, and the system will respond to that query by content filtering to extract the features or content of a movie from the query. Collaboration is done between the user request and system-generated movies. The system provides the watched window to unregistered users for watching the recommended movies. Unregistered users give feedback by their likeness, and the system uses the feedback for accurate measurement.

4.5.2. Registered Users. If unregistered users sign up, then they can sign in and maintain their profile or history. For registered users, collaborations may be done on the bases of both the query and the history. Registered users may provide feedback to the system by their likeness, and the system uses this feedback for collaborative filtering between liked movies or their movie history and system movies for recommendations of the movie of their choices as well as accurate measurement of the multivariate movie recommendation system. The application also provides a watch window for registered users to watch the recommended movie.

4.6. Cold Start Problem Handling. Collaborative filtering (CF) is done for movie recommendation for registered and unregistered users; but in two cases, the problem may occur:

(i) Case 1: if the registered users request the movie, the system collaborates the requested movie with the system movie and recommends the movies on behalf of user history. Here, one problem arises: if the newly registered users request the movies, then the system recommends the movies according to movies mostly liked by others to solve the cold start problem of newly registered users.

(ii) Case 2: if a new movie arrives for registered or unregistered request, then the system recommends the movies according to the collaboration of new movie trailers, which were mostly liked to solve the cold start problem of newly released movies.

4.7. Similarity Measurement. We use cosine similarity in which there are two vectors for measuring the angle value for similarity manipulation. A smaller angle degree is directly proportional to larger similarity, and vice versa, as shown in Figure 5. It is also known as vector-based similarity. Movie document and search query document correlation is computed where q is the search query document and d is the movie document. The similarity can be calculated by the following equation:

$$\vec{q} \cdot \vec{d} = \|\vec{q}\| \cdot \|\vec{d}\| \cdot \cos\theta,$$
$$\text{sim}(q \cdot d) = \cos\theta = \frac{\vec{q} \cdot \vec{d}}{\|\vec{q}\| \cdot \|\vec{d}\|}. \quad (16)$$

5. Experiments and Results

The procedure followed by data preprocessing is NLP procedures applied for sentiment analysis on fetched data, and then the sentiment score is computed by using SenticNet and obtained results are presented as follows.

Table 4 presents the identification of movies and categories.

Table 5 presents the identification of sites, movies, users, and reviews.

Table 6 presents the movie review from the movie website CinemaBlend.

Movie reviews from movie websites Moviefone and Rotten Tomatoes were also fetched, and semantic scores and emotions were computed.

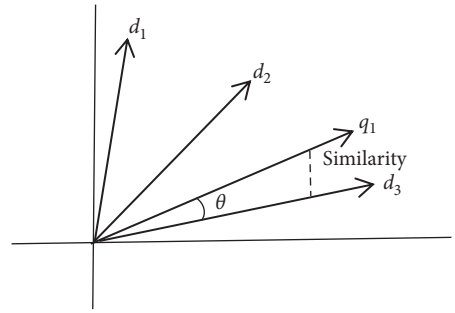

Figure 5: Cosine similarity angle. *Note.* If the angle between vectors is zero degrees, then the cosine similarity value is 1, which means movies are similar or relevant, and if the angle is 90 degrees or above, then the similarity value is zero, which means the movie is irrelevant.

Table 4: Movie ID and movie category ID.

Movie and category IDs		
Movie ID	Movie title	Movie category
m_1	Robin Hood (2018)	Action (c_1)
m_2	The House with a Clock in Its Walls (2018)	Adventure (c_2)
m_3	The Predator (2018)	Fantasy (c_3)
m_4	Venom (2018)	Horror (c_4)
m_5	The Flash	Science fiction (c_5)

Table 5: IDs of sites, movies, user names, and reviews.

ID table					
Site name	Site ID	Movie ID	User name	User ID	Review ID
CinemaBlend	s_1	m_1	Deplorable_me	u_1	d_1
		m_2	Snow gator	u_2	d_2
		m_3	David Curry	u_3	d_3
		m_4	Smedley	u_4	d_4
		m_5	DC villains	u_5	d_5
Moviefone	s_2	m_1	The Guardian Peter Bradshaw	u_6	d_6
		m_2	Snow gator	u_7	d_7
		m_3	Relax ad mike	u_8	d_8
		m_4	Jza Smack	u_9	d_9
		m_5	Clifford De Voe	u_{10}	d_{10}
Rotten Tomatoes	s_3	m_1	Jennifer Heaton	u_{11}	d_{11}
		m_2	Carlos Díaz Reyes	u_{12}	d_{12}
		m_3	Jeffrey Bloomer	u_{13}	d_{13}
		m_4	ugene Bernabe	u_{14}	d_{14}
		m_5	Dee R.	u_{15}	d_{15}

Table 7 presents the movie review tokenization and tagging from movie websites CinemaBlend, Moviefone, and Rotten Tomatoes.

The parsing of movie reviews' tokens taken from CinemaBlend, Moviefone, and Rotten Tomatoes was performed using the Stanford parser. Here, Table 8 presents the sentiment score of movie reviews from CinemaBlend, Moviefone, and Rotten Tomatoes.

Table 9 presents the normalized Twitter likes of movies from Twitter.

Table 10 presents the normalized rating score of movies from CinemaBlend, Moviefone, and Rotten Tomatoes.

Table 11 presents the normalized vote score of movies from CinemaBlend, Moviefone, and Rotten Tomatoes.

Table 12 presents the final score, movie category, medal rank, and genres of movies from CinemaBlend, Moviefone, and Rotten Tomatoes.

Figure 6 presents the multivariate movie ranked recommendation of movies from CinemaBlend, Moviefone, and Rotten Tomatoes.

Figure 7 presents differences in the rating of the movie from CinemaBlend, Moviefone, and Rotten Tomatoes.

Figure 8 presents differences in the votes of the movie from CinemaBlend, Moviefone, and Rotten Tomatoes.

Figure 9 presents differences in the sentiment of the movie from CinemaBlend, Moviefone, and Rotten Tomatoes.

6. Evaluation and Discussion

We evaluate the sentiment classification models as well as recommendation models as follows.

6.1. Sentiment Classification Model Evaluation. For sentiment, classification models are evaluated by accuracy and RMSE, which measure the overall performance of the sentiment classification model and divergence between predicted and truth ground sentiment classes, respectively. We compare the several base sentiment classification methods using three datasets imdb, yulp13, and yulp14, which contain reviews about movies using Stanford CoreNLP. Majority of the baseline sentiment classification models refer to categorization of document sentiments in the training set by an *SVM* classifier with unigram, bigram, and trigram. Text feature extraction including character *n*-gram and -word is done by the SVM classifier. Use of leniency feature is extracted by UPF [66]. Document representation is obtained by AvgWordvec, which nourished into SVM. Feature generation is by SSWE (sentiment-specific word embedding) [67]. Sentence representation is by the RNTN (recursive neural tensor network) [68]. Document classification is by

Table 6: Semantic emotion of movie reviews of users from CinemaBlend.

Reviews of users about movies				
Movie ID	User name	Review ID	Reviews	Semantic emotion
m_1	u_1	d_1	Unwatchable I made it through 20 minutes I think.	Average Favorable
m_2	u_2	d_2	I did not like it. Thought it was uneven and wasted some great talent. Not funny enough, too much turd humor, and think it is a made for USA level of quality with better effects. It is worth seeing for Blanchett. She does steal every scene, and when the sequel happens—and it has made more than enough money for one—I hope she is front and center as the main character. She and Black do have fantastic chemistry.	Average Favorable
m_3	u_3	d_3	Predator 1 and 2 had comedy in it. Shane Black helped write the original (everyone should know that by now). Predators are the most serious movie of the franchise.	Average Favorable
m_4	u_4	d_4	I went to see it today with open expectations (professional reviews bad, viewer reviews good) and thought it was a fun movie. It cracked me up a couple of times.	Highly Favorable
m_5	u_5	d_5	The Flash has done a fantastic job of incorporating classic from the hero's comic book history	Highly Favorable

Table 7: Movie review tokenization and tagging.

		Tags
User ID	Tokens per document	Tagging
u_1	9	Unwatchable/VB I/PRP made/VBD it/PRP through/IN 20/CD minutes/NNS I/PRP think/VBP
u_2	91	Thought/RB it/PRP was/VBD uneven/JJ and/CC wasted/VBD some/DT great/JJ talent/NN ./. Not/RB funny/JJ enough/RB ,/, too/RB much/JJ turd/VBD humor/NN ,/, and/CC think/VBP it/PRP is/VBZ a/DT made/VBN for/IN USA/NNP level/NN of/IN quality/NN with/IN better/JJR effects/NNS ./. It/PRP is/VBZ worth/JJ seeing/VBG for/IN Blanchett/NNP ./.She/PRP does/VBZ steal/VB every/DT scene/NN ,/, and/CC when/WRB the/DT sequel/NN happens/VBZ -/: and/CC it/PRP has/VBZ made/VBN more/RBR than/IN enough/JJ money/NN for/IN one/CD -/: I/PRP hope/VBP she/PRP is/VBZ front/NN and/CC center/NN as/IN the/DT main/JJ character/NN ./. She/PRP and/CC Black/NNP do/VBP have/VB a/DT fantastic/JJ chemistry/NN ./.
u_3	34	Predator/NNP 1/CD &/CC 2/CD had/VBD comedy/NN in/IN it/PRP ./. Shane/NNP Black/NNP helped/VBD write/VB the/DT original/NN -LRB-/-LRB- everyone/NN should/MD know/VB that/DT by/IN now/RB -RRB-/-RRB- ./. Predators/NNS is/VBZ the/DT most/RBS serious/JJ movie/NN of/IN the/DT franchise/NN ./.
u_4	34	I/PRP went/VBD to/TO see/VB it/PRP today/NN with/IN open/JJ expectations/NNS -LRB-/-LRB- professional/JJ reviews/NNS bad/JJ ,/, viewer/CD reviews/NNS good/JJ -RRB-/-RRB- and/CC thought/VBD it/PRP was/VBD a/DT fun/NN movie/NN ./. It/PRP cracked/VBD me/PRP up/IN a/DT couple/NN times/NNS ./.
u_5	17	The/DT Flash/NNP has/VBZ done/VBN a/DT fantastic/JJ job/NN of/IN incorporating/VBG classic/NN from/IN the/DT hero/NN's/POS comic/JJ book/NN history/NN
—	—	—
u_n	—	—

Table 8: Movie reviews' semantic score.

		Aggregated semantic score		
Movie ID	Review ID	CinemaBlend	Moviefone	Rotten Tomatoes
m_1	d_1	0.4	0.6	0.2
m_2	d_2	0.2	0.2	0.4
m_3	d_3	0.2	0.6	0.6
m_4	d_4	0.6	0.2	0.4
m_5	d_5	0.8	0.0	0.0

Table 9: Twitter likes.

	Twitter likes	
Movie name	Unnormalized	Normalized
m_1	366	366
m_2	154	154
m_3	258	258
m_4	3	3
m_5	2196K	2169

paragraph vector: distributed memory model (PVDM) [69], topic modeling, and collaborative filtering JMARS (https://jmars.asu.edu/). Sentiment classification is by vector representation and text preference matrix for the user product neural network (UPNN) [70].

Table 13 presents the comparison between different sentiment classification models using and without using users/product/movies information.

In our approach, the core implementation is neural sentiment classification (NSC) using local user and movie information [71], which provides the significant result, as

Table 10: Movie rating.

Movie ID	Rating					
	Unnormalized			Normalized		
	CinemaBlend (%)	Moviefone (%)	Rotten Tomatoes (%)	CinemaBlend	Moviefone	Rotten Tomatoes
m_1	70	16	39	7	1.60	3.90
m_2	80	55	60	4.0	5.50	6.00
m_3	70	25	49	3.5	2.50	4.90
m_4	40	Nil	39	2.0	Nil	3.90
m_5	73	69	93	7.3	6.90	9.30

Table 11: Normalized movie votes.

Movie ID	Votes					
	Unnormalized			Normalized		
	CinemaBlend	Moviefone	Rotten Tomatoes	CinemaBlend	Moviefone	Rotten Tomatoes
m_1	1.5K	679	4.5K	1500	679	4500
m_2	870	760	6.5K	870	760	6500
m_3	797	890	94	797	890	94
m_4	3.46K	6.9K	910	3460	6900	910
m_5	67	76	8.3K	670	760	8300

Table 12: Final score, movie category, medal rank, and genres.

Movie ID	Final score	Genre category	Medal rank	Recommendation of movie
m_1	1.30	Action	Copper	Highly Unpopular
m_2	4.41	Adventure	Silver	Average Popular
m_3	3.63	Fantasy	Bronze	Unpopular
m_4	4.59	Horror	Silver	Average Popular
m_5	4.47	Science fiction	Silver	Average Popular

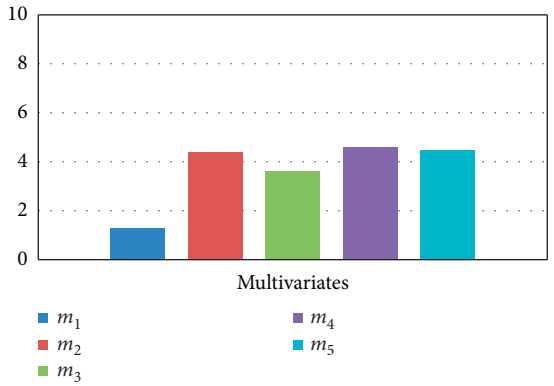

Figure 6: Multivariate movie ranked recommendation.

shown in Table 13, which represents the significant 4% improvement/difference with all the baseline methods, which use the local textual information about users and movies. While using global information about users and movies, the UPNN gains 3% improvement, and our approach NSC-UMA achieves 9% improvement. Our approach uses the vector for embedding the user and movie information, which is suitable for larger datasets, while the UPNN uses the matrix and vector simultaneously. NSC-UMA is considerable for capturing the information from each semantic layer. Therefore, our model incorporates using user movie global information in an efficient and effective way.

Word-level attention and sentence-level attention are considerable to outperform to reflect the semantic information of user and movie characteristics at multiple levels, which leads to introduction of the user movie attention (UMA) in sentiment classification. Furthermore, perceptions of user taste or preferences are more understandable than movie attributes, so both user and movie

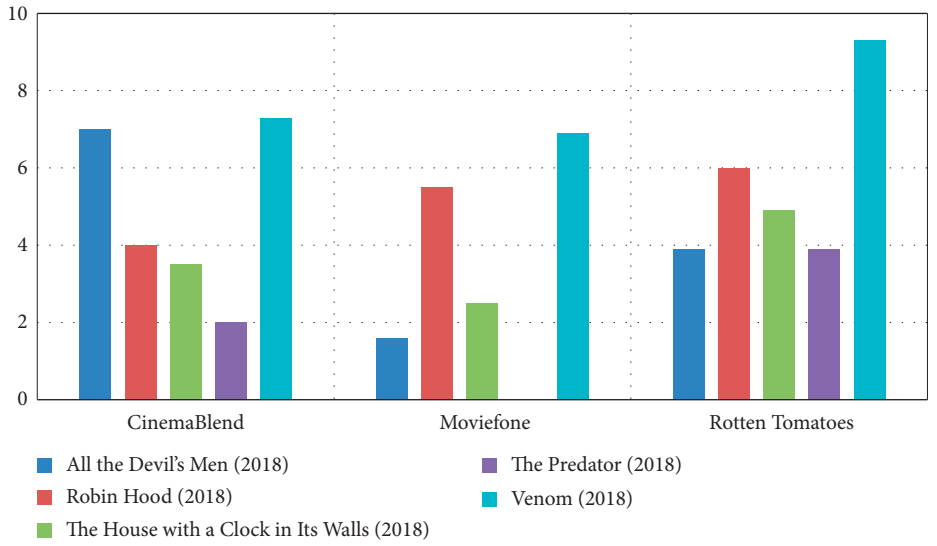

Figure 7: Rating difference.

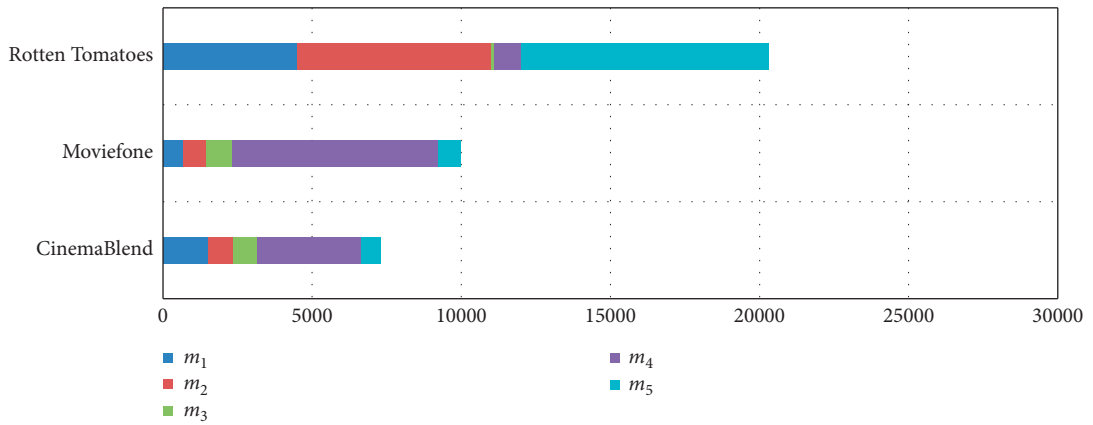

Figure 8: Vote difference.

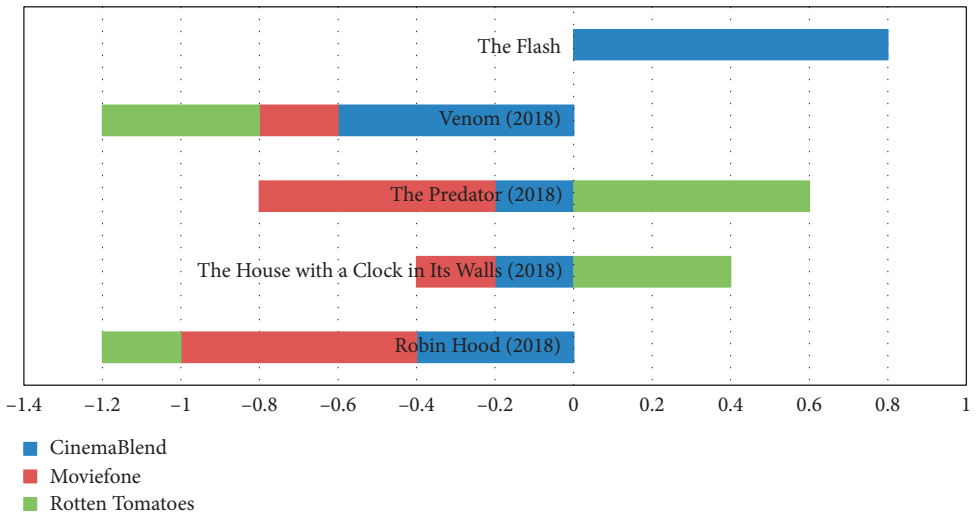

Figure 9: Differences in sentiment scores.

Table 13: Comparison between sentiment classification models.

Classification models	IMDB		Yelp 2013		Yelp 2014	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Without using user and product information						
Majority	0.196	2.495	0.411	1.060	0.392	1.097
Trigram	0.399	1.783	0.569	0.814	0.577	0.804
Text feature	0.402	1.793	0.556	0.845	0.572	0.800
AvgWordvec + SVM	0.304	1.985	0.526	0.898	0.530	0.893
SSWE + SVM	0.312	1.973	0.549	0.849	0.557	0.851
Paragraph vector	0.341	1.814	0.554	0.832	0.564	0.802
RNTN + recurrent	0.400	1.764	0.574	0.804	0.582	0.821
CNN and without UP (UPNN)	0.405	1.629	0.577	0.812	0.585	0.808
NSC	0.443	1.465	0.627	0.701	0.637	0.686
NSC + LA	**0.487**	**1.381**	**0.631**	**0.706**	**0.630**	**0.715**
Using user and product information						
Trigram + UPF	0.404	1.764	0.570	0.803	0.576	0.789
Text feature + UPF	0.402	1.774	0.561	1.822	0.579	0.791
JMARS	N/A	1.773	N/A	0.985	N/A	0.999
UPNN (CNN)	0.435	1.602	0.596	0.784	0.608	0.764
UPNN (NSC)	0.471	1.443	0.631	0.702	N/A	N/A
NSC + UMA	**0.533**	**1.281**	**0.650**	**0.692**	**0.667**	**0.654**

information is essential to pay attention in the document for semantic information which impacts movie ranking for recommendation.

6.2. Comparative Analysis of Recommendation Models. Table 14 presents the comparison between different models. Major differences which show our work as a novel approach are that the first one is LSTM-UMA for sentiment classification, the second one is the NoSQL distributed environment to deal with the big data issues, the third one is the multivariate (qualitative and quantitative) score fetched by a web bot from three different reliable external data sources, and the fourth one is app features (movie category and popularity).

In [15] which only uses the implicate and explicate ratings, no user and production attention are used by LSTM, adoptive deep learning is not used to determine the preference and taste of users about a movie from microblogs, so we can say that the authors did not use the qualitative data. It does not declare how data are fetched nor categorized with their popularity. In [15, 72–74], multivariates are not used, and the study [73] is just based on microblogs, while the study [74] uses movie feature ratings.

6.3. Results of the Experiments. True positive (recommended interesting movie) predicted by a search divided by actual movies (total movies) is called precision:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{17}$$

True positive (recommended interesting movie to users) predicted by a search divided by predicted movies (totally recommended movies) is called recall:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{18}$$

Figure 10 presents different decisions made by the movie recommender.

If the recommender grows in precision, then recall is declined:

$$F \text{ score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{19}$$

Table 15 presents the comparisons of the parameters of the multivariate recommendation system and other models.

Table 16 presents the evaluation of the multivariate recommender system with other parameters and other works.

The results of F score in our system are compared with other predicting parameters as well as with other recommendation frameworks. The accuracy of the multivariate system is nearly about 98.70%. The true positive rate of the multivariate system is 0.99106, which means the system recommended movies truly interested for users, and the false positive rate is 0.01814, which means the multivariate system did not recommend movies truly not interested for users.

Figure 11 presents differences in different decision parameters (precision, recall, and accuracy) for recommendation of movies from movie sites CinemaBlend, Moviefone, and Rotten tomatoes and other recommendation models.

Figure 11 justifies the difference between our approach and other works [21, 74]. In [21], just finding the polarity of the term is not enough to evaluate the reviews for significant recommendation, and in [74], LSTM is used to determine the user and movie information, while we used NSC-UMA to evaluate the sentiment score of reviews for significant recommendation.

Table 14: Recommendation model comparisons.

Ref.	NLP/ RNN/ LSTM	User preferences	NoSQL	Heterogeneous data	Quantitative score (votes, likes, and ratings)	Qualitative score (analysis of reviews)	Multivariates	Multiple data source sites	Popularity medals	Web bot	Categories	User app
[15]	×	✓	✓	×	✓	×	×	×	×	×	×	✓
[72]	✓	✓	✓	×	×	✓	×	×	×	×	×	✓
[73]	✓	✓	×	×	×	✓	×	×	×	×	×	×
[74]	✓	✓	×	×	×	×	×	×	×	×	×	×
[21]	✓	×	×	✓	✓	×	✓	✓	×	✓	✓	✓
Proposed work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

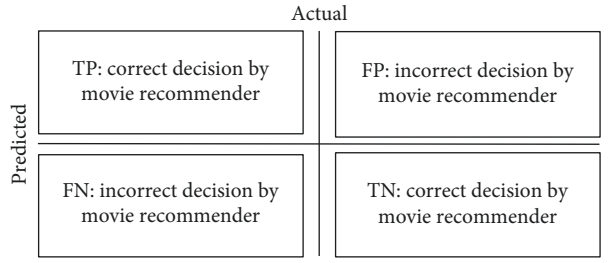

Figure 10: Different decisions made by the movie recommender. TP: the recommended movie is interesting. TN: the recommended movie is uninteresting. FP: the recommended movie is actually interesting. FN: the recommended movie is actually uninteresting.

Table 15: Comparisons between different recommendation decision parameters.

Decision parameters	TP	TN	FP	FN
AS	341	237	207	215
WAS	375	355	95	175
AWAS	406	347	116	131
[74]	525	250	90	135
[21]	442	387	114	57
Multivariate system	554	433	8	5

Table 16: Results of the experiments.

	AS	WAS	AWAS	[74]	[21]	Multivariate system
Precision	0.6223	0.7979	0.7778	0.8537	0.7950	0.9858
Recall	0.6133	0.6818	0.7561	0.7955	0.8858	0.9911
F score	0.6178	0.7353	0.7668	0.8235	0.8379	0.9884
Accuracy	0.5780	0.7300	0.7530	0.7750	0.8290	0.9870

Figure 11: Differences in different decision parameters precision, recall, *F* score, and accuracy and recommendation models.

7. Conclusion

In many movie recommendation systems, suggestion and ranking are done on the bases of only likes or ratings. Furthermore, most of the systems extract data from only one or two sites. Semantics, ratings, or votes for the movie ranking are not reliable and insignificant as they could not provide better recommendation services and there is a huge gap between statistical information (ratings, votes, and likes) and reviews of movie websites; so they are not reliable using from one site as a few of the websites are producing the qualitative score showing high popularity of a movie and other ones are showing low popularity of the same movie. In study [21], deep learning is not used and only word frequency is used. Word frequency is no better way to evaluate the reviews semantically. It only produces the polarity of the term. Therefore, significant values and semantic information are required in an efficient way using the LSTM-attention learning algorithm for better semantic analysis, so semantic emotional value increases the significance of the recommendation system. Document semantic classification is improved by using the user movie attention at the word and sentence levels by the average pooling of word and sentence level to improve the semantic and emotion information

about reviews or document. The reason for applying the attention at the word level is to improve the semantic information of the document as compared to only applying it at the sentence level. Big data issue is covered by adopting the NoSQL environment.

8. Future Work

This work may be enhanced by adding more parameters like session, playlist, users group, session, smiley, tag, context, the feature of movie and video content to improve the work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. Ratnaparkhi, *Recommender System for Food in a Restaurant Based on Natural Language Processing and Machine Learning*, National College of Ireland, Dublin, Ireland, 2018.

[2] R. Mukherjee, P. S. Dutta, and S. Sen, *Movies2go—A New Approach to Online Movie Recommendation*, Department of Mathematical & Computer Sciences, University of Tulsa, Tulsa, OK, USA, 2001.

[3] U. Fasahte, D. Gambhir, M. Merulingkar, A. Monde, and A. Pokhare, "Hotel recommendation system, information technology, Atharva College of Engineering, India," *Imperial Journal of Interdisciplinary Research (IJIR)*, vol. 3, no. 11, 2017.

[4] P. M. Chawan, K. Suvarna, Y. Goyal, J. Thakker, and A. Bandiwadekar, "Recommendation system for dining, department of computer engg and info. tech. VJTI, Mumbai, Maharashtra, India," *Journal of Engineering Research and Application*, vol. 8, no. 5, pp. 48–52, 2018.

[5] J. Neidhardt, T. Kuflik, and W. Wörndl, "Special section on recommender systems in tourism," *Information Technology & Tourism*, vol. 19, no. 1–4, pp. 83–85, 2018.

[6] E. U. Okon, B. O. Eke, and P. O. Asagba, "An improved online book recommender system using collaborative filtering algorithm," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 41–48, 2018.

[7] J. A. Xu and K. Araki, "An SVM-based personal recommendation system for TV programs," in *Proceedings of the 12th International Multi-Media Modelling Conference*, Beijing, China, January 2006.

[8] S. Baluja, R. Seth, D. Sivakumar et al., "Video suggestion and discovery for YouTube: taking random walks through the view graph," in *Proceeding of the 17th International Conference on World Wide Web-WWW'08*, Beijing, China, April 2008.

[9] E. Aalipour and M. Ghazisaeedi, "Recommender system introduction for requests of cancer world," *International Journal of Community Medicine and Public Health*, vol. 4, no. 2, pp. 275–280, 2017.

[10] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 95–116, 2017.

[11] M.A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[12] J. Stan, F. Muhlenbach, and C. Largeron, "Recommender systems using social network analysis: challenges and future trends," in *Encyclopedia of Social Network Analysis and Mining*, pp. 1–22, Springer, Berlin, Germany, 2014.

[13] S. K. T. Lam, D. Frankowski, and J. Riedl, *Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems*, GroupLens Research Computer Science and Engineering, University of Minnesota Minneapolis, Minneapolis, MN, USA, 2006.

[14] K.-R. Kim, J.-H. Lee, J.-H. Byeon, and N.-M. Moon, *Recommender System Using the Movie Genre Similarity in Mobile Service*, Department of IT App. Tech. GSV, Hoseo University, Asan, South Korea, 2010.

[15] M.-Y. Hsieh, W.-K. Chou, and K.-C. Li, "Building a mobile movie recommendation service by user rating and APP usage with linked data on Hadoop," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3383–3401, 2017.

[16] K.-R. Kim and N. Moon, "Recommender system design using movie genre similarity and preferred genres in SmartPhone," *Multimedia Tools and Applications*, vol. 61, no. 1, pp. 87–104, 2012.

[17] M. Gao and X. Zhang, *A movie recommender system from tweets data*, January 2019, http://cs229.stanford.edu/proj2015/299_report.pdf.

[18] K. Badge and Jyoti Patil, "A survey on product recommendation systems using social data and microblogging information," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 12, no. 5, pp. 17513–17517, 2017.

[19] Y. Ishida, T. Uchiya, and I. Takumi, "Design and evaluation of a movie recommendation system showing a review for evoking interested," *International Journal of Web Information Systems*, vol. 13, no. 1, pp. 72–84, 2017.

[20] I. Vagliano, D. Monti, and M. Morisio, "SemRevRec: a recommender system based on user reviews and linked data," in *Proceedings of the RecSys 2017 Posters*, Como, Italy, August 2017.

[21] M. Ibrahim and I. Bajwa, "Design and application of a multivariant expert system using apache hadoop framework," *Sustainability*, vol. 10, no. 11, p. 4280, 2018.

[22] F. Ren and C. Quan, "Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing," *Information Technology and Management*, vol. 13, no. 4, pp. 321–332, 2012.

[23] K. Wakil, R. Bakhtyar, K. Ali, and K. Aladdin, "Improving web movie recommender system based on emotions," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, 2015.

[24] Y. Tom, D. Hazarikaz, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," 2018, https://arxiv.org/pdf/1708.02709.

[25] S. Postmus and S. Bhulai, *Recommender System Techniques Applied to Netflix Movie Data*, Vrije Universiteit Amsterdam, Amsterdam, Netherlands, 2018.

[26] P. T. Nguyen, P. Tomeo, T. Di Noia, and E. Di Sciascio, "Content-based recommendations via DBpedia and Freebase: a case study in the music domain," SisInf Lab, Polytechnic University of Bari, Bari, Italy, 2015.

[27] R. Mirizzi, T. Di Noia, A. Ragone, and V. C. Ostuni, "Movie recommendation with DBpedia," Polytechnic University of Bari, Bari, Italy, 2012.

[28] X.-Y. D. ZhenWu, C. Yin, S. Huang, and J. Chen, "Improving review representations with user attention and product attention for sentiment classification," 2018, https://arxiv.org/abs/1801.07861.

[29] R. Ferreira, O. Holanda, J. Melo, I. Ibert, F. Freitas, and E. Costa, *An Agent-Based Semantic Web Blog*, Federal University of Alagoas, Maceió, Brazil, 2012.

[30] W. W. Cohena and W. Fan, "Web-collaborative filtering: recommending music by crawling the Web," *Computer Networks*, vol. 33, no. 1–6, pp. 685–698, 2000.

[31] A. V. Jose and K. M. Jini, "Personalized movie recommender system using rank boosting approach on hadoop," *International Journal for Innovative Research in Science & Technology*, vol. 2, no. 2, 2015.

[32] G. Godhani and M. Dhamecha, "A study on movie recommendation system using parallel map reduce technology," *International Journal of Engineering Development and Research*, vol. 5, no. 1, 2018.

[33] B. Li, Y. Liao, and Q. Zheng, "Precomputed clustering for movie recommendation system in real time," *Journal of Applied Mathematics*, vol. 2014, Article ID 742341, 9 pages, 2014.

[34] L. Zhao, Z. Lu, S. Jialin, and Q. Y. Pan, "Matrix factorization+ for movie recommendation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, Hong Kong University of Science and Technology, Hong Kong Nanyang Technological University, New York, NY, USA, July 2016.

[35] J. Nessel and B. Cimpa, "The movie oracle-content based movie recommendations," in *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, EdgeWorks Software Ltd., Lyon, France, August 2011.

[36] D. Sánchez-Moreno, B. Ana, G. Gil et al., *A Collaborative Filtering Method for Music Recommendation Using Playing Coefficients for Artists and Users*, Department of Computing and Automation, University of Salamanca, Salamanca, Spain, 2016.

[37] K. Soni, R. Goyal, B. Vadera, and S. More, "A three-way hybrid movie recommendation system," *International Journal of Computer Applications*, vol. 160, no. 9, pp. 29–32, 2017.

[38] H. Khalid and S. Wu, "Reducing the cold-start problem by explicit information with mathematical set theory in recommendation systems," *International Journal of Engineering and Computer Science*, vol. 5, no. 8, pp. 17613–17626, 2016.

[39] A. Sarker, R. Ginn, A. Nikfarjam et al., "Utilizing social media data for pharmacovigilance: a review," *Journal of Biomedical Informatics*, vol. 54, pp. 202–212, 2015.

[40] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with Twitter: what 140 characters reveal about political sentiment," *ICWSM*, vol. 10, pp. 178–185, 2010.

[41] W. He, S. Zha, and L. Li, "Social media competitive analysis and text mining: a case study in the pizza industry," *International Journal of Information Management*, vol. 33, no. 3, pp. 464–472, 2013.

[42] E. L. Murnane and S. Counts, "Unraveling abstinence and relapse: smoking cessation reflected in social media," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1345–1354, ACM, Toronto, Canada, April 2014.

[43] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine, "Diamonds in the rough: social media visual analytics for journalistic inquiry," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122, IEEE, Salt Lake City, UT, USA, October 2010.

[44] T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. Wang, *How Noisy Social Media Text, How Different Social Media Sources?*, University of Melbourne, Melbourne, Australia, 2013.

[45] C. Corley, D. Cook, A. Mikler, and K. Singh, "Text and structural data mining of influenza mentions in web and social media," *International Journal of Environmental Research and Public Health*, vol. 7, no. 2, pp. 596–615, 2010.

[46] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, https://arxiv.org/abs/1506.00019.

[47] W. Yiny, K. Kanny, Y. Mo, and H. Schützey, "Comparative study of CNN and RNN for natural language processing," 2017, https://arxiv.org/abs/1702.01923.

[48] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[49] J. A. Konstan, "Introduction to recommender systems: algorithms and evaluation," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 1–4, 2004.

[50] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of net news," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, pp. 175–186, ACM, Chapel Hill, NC, USA, October 1994.

[51] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, pp. 210–217, ACM Press/Addison-Wesley Publishing Co., Denver, CO, USA, May 1995.

[52] A. Chang, J. F. Liao, P. C. Chang, C. H. Teng, and M. H. Chen, "Application of artificial immune systems combines collaborative filtering in the movie recommendation system," in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 277–282, IEEE, Hsinchu, Taiwan, May 2014.

[53] E. Reich, "Users are individuals: individualizing user models," *International Journal of Man-Machine Studies*, vol. 18, no. 3, pp. 199–214, 1983.

[54] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[55] S.-H. Min and I. Han, "Detection of the customer time-variant pattern for improving recommender systems," *Expert Systems with Applications*, vol. 28, no. 2, pp. 189–199, 2005.

[56] B. N. Miller, J. A. Konstan, and J. Riedl, "PocketLens," *ACM Transactions on Information Systems*, vol. 22, no. 3, pp. 437–476, 2004.

[57] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Group lens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[58] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert Systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[59] R. Schaefer, "Rules for using multi-attribute utility theory for estimating a user's interests," in *Proceedings of the ABIS Workshop, Adaptivität und Benutzermodellierung in*

interaktiven Softwaresystemen, Dortmund, Germany, October 2001.

[60] M. Paul, *Providing Actionable Recommendations: A Movie Recommendation Algorithm with Explanatory Capability*, Joseph Eul Verlag, Siegburg, Germany, 2013.

[61] Y. Zhang, M. Zhang, and Y. Liu, "Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 435–440, Shanghai, China, February 2015.

[62] I. S. Bajwa and S. Iqbal, "A semi supervised semantic analysis of natural language constraints using Drt and Markov logic," *Science International*, vol. 28, no. 3, pp. 2783–2790, 2016.

[63] X. Su and T. M. Khoshgoftaar, *A Survey of Collaborative Filtering Techniques*, Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA, 2009.

[64] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, New Orleans, LA, USA, February 2018.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[66] I. Bajwa, H. Ismail, H. Bukhari, and R. Amin, "Automated sentiment analysis of natural language text using machine learning," *Sindh University Research Journal-Surj (Science Series)*, vol. 48, no. 3, 2016.

[67] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1555–1565, Baltimore, MD, USA, June 2014.

[68] R. Socher, A. Perelygin, J. Y. Wu et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Empirical Methods in Natural Language Processing*, p. 1642, Seattle, WA, USA, October 2013.

[69] U. Ghani, I. Bajwa, and A. Ashfaq, "A fuzzy logic based intelligent system for measuring customer loyalty and decision making," *Symmetry*, vol. 10, no. 12, p. 761, 2018.

[70] K. Yoon, "Convolutional neural networks for sentence classification," in *Proceedings of the EMNLP*, Doha, Qatar, October 2014.

[71] Z. Yang, D. Yang, C. Dyer, X. He, Alex Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, June 2016.

[72] Y. Wang, M. Wang, and W. Xu, "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8263704, 9 pages, 2018.

[73] S. Kumar, S. S. Halder, K. De, and P. P. Roy, "Movie recommendation system using sentiment analysis from microblogging data," 2018, https://arxiv.org/abs/1811.10804.

[74] H.-T. Zheng, J.-Y. Chen, N. Liang, A. K. Sangaiah, Y. Jiang, and C.-Z. Zhao, "A deep temporal neural music recommendation model utilizing music and user metadata," *Applied Science*, vol. 9, no. 4, p. 703, 2019.