

Article

# Design and Application of a Multi-Variant Expert System Using Apache Hadoop Framework

Muhammad Ibrahim \* and Imran Sarwar Bajwa 

Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan; imran.sarwar@iub.edu.pk

\* Correspondence: ibrahimbwp@gmail.com

Received: 20 October 2018; Accepted: 12 November 2018; Published: 19 November 2018



**Abstract:** Movie recommender expert systems are valuable tools to provide recommendation services to users. However, the existing movie recommenders are technically lacking in two areas: first, the available movie recommender systems give general recommendations; secondly, existing recommender systems use either quantitative (likes, ratings, etc.) or qualitative data (polarity score, sentiment score, etc.) for achieving the movie recommendations. A novel approach is presented in this paper that not only provides topic-based (fiction, comedy, horror, etc.) movie recommendation but also uses both quantitative and qualitative data to achieve a true and relevant recommendation of a movie relevant to a topic. The used approach relies on SentiwordNet and tf-idf similarity measures to calculate the polarity score from user reviews, which represent the qualitative aspect of likeness of a movie. Similarly, three quantitative variables (such as likes, ratings, and votes) are used to get final a recommendation score. A fuzzy logic module decides the recommendation category based on this final recommendation score. The proposed approach uses a big data technology, “Hadoop” to handle data diversity and heterogeneity in an efficient manner. An Android application collaborates with a web-bot to use recommendation services and show topic-based recommendation to users.

**Keywords:** recommender systems; opinion mining; SentiWordNet; polarity scores

## 1. Introduction

Since the advent of web intelligence, artificial intelligence-based services, frameworks and products have become popular in the World Wide Web. One of the key services of such web intelligence applications is a recommendation system. In recent times, recommendation systems have become popular in the domain of movies, music, books, restaurants, garments, mobile applications and many other fields of life. Such recommendation systems filter huge amount of structured and unstructured data and predict the preference of a user that one would give to an item. In the last decade, a few movie recommendation systems have been presented using conventional methods [1–3]. However, previous movie recommender systems lack various features and/or accuracy of true recommendation. A majority of these recommender systems use quantitative variables (likes or ratings) and a few others use qualitative variables (polarity score, etc.) [4–7]. This paper proposes an intelligent and automated recommendation system that provides two-fold novelty. First, our recommender system uses a multi-variant popularity matrix to recommend a suitable movie to a user on the basis of both quantitative and qualitative variables to achieve true recommendations. Secondly, a fuzzy logic-based module provides the final recommendation of movies in a particular field of user’s choice (such as comedy, action, horror, fiction, etc.), whereas the currently available systems give general recommendations.

In our multi-variant recommendation system, one of the challenges was opinion mining of users’ reviews to calculate a polarity score that shows the degree of likeness or dis-likeness of a movie by



During the literature review of modern recommender systems, Hsieh's movie recommender system [13] was identified as the relevant. This system has used the benefits of big data solutions and also provides a mobile app to interact with the recommender system. However, the key short-coming in this work is the limited approach used to recommend movies. Major issues with this recommendation system are discussed and comparison with our approach is given in Table 2.

**Table 2.** Difference to Hsieh's work [13].

#	Hsieh's Work	Our Approach
1	It is a general recommendation system for movies.	Our approach supports topic wise recommendation of movies such as drama, comedy, action, horror, etc.
2	This approach only uses quantitative data (ratings and likes) for recommendation that provides less accuracy. .	Our approach uses both quantitative (votes, likes, etc.) and qualitative data (polarity score) for true recommendation of movies.
3	This approach banks on simplistic calculation in the base of similarity measures. No real decision making approach is used that makes quality of results questionable.	Our approach uses Fuzzy Logic approach for better decision making and true recommendations of movies.
4	True likeness of users is not reflected by this approach.	Our approach reflects true likeness of the users as qualitative aspects of likeness is also considered.
5	This approach is tested only on one limited dataset.	Our approach is tested on three large datasets.
6	This approach calculates recommendation on two quantitative variables.	Our approach calculates recommendation on three quantitative variables and one qualitative variable.

The recommender systems field has made significant progress with many new techniques proposed and new systems developed. However, modern systems still require significant improvements to provide better recommendations. The major contribution to knowledge and novelty of the work is outlined below:

- i. A topic (action, comedy, horror, etc.) based recommendation is supported.
- ii. Multi-variant (ratings, votes, likes and polarity score) parameters are used.
- iii. Both quantitative and qualitative data is used for movie recommendations.
- iv. Three external data resources are used for datasets (Metacritics, IMDB, and Fandango).
- v. A web-bot used to fetch the web contents collaborates with the server.
- vi. Filters and integrates movie descriptions from linked data or ontology (linkedmdb).
- vii. Recommender system is developed using NoSQL environment with apache Hadoop.
- viii. Fuzzy sets are established for movie ranking categorization.
- ix. Front end App collaboration with the movie recommender through web services is supported.

The rest of this paper is organized as follows. In Section 2 related work is discussed, where recommender systems running for different subjects could clear up the native problems of user's data processing. A multi-variant ranking model is presented, for movie recommender using a mobile application, and Apache Hadoop in Section 3. Experiments and results, and evaluation of the system are discussed in Sections 4 and 5, respectively. The conclusion and future works are presented in Sections 6 and 7, respectively.

## 2. Related Work

Recommendation services typically rely on customer reviews or customer ratings and such recommendations can provide a useful service for new customers. Emotional expressions,

social interaction and behavior changes of the users are studied on the Twitter, allowing management to distinguish clients who do or do not return [20]. Vox Civitas obtains responses from social media (e.g., Twitter), which can support journalistic investigations in more effective ways [21]. Anomalies are removed and pure data is obtained which reflect the United Kingdom's worst influenza [22]. Detection of noise in text (tweets) from micro-blogs is discussed in [23].

Sentiment analysis approaches can be used to extract sentiments associated with positive or negative polarities for specific subjects from a document, instead of classifying the whole document as positive or negative [24]. An NLP-based methodology of sentiment evaluation on user's comment has been used as a way to retrieve the best and perfect YouTube videos. The process works in four steps. First, a review collection and preprocessing component extracts data (comments) from the particular YouTube video and language preprocessing is undertaken to prepare for the next process. Second, the processed text goes through NLP-based methods to generate data sets. Subsequently, the sentiment classifier (Sentistrength) is applied on the data sets to calculate the positivity and negativity ratings. Finally, the standard deviation applied to get the rating result [25].

Features level sentiment analysis, which is based on the idea that an opinion consists of a sentiment (positive or negative) and a feature of movies is another approach. Each short comment is represented as a sequence of sentiment words and underlying states [9,12]. A linear regression model (LRM), a supervised machine learning technique to classify twitter gossip (positive and negative) has been used to predict the box-office revenue for different movies [8]. Neural networks (NNs) classification of sentiment analysis of large movie reviews has been handled by introducing a method. Recursive neural networks wrap the previous sentence-level-sentiment classification and are used with recurrent neural networks. Recursive neural networks are used for sentence-level analysis and a recurrent neural network is used for whole passage analysis to create better results [26].

The vector space model (VSM) was used to implement the instance-based learning (IBL) classification method. Text documents were treated as vectors in IBL algorithms to identify the class (positive or negative review) of the document [27]. Sentiment analysis is negative when the text includes some negative words, such as "bad acting, stilted dialog." It is positive if the text includes some positive words such as "it's funny". A suggestion instead of an exact rating is done by sentiment classification of the comments (polarity), and then aggregated into a rating score selected as the recommended list of popular movies [11,17]. For example, the hotel management of Starwood Hotels and Resorts use social media's strength to stay connected with their guests, to guide them, and seek responses to the services they provide. [28].

Other related work includes typical recommender frameworks that construct calculations in light of different fuzzy set theoretic likeness measures (the fuzzy set augmentations of the Jaccard list, cosine, closeness or relationship similitude measures), and aggregation techniques for figuring suggestion certainty scores (the maximum-minimum or weighted-whole fuzzy set theoretic accumulation strategies) for recommendation [4]. The strategy for ranking in light of the content involves building a sentiment graph from the collocation of adjectives, PageRank algorithm and a very small set of adjectives (such as 'good', 'excellent', etc.) that rank different movies using reviews of box office movies by users of a popular movie review site [18]. With regard to the utilization of labels with the end goal of recommendation of movies, the German movie website, Moviepilot uses viewers, and movie ratings, and all out labels are marked to every movie. Labels are allotted by a group of moderators and viewer are then able to rate how well the labels fit every motion picture [3]. This collaborative filtering was first applied elsewhere in filtering the information in Usenet news [29].

Music recommender systems provide personalized music recommendations and Ringo Agent was one of the first applications [30]. Content-based filtering recommends movies based on a comparison between user profile data and content of movies. Content-based filtering is also called cognitive-filtering. The recommendations are generated by matching users and movie content [4]. Collaborative filtering is also called social filtering. The fundamental rule behind collaborative filtering is that if a user likes a certain category of movie in the past, then they may like similar movies in the

future. This information is used in deciding which movie to suggest [19,29,30]. Hybrid filtering is a combined technique of content filtering and collaborative filtering [31].

The previous work discussed above suggests that most of the approaches used for recommendation services, especially for movie recommendation are uni-variant and use variables such as ratings, which tend to provide results with low accuracy. There are other variables including likes, number of reviews, and the sentiment score of a review that can help in achieving an accurate and efficient recommendation, and in this paper we aim to use these new variables for the proposed movie recommendation service.

### 3. Multi-Variant Expert System

The proposed approach works on the fetched data (scores and reviews) from a set of movie websites and databases. The collected data is heterogeneous in nature, such as numeric data (for example, number of votes, number of likes and number of ratings) and text data (for example, user reviews of movies). A web crawler was developed to fetch structured and unstructured data and store the fetched data in a NoSQL database on a server machine for further processing.

The used approach works in two parallel streams. In the first stream, the text data (such as movie reviews) is preprocessed using NLP modules, tf-idf algorithms and the SentiWordNet auxiliary database to identify polarity scores of terms (lexicons) in the form of negative and positive scores. All the movies reviews are processed for an aggregate polarity score for each movie from each participating external data source. In the second stream, all the numeric scores and weights (rating, votes and likes) of the movies are normalized and computed to achieve weighted aggregate of polarity scores. The result of a search query of the movies is shown in the user interface of an Android app (see Figure A1). The user query interacts with the server and the server processes the request by forwarding it to the web crawler. The web crawler module responds to the server's request by crawling the web for keywords (lexicons) matching and downloading the webpages to the server, and then the server processes the data to generate a recommendation as shown in Figure 1.

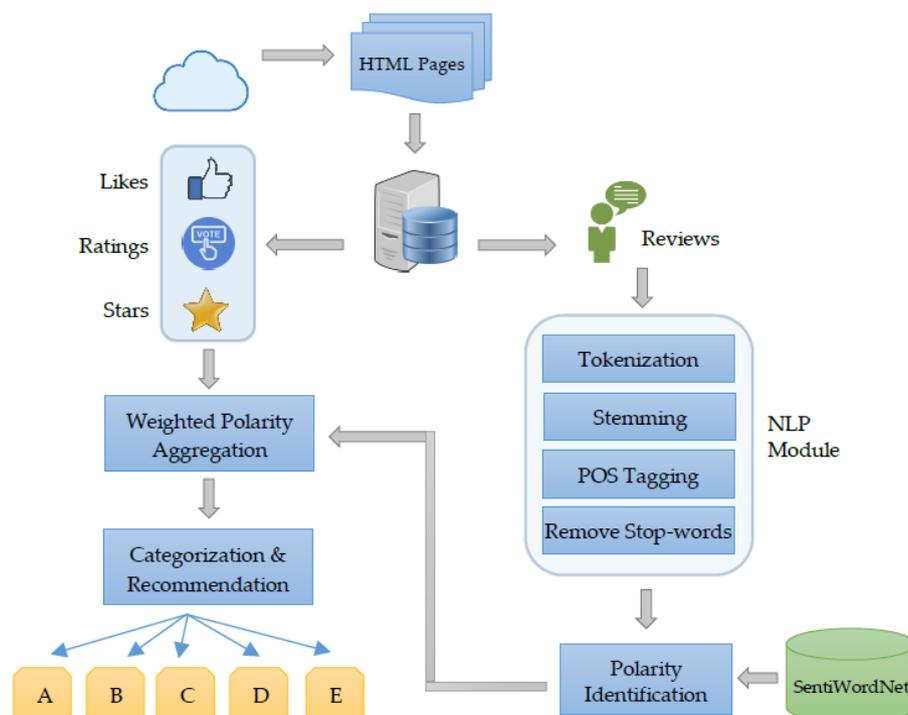


Figure 1. Multi-variant expert system for movie recommendation.

### 3.1. NLP Module

Real-world data is generally incomplete and noisy, and is likely to contain irrelevant and redundant information or errors. By pre-processing, raw unstructured data can be converted into a structured, understandable form as shown in Figure 2. Since the real-world data can contain ambiguity and anomalies, it is necessary to remove these abnormalities before the actual analysis of data. The data was pre-processed to remove anomalies and to identify the abbreviated language of the actual English and also remove the reviews which were in other languages other than English [32–34].

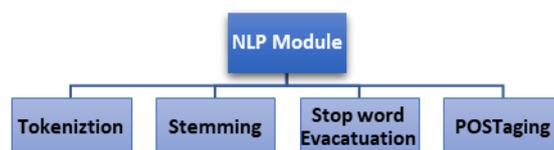


Figure 2. Preprocessing data using NLP module.

#### 3.1.1. Tokenization

Then, the given character stream is separated into units called tokens. The tokens might be words or numbers or highlighting check. Tokenization does this by finding word limits. For example, “The message of this film is simple” string is tokenized as [The] [message] [of] [this] [film] [is] [simple] [35,36].

#### 3.1.2. Stemming (Lemmatization)

This is optional; most stemming uses the Porter Stemmer. English words like “look” can be arched with a morphological suffix to deliver “looks, looking, looked”. These have a similar stem, “look” [37,38].

#### 3.1.3. Stop Word Evacuation

Most regularly used words do not convey much significance. For example: “the, an, of, for, in ...”. We used a small corpus based library to exclude stop words from the input data. This library is developed in Java.

#### 3.1.4. POS-Tag Generation

The query was then analyzed and POS tags were generated of all the words in the query. Then, the resulting string of words and their relevant POS (Parts of Speech) tags are tokenized on the basis of space. A good example of an English sentence, “This movie is so riddled”, is Pos-tagged as [this/DT movie/NN is/VBZ so/RB riddled/JJ]. The Treebank Project (URL: <https://catalog.ldc.upenn.edu/docs/LDC95T7/treebank2.index.html>) shows 36 POS-tags [39], e.g., determiner [DT], adjective [JJ] and adverb [RB], etc.

### 3.2. Polarity Computation

#### 3.2.1. Lexical Frequency Measuring

For this purpose, we applied a tf-idf frequency measure [40,41]; we first calculated the frequency of the valid/important terms, which represents the number of times that term ‘t’ occurs in the document (review) ‘d’, as in the following Equation (1):

$$tf(t, d) = f(t, d) \quad (1)$$

After calculating  $tf$  we calculated  $idf$  (inverse document frequency) of the terms to obtain information about how rare or common that term is in the documents (reviews). We used the Equation (2):

$$idf(t, D) = \log N / dt \quad (2)$$

where as  $d \in D$  and  $t \in d$ ,  $N$  is total number of documents in the corpus  $N = |D|$ . The end results are then obtained by applying the Equation (3):

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (3)$$

### 3.2.2. Polarity Identification

SentiWordNet 3.0 automatically annotates all WordNet 2.0 (synsets) according to their degrees of positivity, negativity and neutrality. In this step, the SentiWordNet score was used in the sentiment analysis of the documents (reviews) [42,43]. For this purpose, we applied the Equation (4):

$$\text{Polarity\_term\_score} = \text{SentiWordNetScore} * \text{Frequency}(tfidf) \quad (4)$$

The SentiWordNetScore (positive or negative) of the term and its frequency were computed to get the overall sentiment of the terms in the documents.

The SentiWordNetScore of each term for all reviews of the movie is calculated and the score (negative or positive) tells us how many terms are positively or negatively important in the review. Then, all the positive terms scores are added to obtain the positive term's weight, and also all the negative terms scores are combined to obtain the negative term's weight in a review. The polarities of all the reviews of the movies from each participating website are calculated as follows.

#### Polarity of a Term

By applying sign  $P(t)$  on term, if SentiWordNetScore ( $S_{t_i}$ ) of terms ( $t_i$ ) of the review ( $r_i$ ) of movie ( $m_i$ ) is less than zero then the term lies in the negative poll (nt), if greater than zero than it lies in the positive poll (pt) and if it is equal to zero than it lies in the neutral poll (tn) as shown in equation (5).

$$P(t) = \begin{cases} -S_{t_i}, & S_{t_i} < 0 & (\text{nt} = \text{negative term}) \\ S_{t_i}, & S_{t_i} = 0 & (\text{tn} = \text{neutral term}) \\ +S_{t_i}, & S_{t_i} > 0 & (\text{pt} = \text{positive term}) \end{cases} \quad (5)$$

#### Polarity of a Document (Reviews)

For calculating the polarity of a document (review)  $\text{polarity}_{r_i}(r)$ , positive terms  $\text{pt}_{r_i}$  and negative terms  $\text{nt}_{r_i}$  are aggregated for each document (review) from each participating websites'  $\text{negative\_term}_{r_i}(x)$  and  $\text{negative\_term}_{r_i}(y)$  and then take their differences are taken to find the polarity of each documents (reviews) by applying the sign function  $f(r)$  as shown in Equations (6) and (7).

$$\text{positive\_term}_{r_i}(x) = \sum_{i=0}^n \text{pt}_{r_i} \quad (6)$$

$$\text{negative\_term}_{r_i}(y) = \sum_{i=0}^n \text{nt}_{r_i} \quad (7)$$

where as  $\text{pt}_{r_i} \wedge \text{nt}_{r_i} \in r_i$  and  $r_i \in m_j$

Polarity of a review is calculated as shown in Equations (8) and (9):

$$\text{polarity}_{r_i}(R) = \text{sgn} \left[ |x_{r_i}| - |y_{r_i}| \right] \quad (8)$$

$$f(r) = \begin{cases} -p_{r_i}, & p_{r_i} < 0 & (\text{nr} = \text{negative review}) \\ p_{r_i}, & p_{r_i} = 0 & (\text{rn} = \text{neutral review}) \\ +p_{r_i}, & p_{r_i} > 0 & (\text{pr} = \text{positive review}) \end{cases} \quad (9)$$

by applying sign  $f(r)$  on each review, If the difference of aggregated positive\_term $_{r_i}(x)$  of review ( $r_i$ ) of the movie ( $m_j$ ) from website ( $w_k$ ) and aggregated negative\_term $_{r_i}(y)$  is less than zero then review is sentimentally lie in negative poll (nr), if greater than zero than lie in positive poll (pr) and if equal to zero than lie in neutral poll (rn).

#### Polarity of a Collection (Movie Reviews)

review\_positive\_score $_{m_j}(a)$  is the aggregated polarity score of positive reviews pr and review\_negative\_score $_{m_j}(b)$  is the aggregated polarity score of negative reviews nr of a particular movie  $m_j$  from a particular website ( $w_k$ ) used to calculated the polarity of the movie from participating sites as given in Equations (10) and (11).

$$\text{review\_positive\_score}_{m_j}(g) = \sum_{i=0}^n pr_i \quad (10)$$

$$\text{review\_negative\_score}_{m_j}(h) = \sum_{i=0}^n nr_i \quad (11)$$

where as  $pr_i \wedge nr_i \in m_j$  and  $m_j \in w_k$ .

Polarity of a collection is calculated using Equation (12):

$$\text{polarity}_{r_i}(p) = \text{sgn} \left[ \left| g_{m_j} \right| - \left| h_{m_j} \right| \right] \quad (12)$$

where as  $g_{m_j} \wedge h_{m_j} \in m_j$  and  $m_j \in w_k$ . Here  $w_k$  is a movie website such as IMDB.

#### 3.3. Weighted Polarity Manipulation

Opinion mining determines the emotions (positive or negative) of textual communication on social media, and examines the positive or negative emotions by simply extracting polarity scores from the review (number of stars or thumbs up/down and votes etc.). However, we used both the polarity score and weight score (rating, votes and likes) of the movies. First, we computed the aggregated polarity score of each movie from each participating site, and then we took the average of the aggregated polarity by total reviews of the respective movie and their site. Again, we take aggregation of average polarity score. Also, total likes of the movie were combined with weighted\_average\_polarity to find the aggregated\_weighted\_average\_polarity. After that, the final score of the movie was rescaled to get the ranked score and category of the movie. In this computation, Equations (13)–(18) are used.

$$\text{aggregate\_polarity\_Score}_{m_j}(g) = \sum_{i=1}^n p_i \quad (13)$$

whereas  $g_{m_j} \in m_i$ ,  $m_i \in w_k$ .

$$\text{weight}_{m_j} = \left[ (\text{votes}_{w_k}) + (\text{rating}_{w_k}) \right] \quad (14)$$

whereas  $\text{weight}_{m_j} \in m_i$ ,  $m_i \in w_k$ .

$$\text{weighted\_average\_polarity}_{m_j}(a) = \frac{g_{m_j}}{n} + (\text{weight}_{m_j}) \quad (15)$$

whereas  $n$  is number of reviews of movie ( $m_j$ ) from movie website ( $w_k$ ).

$$\text{aggregated\_weighted\_average\_polarity}_{m_j}(G) = \sum_{k=1}^N a_{w_k} + \text{likes}_{m_j} \quad (16)$$

$$\text{average\_aggregated\_weighted\_average\_polarity}_{m_j}(A) = \frac{G_{m_j}}{N} \quad (17)$$

where as  $a_{w_k} \in m_j$ ,  $m_j \in w_k$ ,  $w_k \in N$ .

( $N$ ) is the number of movie websites (Metacritic, IMDBand Fandango) which has huge collection of material.

$$\text{Rank\_score}_{m_j}(R) = \frac{r_{m_j} - \min(r_{m_j})}{\max(r_{m_j}) - \min(r_{m_j})} * 10 \quad (18)$$

Here  $R$  is the rescale value of the normalized average aggregated score ( $r_{m_j}$ ) of the movie ( $m_j$ ) from movie websites ( $N$ ) to rank the top five movies ( $M$ ).

### 3.4. Categorization

Movie genres are various forms or identifiable types, categories, classifications or groups of movies (genre comes from the French word meaning “kind”, “category”, “or “type”). <http://www.filmsite.org/filmgenres.html>.

The user can directly query for content (e.g., “news London”, “golf 1940”, “documentary Alfred Hitchcock”, “movies tonight”). The querying is done on different fields describing the content (e.g., title, creator, year, genre, language, location). The search results are presented as organized according to concepts from common vocabularies (e.g., Time Ontology, Geo Ontology, WordNet) [29,44–46].

#### 3.4.1. WordNet

The WordNet library was used in our approach to find synonyms and alternative forms of query terms, e.g., “weather” = {“weather report”, “weather forecast”, etc.}. Identification of synonyms in data helps in obtaining better and more accurate results.

#### 3.4.2. Geo Ontology

Finds related geographical areas, e.g., “London” = {City of London, Camden, Westminster, Greenwich, Greater London, England, and UK}.

#### 3.4.3. Time Ontology

Determines temporal context e.g., “tonight” = {18:00–24:00}, or “this week” = {10/02–17/02}.

#### 3.4.4. TVA-CS

Finds related genres (e.g., “sports” = {sport reports, sport live, sport news, sport documentary, football game, etc.} Documentary, football game, etc.).

### 3.5. Recommendation

The final recommendation is achieved by using the fuzzy logic approach on the following fuzzy set to evaluate the final score and find the category of the movie as follows.

**Step 1** if final score  $\geq 8$  then Category: “A: Recommended”

**Step 2** else if final score  $\geq 6$  then Category: “B: Top Recommended”

**Step 3** else if final Score  $\geq 4$  then Category: “C: Recommended Average”

**Step 4** else if Final Score  $\geq 2$  then Category: "D: Least recommended"

**Step 6** else Category: "F: Not recommended"

Figure 3 shows the final recommendations of the movies in a particular category (such as comedy, horror, fiction, etc.) in one of the five different classes. The user interface showing the output is discussed in Appendix A and Figure A1.



Figure 3. Multi-variant ranked category.

#### 4. Experimental Setup

##### 4.1. NoSQL for Big Data Storage

The number of publicly available test corpora is quite limited and comparatively of small size with respect to the number of texts documents in a corpus. Thus, producing adequately precise comparisons between reported performances is difficult. So, we decided to build a new corpus and for this purpose, we used three different external data source websites to extract a large number of reviews, votes, ranking and likes. The data for our corpus was retrieved by a web bot implemented in PHP. We wrote a webpage (web-bot) scraping scripts which extract movie URLs with matching user’s queries, if the query keywords (lexicons) are matched then the crawler downloads the page in a server machine NoSQL environment using Hadoop, otherwise this page is discarded [13,14,47–52]. This procedure is depicted in Figure 4. The process of data extraction uses the following steps.

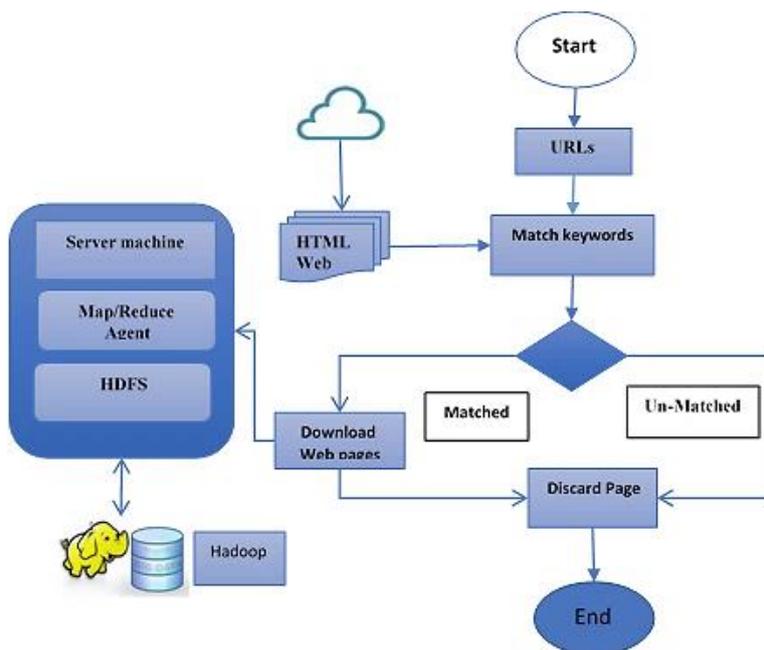
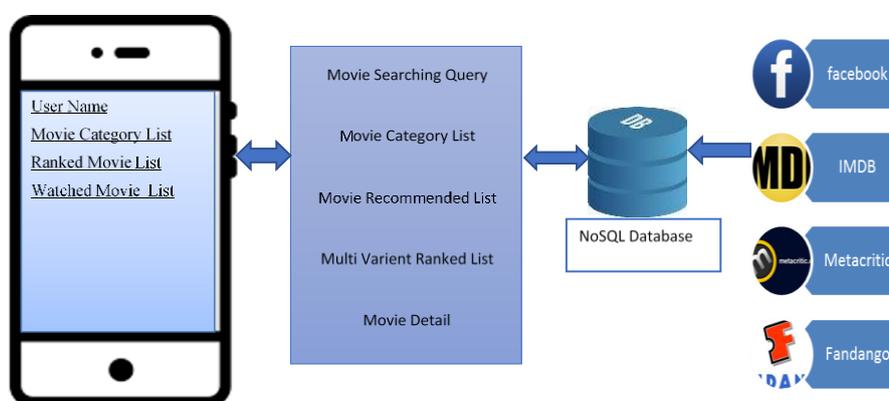


Figure 4. Data collection process from websites using Web Crawler.

- Step 1** Receive URLs for a movie type i.e., comedy, horror, fiction, etc.
- Step 2** Matches the keywords from the query to the page If
- Step 3** Keywords matched Then
- Step 4** Download the web page
- Step 5** Send it for storage
- Step 6** Discard the page
- Step 7** Repeat the step 2 to 6 until all the matched web pages are found.

Web crawler (web-bot) downloads the webpages (crawled pages) by which it extracts more contents (Meta tags) like movie reviews, rating, votes and likes and other irrelevant pages discarded as shown in Figure 4.

Computational processing can occur on data stored either in a file-system (unstructured) or in a database (structured) as shown in Figure 5. Apache Hadoop is the de facto data operating system. It is an open-source software framework processing of big data on clusters of commodity hardware.



**Figure 5.** Multi-variant recommendation system.

A multi-variant web agent is implemented in the Hadoop environment to handle big data generated by recommendation systems in order to improve the scalability and efficiency. The above-mentioned Figure 5 shows the interaction and computation in the Hadoop environment between the Android user app, web bot and the external participating sites for data. In each Mapper, there are various algorithms porterStemmer(), tokenizer(), POStager(), polarityComputation(), weightedRanking(), Webcrawler(), etc. and in Table A1, the Server machine's specification and Android device specification are presented. The hardware used in implementation is discussed in Appendix B and mentioned in Table A1.

#### 4.2. Experiment and Results

We used three repositories whose reviews should be trusted, that is, IMDB, Metacritic and Fandango, which contain all the required data (reviews and scores). These repositories contain movie (2016 and 2017) data for 1000 of the most popular movies (with a significant number of votes and ratings) and their reviews were released in 2016 and 2017, and as of 22 March 2017.

We computed the polarity score of the text data (reviews) by computing the movie's reviews corpus which was fetched from each participating external data source sites. This procedure was followed by data preprocessing, tf-idf classification and polarity identification using SentiWordNet to compute the polarity scores of each term of each document from each participating data source sites. The following tables illustrate the values of movies fetched data which were processed and evaluated.

Here, Table 3 shows the movie's title and the corresponding movie ID as follows.

**Table 3.** Movie Id of movies used in the experiments.

Movie Title	Movie ID
Avengers: Age of Ultron (2015)	m1
Cinderella (2015)	m2
Ant-Man (2015)	m3
Do You Believe? (2015)	m4
Hot Tub Time Machine 2 (2015)	m5

Table 4 shows the popular external data source sites and their corresponding movies sites ID for better formulation.

**Table 4.** Movie database sites ID.

Movie Database Site	Movie Database Sine ID
Metacritic	w1
IMDB	w2
Fandango	w3

Some computed values such as polarity scores of movie reviews from participating sites, which are already labeled are shown in Table 5.

**Table 5.** Calculated polarity cores.

Movie ID	Polarity Scores		
	w1	w2	w3
m1	21	29	3
m2	−17	16	26
m3	2	2	4
m4	−2	2	13
m5	1	19	7

Here we normalized the scores “likes” metacritic and IMDB rating scores are normalized ratings to a (0–5) scale because Metacritic and IMDB user rating is out of ten stars, but Fandango rating is out of five stars so we normalized the Metacritic and IMDB to five stars. These normalized values are given in the Table 6.

**Table 6.** Normalized and un-normalized rating.

Movie ID	Un-Normalized Rating Scores			Normalized Rating Scores		
	w1	w2	w3	w1	w2	w3
m1	7.1	7.8	5	3.55	3.9	5
m2	7.5	7.1	5	3.75	3.55	5
m3	8.1	7.8	5	4.05	3.9	5
m4	4.7	5.4	5	2.35	2.7	5
m5	3.4	5.1	3.5	1.7	2.55	3.5

Here Metacritic\_votes, IMDB\_votes and Fandango\_votes are the number of the votes, which are allotted by users to the particular movies from specific movie websites and are represented in the Table 7.

After calculating and aggregating the polarity score from each participating movie site and taking an average of polarity scores by total movies, and taking the weighted average polarity by adding the weights (normalized ranking and votes) to each average polarity, Likes may also represent the

behavior of users, which impact the movie rating. That is why we selected the likes in our model to present the multi-variant approach. We added Facebook likes to them to take the aggregated weighted average polarity for better recommendations. The classified scores are shown in Table 8.

Table 7. Movie votes.

Movie ID	Movie Votes		
	w1	w2	w3
m1	1330	271,107	14,846
m2	249	65,709	12,640
m3	627	103,660	12,055
m4	31	3136	1793
m5	88	19,560	1021

These multi variants (votes, ranking and likes) are computed according to our model, which ranked the movies as mentioned from the corpora of different movie data, the final score and category are represented in Table 9.

The multi-variant score of m1 movie “Avengers: Age of Ultron” is greater than six which is why it is categorized “B”, the m2 movie “Cinderella” score is greater than four so it lies in Category “C”, m3 and m4 “Ant-Man” and “Do You Believe?”, respectively, are greater than six so these are categorized “B”, and the m5 “Hot Tub Time Machine 2” movie score is greater than four so it is also lies in “C” category. Figure 6 represents the final ranking score of a movie’s particular category such as fiction, horror, drama, etc.

Figure 7 shows the rating scores of a particular movie from three specific movie websites We compared their normalized rating scores among these sites, and we observed that rating by Fandango is so high that is not significant individually.

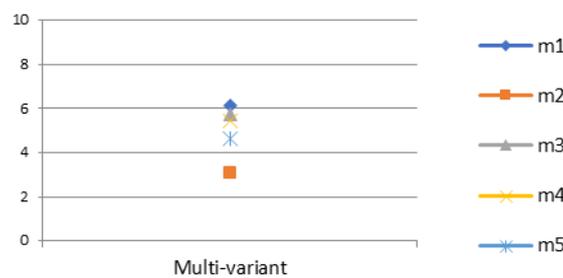


Figure 6. Multi-variant movie ranking.

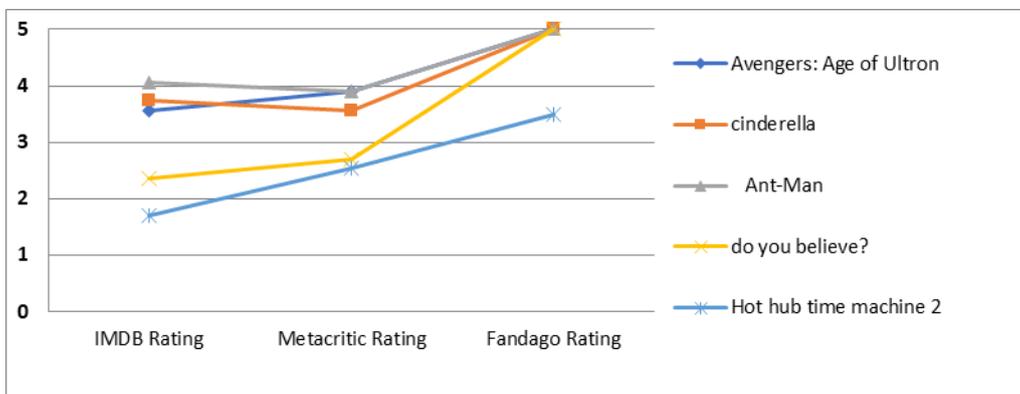


Figure 7. Rating difference.

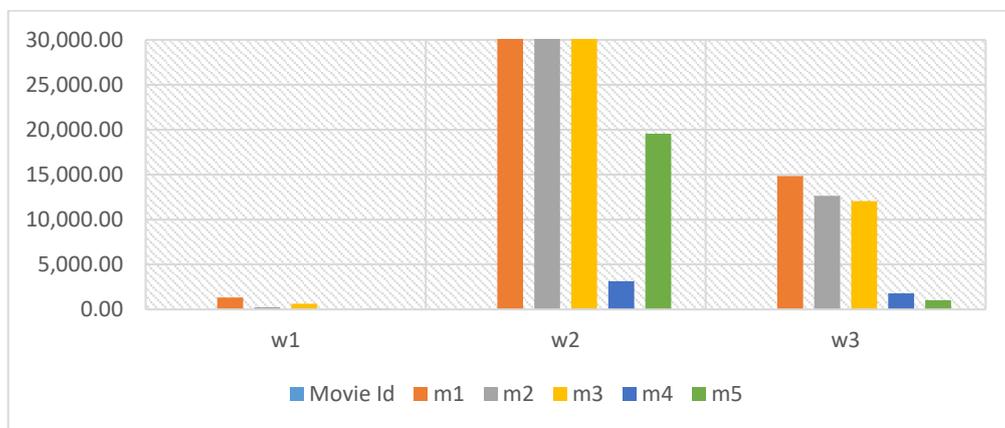
Table 8. Weighted average polarity scores.

	Reviews			Aggregated Polarity			Average Polarity			Weighted Average Polarity		
	w1	w2	w3	w1	w2	w3	w1	w2	w3	w1	w2	w3
Movie ID	Reviews	Reviews	Reviews	Aggre. Polarity	Aggre. Polarity	Aggre. Polarity	Average Polarity	Average Polarity	Average Polarity	Weighted Average Polarity	Weighted Average Polarity	Weighted Average Polarity
m1	66	1168	30	21	29	3	0.318	0.0248	0.1	1333.86	271,110	14,851.1
m2	67	363	27	−17	16	26	0.253	0.0440	0.962	252.49	65,712.5	12,645.9
m3	64	605	24	2	2	4	0.031	0.003	0.166	631.08	103,663	12,060.1
m4	22	69	22	−2	2	13	0.090	0.0289	0.590	33.25	3138.72	1798.59
m5	29	101	20	1	19	7	0.034	0.1881	0.35	89.73	19,562.7	1024.85

**Table 9.** Final scores and movie category.

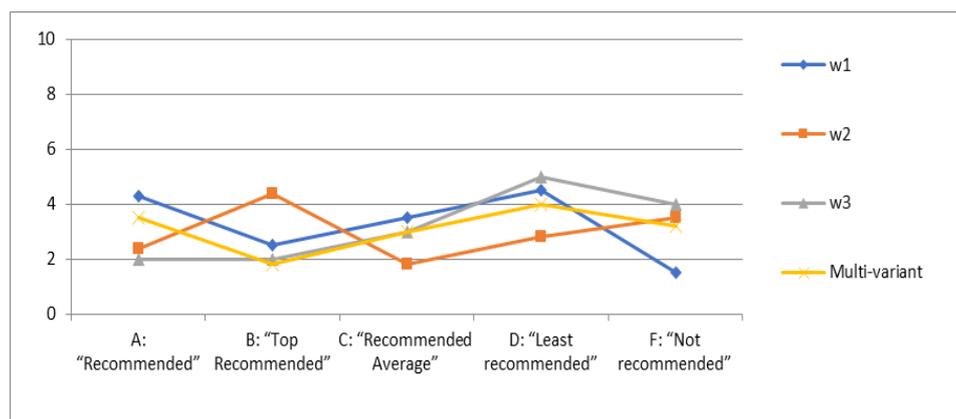
Movie ID	Likes	Aggregated Weighted Average Polarity	Final Ranking Score	Movie Category
m1	308,130	403,895.2977	7.44944873	B
m2	331	26,534.68444	5.878236708	C
m3	140,000	178,785.0504	6.979146632	B
m4	97,000	98,656.85966	6.636052698	B
m5	14,000	20,892.44087	5.740277373	C

In Figure 8, voting for movies also represents the user’s interest in movies from different participating websites, which indicates a huge difference if we select only one site. One site is not adequate for a ranking approach, which is the reason we selected multi-variants from different sites.



**Figure 8.** Movie votes difference.

Here, Figure 9 represents the differences in weighted average polarity scores by computing the multi-variant to show the categories.



**Figure 9.** Comparison scores of movies.

Time complexity was computed and also the watched time at different machines was observed. Time complexity in the worst case of our approach is  $O(n)$  because  $n$  number of datasets are used, and all following operation take one unit time, so time complexity of following is operations  $O(1)$ . The computation watched time details are presented in Table 10.

**Table 10.** Computational watched time at different machines.

CPU Clock	Move	Add.	Sub.	Mul.	Div.	Comp.	Speed
TMS320C30 (16.67 MHz)	22	3n	4	3	4	19	6 ms
MC68000 (16 MHz) 5 20 20 70 160 15 59 ms	22	3n	4	3	4	19	59 ms
Z80 (8 MHz)	22	3n	4	3	4	19	280 ms

## 5. Evaluation

Recall is defined as the number of relevant movies retrieved by a search divided by the total number of existing relevant movies, while precision is defined as the number of relevant movies retrieved by a search divided by the total number of movies retrieved by the search. The precision is the proportion of recommendations that are good recommendations,

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp}) \quad (19)$$

and recall is the proportion of good recommendations that appear in top recommendations.

$$\text{Recall} = \text{tp}/(\text{tp} + \text{fn}) \quad (20)$$

tp: predicted positive interested movie it is true, it is really interested.

tn: predicted positive uninterested movie it is true, it's really uninterested.

fp: predicted positive interested movie but wrong, it is actually interesting.

fn: predicted negative uninterested movie but wrong, it is actually uninteresting.

In the recommendation domain, a perfect precision score of 1.0 means that every movie recommended in the list was good (although this says nothing about if all good recommendations were suggested) whereas a perfect recall score of 1.0 means that all good recommended movies were suggested in the list. Typically, when a recommender system is tuned to increase precision, recall decreases as a result (or vice versa).

$$\text{F-Score} = 2 \cdot (\text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall}) \quad (21)$$

Table 11 shows some outcomes of our recommendation system.

**Table 11.** Outcomes of multi-variant recommendation system.

Evaluating Parameters	TP	TN	FP	FN
Aggregated Polarity	220	356	278	146
Weighted Average Polarity	486	244	196	74
aggregated weighted Average Polarity	640	138	97	125
Final Ranking Score	953	11	16	5

In Table 12 and Figure 10 for recommendations in this domain, a single value is obtained by combining both the precision and recall measures and indicates the overall utility of the recommendation list. One thousand movies were used as exemplary data sets. Evaluations are really important in the recommendation engine building process, which can be used to empirically discover improvements to a recommendation algorithm.

This research used the MovieLens 1K dataset. There are 943 users and 1000 movies; we used the 1000 ratings, votes, likes and views from the users on the films to test the performance of proposed method.

The results of the f-measure differentiated the accuracy of our work from others. If we use the multi-variants system it provided an accuracy of about 98.6%.

Table 12. Weighted average polarity scores.

Evaluating Parameters	Aggregated Polarity	Weighted Average Polarity	Aggregated Weighted Average Polarity	Multi-Variant
Precision	0.3819	0.6658	0.8226	0.9886
Recall	0.4418	0.7126	0.8684	0.9835
F-measure	0.4097	0.6884	0.8449	0.9860

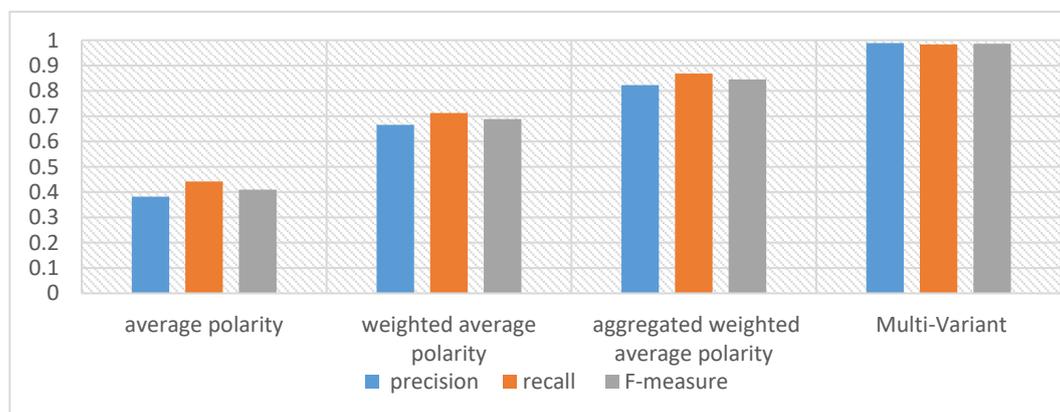


Figure 10. Performance comparison of movie recommendation system.

## 6. Conclusions

This paper presented an intelligent and automated recommender system to provide topic (action, comedy, horror, etc.) based, accurate recommendations of movies to users. The used approach relies on both quantitative and qualitative data for achieving authentic recommendations. The used quantitative data includes ratings, votes, likes, etc., and the quantitative data is the polarity score that is calculated from user reviews using NLP and opinion mining techniques. The developed application was tested on three external data sources such as Metacritics, IMDB, and Fandango and achieved better results in terms of true recommendations as compared to previous approaches. The presented recommender system was developed using a NoSQL environment with Apache Hadoop to filter and integrate movie descriptions from linked data or ontology (linkedmdb). Our approach used a Fuzzy logic approach for movie ranking categorization. A front-end application is designed in Android for the interaction of a user with the movie recommender through web services. When users search for a movie through a mobile app then the server effectively responds to users with a recommended list of movies. Thus, users can take a decision before and secure a watch time for the movie and can conserve other important resources, like money and energy, etc.

## 7. Future Work

Further work is required to enhance the system for both registered and unregistered viewers or users of apps by adding more parameters, such as showbiz industry influence, movie quality, movie trends and a user's profile-based movie recommendation system in a NoSQL distributed environment. Semantic and sentiment computation are required to find the semantic relation between the movies and users as well as the psychological influence of movies.

**Author Contributions:** M.I. designed the algorithm and conducted the experiments. I.S.B. supervised the research work.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A.

One android app provides the following listed features in the user interface for users' usage, by which users can request or query a movie and app will respond with movie list, the list is provided by the server machine by interaction of the app. The following Figure A1. illustrates the specification of the mobile application.



Figure A1. Illustration of multi-variant recommendation.

## Appendix B.

All the experiments were performed to test the performance and accuracy of the proposed approach using Intel i7 @ 3.4 GHz, operating on Linux/Ubuntu 14.04, 64-bit with 8 GB memory. The nltk tool kit is written in Python under the GPL open source license, the Stanford CoreNLP Natural Language Processing Toolkit and libraries [50–52] as well as Apache Hadoop 2.0 are used for the deployment of the NoSQL environment for our movie recommendation system. Table A1 represents the hardware and software specifications.

Table A1. Server machine and Android device specification.

Resources	Specification of Server Machine	Specification of Android Device
Processor	INTEL i7 processor with 3.4 GHz clock rate	Qualcomm Snapdragon 835, 2.45 GHz octa-core Kryo 280 CPU, Adreno 540 GPU
RAM	8 GB/machine	4 GB
NoSQL	Hadoop 2.0, Apache Cassandra.	
Operating System	Linux/Ubuntu 14.04	Android 7.1.1
Storage	1 TB	128 GB (UFS 2.1)
Topology	Connected by gigabit Ethernet cable	802.11ac Wi-Fi with MIMO, Bluetooth 5.0 LE
APP	web-bot	android app

## References

1. Raigoza, J.; Karande, V. A Study and Implementation of a Movie Recommendation System in a Cloud-based Environment. *Int. J. Grid High Perform. Comput.* **2017**, *9*, 25–36. [CrossRef]
2. Christakou, C.; Vrettos, S.; Stafylopatis, A. A hybrid movie recommender system based on neural networks. *Int. J. Artif. Intell. Tools* **2007**, *16*, 771–792. [CrossRef]
3. Said, A.; Kille, B.; de Luca, E.W.; Albayrak, S. Personalizing Tags: A Folksonomy-Like Approach for Recommending Movies. In Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, Chicago, IL, USA, 27 October 2011; pp. 53–56.
4. Zenebea, A.; Norciob, A.F. Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. *Fuzzy Sets Syst.* **2009**, *160*, 76–94. [CrossRef]
5. Singh, D.K.; Gangwar, A.; Sharma, A. Movie Recommendation System. Volume 4. Available online: [www.ijariit.com](http://www.ijariit.com) (accessed on 23 July 2018).
6. Wang, Z.; Yu, X.; Feng, N.; Wang, Z. An improved collaborative movie recommendation system using computational intelligence. *J. Vis. Lang. Comput.* **2014**, *25*, 667–675. [CrossRef]
7. Jain, K.N.; Kumar, V.; Kumar, P.; Choudhury, T. Movie Recommendation System. In *Intelligent Computing and Information and Communication*; Springer: Singapore, 2018; pp. 677–686.

8. Yessenov, K.; Misailovic, S. Sentiment Analysis of Movie Review Comments. *Methodology* **2009**, *17*, 1–7.
9. Bhuiyan, H.; Ara, J.; Bardhan, R.; Islam, R. Retrieving YouTube Video by Sentiment Analysis on User Comment. In Proceedings of the 2017 IEEE International Conference on Signal and Image Processing Applications (IEEE ICSIPA 2017), Kuching, Malaysia, 12–14 September 2017.
10. Singh, V.K.; Piriyani, R.; Uddin, A.; Waila, P. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In Proceedings of the 2013 International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), Kottayam, India, 22–23 March 2013.
11. Alsaqer, A.F.; Sasi, S. Movie Review Summarization and Sentiment Analysis using RapidMiner. In Proceedings of the 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), Thiruvanthapuram, India, 20–22 July 2017.
12. Ouyang, C.; Liu, Y.; Zhang, S.; Yang, X. Features-level Sentiment Analysis of Movie reviews. *Adv. Sci. Technol. Lett.* **2015**, *81*, 110–113.
13. Hsieh, M.Y.; Chou, W.K.; Li, K.C. Building a mobile movie recommendation service by user rating and APP usage with linked data on Hadoop. *Multimed. Tools Appl.* **2017**, *76*, 3383–3401. [[CrossRef](#)]
14. Godhani, G.; Dhamecha, M. *A Study on Movie Recommendation System Using Parallel Map Reduce Technology*; V.V.P. Engineering College: Rajkot, India, 2017; Volume 5.
15. Reza, M.; Sinha, A.; Nag, R.; Mohanty, P. CUDA-enabled Hadoop cluster for Sparse Matrix Vector Multiplication. In Proceedings of the 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015.
16. Castells, P.; Fernández, M.; Vallet, D. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 161–272. [[CrossRef](#)]
17. Wang, J.; Liu, T. Taiwan Improving Sentiment Rating of Movie Review Comments for Recommendation. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW), Taipei, Taiwan, 12–14 June 2017.
18. Wijaya, D.T.; Bressan, S. A Random Walk on the Red Carpet: Rating Movies with user reviews and pagerank. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; ACM: New York, NY, USA, 2008; pp. 951–960.
19. Chang, A.; Liao, J.F.; Chang, P.C.; Teng, C.H.; Chen, M.H. Application of artificial immune systems combines collaborative filtering in movie recommendation system. In Computer Supported Cooperative Work in Design (CSCWD). In Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, Taiwan, 21–23 May 2014; pp. 277–282.
20. Tumasjan, A.; Sprenger, T.O.; Sandner, P.G.; Welpe, I.M. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *ICWSM* **2010**, *10*, 178–185.
21. He, W.; Zha, S.; Li, L. Social media competitive analysis and text mining: A case study in the pizza industry. *Int. J. Inf. Manag.* **2013**, *33*, 464–472. [[CrossRef](#)]
22. Murnane, E.L.; Counts, S. Unraveling abstinence and relapse: Smoking cessation reflected in social media. In Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, Toronto, ON, Canada, 26 April–1 May 2014; ACM: New York, NY, USA, April 2014; pp. 1345–1354.
23. Diakopoulos, N.; Naaman, M.; Kivran-Swaine, F. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In Proceedings of the 2010 IEEE Symposium on Visual Analytics Science and Technology, Salt Lake City, UT, USA, 25–26 October 2010; pp. 115–122.
24. Baldwin, T.; Cook, P.; Lui, M.; MacKinlay, A.; Wang, L. How Noisy Social Media Text, How Different Social Media Sources? In *IJCNLP*; The Association for Computational Linguistics: Stroudsburg, PA, USA, October 2013; pp. 356–364.
25. Corley, C.D.; Cook, D.J.; Mikler, A.R.; Singh, K.P. Text and structural data mining of influenza mentions in web and social media. *Int. J. Environ. Res. Public Health* **2010**, *7*, 596–615. [[CrossRef](#)] [[PubMed](#)]
26. Asur, S.; Huberman, B.A. Predicting the future with social media. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto, ON, Canada, 31 August–3 September 2010; Volume 1, pp. 492–499.
27. Timmaraju, A.; Khanna, V. Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures. Available online: <https://cs224d.stanford.edu/reports/TimmarajuAditya.pdf> (accessed on 14 November 2018).

28. Sarker, A.; Ginn, R.; Nikfarjam, A.; O'Connor, K.; Smith, K.; Jayaraman, S.; Upadhaya, T.; Gonzalez, G. Utilizing social media data for pharmacovigilance: A review. *J. Biomed. Inform.* **2015**, *54*, 202–212. [[CrossRef](#)] [[PubMed](#)]
29. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94), Chapel Hill, NC, USA, 22–26 October 1994; ACM: New York, NY, USA, 1994; pp. 175–186.
30. Shardanand, U.; Maes, P. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*; Katz, I.R., Mack, R., Marks, L., Rosson, M.B., Nielsen, J., Eds.; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 1995; pp. 210–217.
31. Lekakos, G.; Caravelas, P. *A Hybrid Approach for Movie Recommendation*; Springer Science + Business Media: New York, NY, USA, 21 December 2006.
32. Tumsare, P.; Sambare, A.S.; Jain, S.R. Sentiment Analysis Approach for Movie Reviews of Natural Language. *Int. J. Res. Comput. Commun. Technol.* **2014**, *3*, 256–261.
33. Kreutzer, J.; Witte, N. *Opinion Mining Using SentiWordNet Semantic Analysis*; HT 2013/14; Uppsala University: Uppsala, Sweden, 2013.
34. Haddia, E.; Liua, X.; Shib, Y. The Role of Text Pre-processing in Sentiment Analysis. *Procedia Comput. Sci.* **2013**, *17*, 26–32. [[CrossRef](#)]
35. Webster, J.J.; Kit, C. Tokenization as the initial phase in NLP. In Proceedings of the 14th conference on Computational linguistics, Nantes, France, 23–28 August 1992.
36. Vijayarani, S.; Janani, M.R. Text mining: Open source tokenization tools—An analysis. *Adv. Comput. Intell. Int. J.* **2016**, *3*, 37–47.
37. Issac, B.; Jap, W.J. Implementing spam detection using bayessian and porter stemmer keyword stripping approaches. In Proceedings of the TENCON 2009–2009 IEEE Region 10 Conference, Singapore, 23–26 November 2009; pp. 1–5.
38. Porter, M.F. An Algorithm for Suffix Stripping. *J. Program.* **1980**, *14*, 130–137. [[CrossRef](#)]
39. Alphabetical List of Part-Of-Speech Tags Used in the Penn Treebank Project. Available online: [http://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html) (accessed on 11 June 2018).
40. Tf-idf Weighting. Available online: <https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html> (accessed on 7 April 2018).
41. Hakim, A.A.; Erwin, A.; Eng, K.I.; Galinium, M.; Muliady, W. Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In Proceedings of the 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 7–8 October 2014; pp. 1–4.
42. Esuli, A.; Sebastiani, F. Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining. Available online: <http://nmis.isti.cnr.it/sebastiani/Publications/LREC06.pdf> (accessed on 21 September 2018).
43. Baccianella, S.; Esuli, A.; Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *LREC* **2010**, *10*, 2200–2204.
44. Penalver-Martinez, I.; Garcia-Sanchez, F.; Valencia-Garcia, R.; Rodriguez-Garcia, M.A.; Moreno, V.; Fraga, A.; Sanchez-Cervantes, J.L. Feature-based opinion mining through ontologies. *Expert Syst. Appl.* **2014**, *41*, 5995–6008. [[CrossRef](#)]
45. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. [[CrossRef](#)]
46. Pedersen, T.; Patwardhan, S.; Michelizzi, J. *WordNet::Similarity—Measuring the Relatedness of Concepts*; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2004.
47. Bird, S.; Loper, E. *NLTK: The Natural Language Toolkit*; The Association for Computational Linguistics: Stroudsburg, PA, USA, 2004.
48. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations 2014, Baltimore, MD, USA, 22–27 June 2014.
49. Atserias, J.; Casas, B.; Comelles, E.; González, M.; Padró, L.; Padro, M. *FreeLing 1.3: Syntactic and Semantic Services in an Open-Source NLP Library*; TALP Research Center Universitat Politècnica de Catalunya: Barcelona, Spain, 2006.

50. Tiwari, J.; Pawar, M.; Pandey, A. A hadoop based collaborative filtering recommender system accelerated on gpu using opencl. *Int. J. Eng. Sci. Res. Technol.* **2017**, *6*, 195–209, Retrieved 5 September 2017.
51. Thangavel, S.K.; Thampi, N.S.; Johnpaul, C.I. Performance Analysis of Various Recommendation Algorithms Using Apache Hadoop and Mahout. *Int. J. Sci. Eng. Res.* **2013**, *4*, 279–287.
52. Jose, A.V.; Jini, K.M. Personalized Movie Recommender System using Rank Boosting Approach on Hadoop. *IJRST Int. J. Innov. Res. Sci. Technol.* **2015**, *2*, 2349–6010.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).