www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 7 Issue 10 October 2018, Page No. 24366-24371
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i10.07

Personalized Movie Recommendation System Using Twitter Profile Extraction

Saurabh Apte[1], Santoshmurti Daptardar*[2], Pratik Pednekar[3]

[1, 2, 3,] Department of Computer Engineering, Vidyalankar Institute of Technology,
Vidyalankar College Marg, Wadala East, Mumbai-37.

Abstract:
Nowadays, data mining is widely used in almost every field to discover hidden information in large amounts of data that is being continuously generated. For any recommendation system to work accurately, it is important to build profile of the user which is done which is done by extracting personal profiles from various sources such as watched movies and the rating given, purchased items, items liked by users, etc. Movie choice is something very personal and hence it is difficult to accurately recommend movies to user. Current movie recommendation systems use collaborative and content based filtering. This method aims to investigate a different method to build personal profiles using information obtained from Twitter to provide personalized movie recommendation service. For a Twitter user, our method utilizes tweets of the user from which important keywords are extracted using Sentiment Analysis and TF-IDF (Term Frequency - Inverse Document frequency) to build a personal profile. The usefulness of this method is validated by implementing a prototype movie recommendation service and by performing a user study. Using TF-IDF, mapping algorithm and finding users preference of genre, we finally get a score for each movie in the database. We recommend trending movies if the user hasn't tweeted anything before or his tweets aren't relevant for recommendation. This alleviates the cold start problem [1] [4]. The prediction accuracy of movie recommendation is predicted against a small group of users.

Keywords: Personalized movie recommendation, data mining, Twitter, tweets, user profile.

1. Introduction:

With rapid improvement in data processing technology and computer science, the amount of data to be processed has become huge. In today's ever-connected world, recommendation systems have become important than ever before. For companies such as Amazon, Netflix, YouTube and Spotify, recommender systems drive significant engagement and revenue. The reason these companies see increased revenue is because they deliver actual value to their customers. Recommender systems provide a scalable way of personalizing content for users in scenarios with many items. Most of the current movie recommendation systems use collaborative or content based filtering [2]. The main idea of collaborative filtering is that you're given a matrix of preferences by users for items, and these are used to predict missing preferences and recommend items with high predictions. On the other hand, content- based algorithms are given user preferences for items, and recommend similar items based on a domain-specific notion of item content. The task of recommending movies to a user is especially difficult than other recommendations because movie choices are very personal and current recommendation systems which use collaborative and content-based filtering are not as effective. For example let us consider two users A and B. User A likes horror, crime and thriller genres of movies and user B like horror and crime. However, we cannot accurately recommend that the user will like thriller movies. Instead, he may like comedy movies.

Moreover, collaborative filtering approach suffers from the problem of cold start for newly registered users with little history. This paper introduces a hybrid approach combining both collaborative and content-based recommendation for movie recommendation. Twitter is one of the most popular websites today. Millions of tweets are tweeted by users daily. Twitter has a character limit of 240 characters. This is an advantage for recommendation systems as we get a concise data, which accurately expresses users feelings [3]. Hence, data cleaning becomes a lot easier. Given a twitter username, our method extracts tweets of the user, from which important keywords are extracted to build a personal profile, which is then used for selecting new, upcoming movies from our movie dictionary. We preferred R programming language for our project because of useful libraries available and better front-end development tools like Rshiny [5] [6]. We validate the usefulness of this method by implementing a prototype news recommendation service and by performing a user study against a small group of users.

2. Related Work:

Many reports suggest the high relevance of utilizing the information about Twitter usage for movies and news recommendation. Twitter has been previously used widely by advertising companies to gain an insight about interest in their product. It was also used in election campaigns to see popularity of a candidate. Reuters reports that over 80 percent of topics mentioned in tweets have some relationships with news, movies or current happenings. An IEEE paper titled User Profile Extraction from Twitter for Personalized News Recommendation was published by Won-Jo Lee, Kyo- Joong Oh, Chae-Gyun Lim, and Ho-Jin Choi that studied the possibilities of using Twitter tweets and profile of a user to recommend him/her personalized news. This was done using TF-IDF. Their recommendation process consisted of two parts: (1) user profiling, and (2) new ranking. In user profiling phase they built user profiles by extracting their tweets, retweets and hashtags and removing the unnecessary information like emoticons, links etc. In the news-ranking phase, all the noisy information from news titles was removed by data cleaning methods to obtain proper sentences. TF-IDF scores for keywords are computed to form a profile of that article. Lastly cosine similarity is used to check similarity between article profile and user profile to provide news recommendation. Our approach in this paper can be seen as an extension to this paper as we also build user profiles and use TF-IDF.

3. Proposed System:

The project is divided into five parts:

1) Extracting Twitter data and Sentiment Analysis

2) Building genre dictionaries

3) Building movie list from which to recommend movies

4) Finding user's preference of genre

5) Developing the mapping algorithm

3.1. Extracting Twitter data and Sentiment Analysis:

1) Extracting Twitter Data: User has to enter his twitter username. Using API provided by Twitter, timelines of the users are collected. Tweets from a particular Twitter handle are captured using function userTimeLine(). Most of the times, tweets captured may not be proper English words, hence the text is cleaned and common words such as helping verbs, prepositions, etc. are removed from the text.

2) Sentiment Analysis: Sentiment Analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Generally speaking, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation, affective state (that is to say, the emotional state of the author or speaker), or the intended emotional communication (that is to say, the emotional effect intended by the author or interlocutor). For a recommender system, sentiment analysis has been proven to be a valuable technique. We have a set of positive words and negative words dictionary. After the cleaning process of tweets, we

compare each word in these extracted tweets with our dictionaries of positive and negative words. +1 is given for each positive word matched and -1 for a negative sentiment word matched. After adding both these scores, we get a net sentiment score. Only those tweets having a net positive score are kept for further processing using TF-IDF.

3) TF-IDF (Term Frequency-Inverse Document frequency):

TF*IDF is an information retrieval technique that weighs a terms frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term. Put simply, the higher the TF*IDF score (weight), the rarer the term and vice versa. The TF*IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the web, which is referred to as corpus. For a term t in a document d, the weight $W_{t,d}$ of term t in document d is given by:

$W_{t,d} = TF_{t,d} \log (N/DF_t)$

Where:

i) $TF_{t,d}$ is the number of occurrences of t in document d.

ii) DF_t is the number of documents containing the term t.

iii) N is the total number of documents in the corpus.

We used TF-IDF to assign weights to each word.

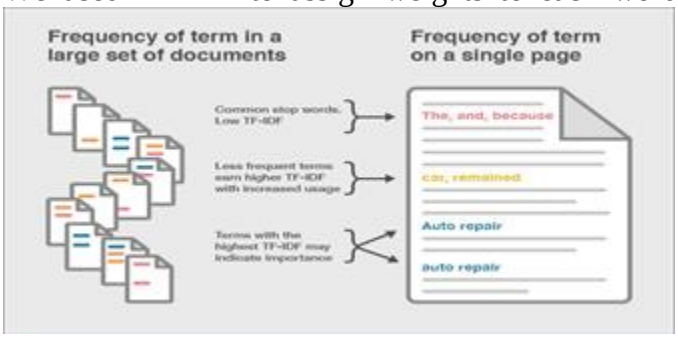

Figure 1: *TF-IDF*

At the end of all these steps, we get a user dictionary.

The entire first step can be explained with the help of following flowchart

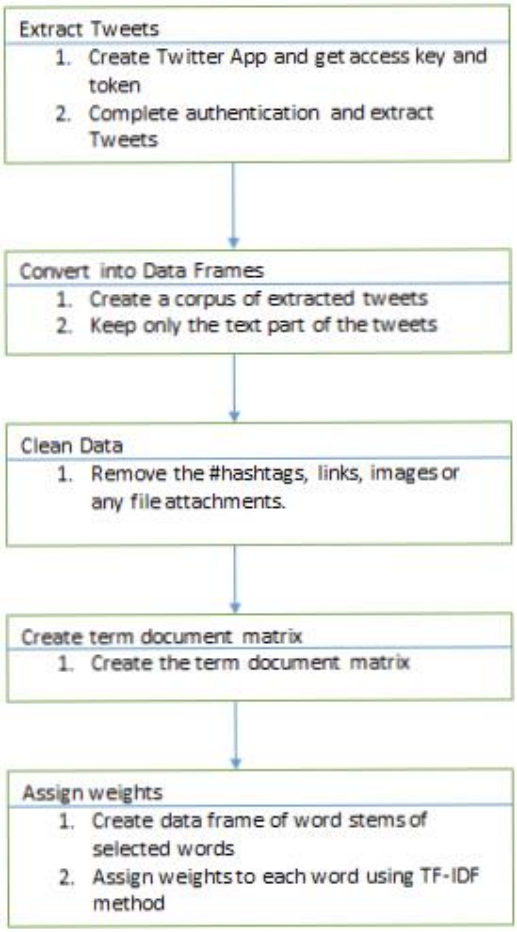

Figure 2: *Steps to assign weights*

3.2 Building Genre Dictionaries:

Movie dictionary refers to a list of words that could represent a particular genre of movies, for example, user may use words like baffling, cryptic, mystical etc. to describe a movie falling into mystery genre. In order to build a movie genre dictionary, we will collect data from 4 different sources. The first two included movie description and movie plots from IMDB and Wikipedia. Third source includes key/tag words for every movie from themoviedatabase.com (tmdb.com). The final step is manual intervention. A list of words is added to every genre, which could represent that genre. Then TF-IDF was performed on this dictionary like it was done on user dictionary to assign relative importance to each word. At the end we get genre dictionary.

Figure 3: *Genres considered*

Action	Animation	Mystery	Comedy	Drama	Family	Horror	Romance	Thriller	Adventu	SciFi
mutant	infant	suspense	singles	play	clan	apprehension	affair	shocker	scene	space
boxer	anime	delphian	student	melodrama	folk	awe	amour	squeaker	trip	technolo
army	disney	delphic	romance	farce	folklore	consternation	attachment	close shave	lifetime	future
hero	magic	abstruse	comedy	broadway	group	disgust	courtship	spine-chiller	chance	futurism
marvel	pixar	ambiguous	college	dramatization	house	dismay	enchantment	cliffhanger	feat	techno
mission	castle	apocryphal.	amnesia	dramaturgy	household	dread	fascination	close call	exploit	fiction
pirate	walt	arcane.	date	showmanship	people	fright	fling	enigma	experienc	war
hero	cartoon	cabalistic	fireman	stagecraft	tribe	hatred	fliratation	riddle	jeopardy	infinity
bond	joy	baffling	lie	theartricals	ancestors	panic	intrigue	secrecy	gladiator	dark
confeden	child	cryptic	speed	show business	ancestry	terror	liaison	subtlety	inception	superher
guns		curious	lol	scene	birth	abhorrence	love	anxiety	revenant	justice le
superhero		magical	rofl	guns	blood	abomination	passion	tension	avatar	marvel
actor		mystical	laughter	tears	brood	alarm	relationship	potboiler	interstell	DC
		mystifying	smile	cry	children	antipathy	sex	page-turner	stellar	annihilati
		obscure	laugh	crying	class	aversion	love story			wonder v
		perplexing	chuckle		descendents	chiller	kiss	uncertainity		valerian
		puzzling	giggle		descent	detestation	flirt	dilemma		martian
		secretive	grin		dynasty	dislike	smooch	doubt		rockets
		strange	howl		extraction	hate	emotion	confusion		guns
		unknown	chortle		forebears	loathing	tenderness			time mac
		weird	guffaw		genealogy	monstrosity	fondnesss			horcrux
		recondite	titter		generations	repugnance	infatuation			harry pot
		covert	roll		in-laws	trepidation	lust			

Figure 4: *Sample Genre Dictionary*

3.3 Building movie dictionary:

A list of around 5000 Hollywood movies and 900 Bollywood movies was developed for the purpose of recommendation. Every movie could fall into 2 to 4 genres out of the 11 major genres selected. A sample list of movies with all the genres is provided below. Binary representation is used to indicate the presence of a movie in a particular genre, either a movie falls into a genre or it doesn't. To include the collaborative filtering in our recommendation, we have also included IMDb (Internet Movie Database) scores of each movie.

Movie_Title	Action	Animation	Mystery	Comedy	Drama	Family	Horror	Romance	Thriller	Adventure	SciFi
Avatar	1	0	0	0	0	0	0	0	0	1	1
Pirates of the Caribbean: At World's End	1	0	0	0	0	0	0	0	0	1	0
Spectre	1	0	0	0	0	0	0	0	1	1	0
The Dark Knight Rises	1	0	0	0	0	0	0	0	1	0	0
Star Wars: Episode VII - The Force Awakens	0	0	0	0	0	0	0	0	0	0	0
John Carter	1	0	0	0	0	0	0	0	0	1	1
Spider-Man 3	1	0	0	0	0	0	0	1	0	1	0
Tangled	0	1	0	1	0	1	0	1	0	1	0
Avengers: Age of Ultron	1	0	0	0	0	0	0	0	0	1	1
Harry Potter and the Half-Blood Prince	0	0	1	0	0	1	0	0	0	1	0

Figure 5: *Movie Dictionary*

3.4 Finding User's preference of genres:

Every word in the user dictionary would be matched with every word in all the 11 genres of genre dictionary. For every word that matched, corresponding scores are multiplied. This will give us score for every genre for that particular user.

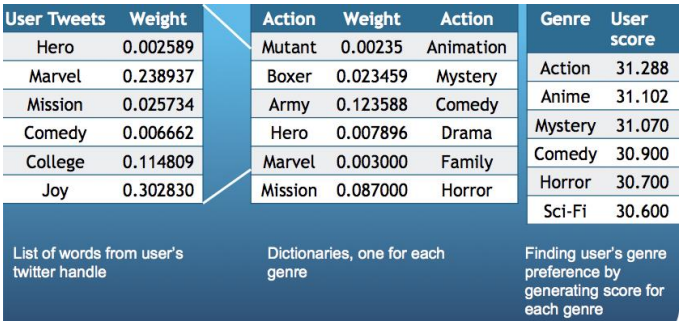

Figure 6: *Finding User's preference*

3.5 Mapping Algorithm:

Following mapping algorithm was used to make the final recommendation:

1. Take dot product of Hollywood and Bollywood movie matrix (5000 x 11) and (900 x 11) with the score vector (11 x 1).

2. Get movie score vector (5000 x 1) and (900 x 1) having score for every movie.[SEP]3. For those movies having same movie score, sort them according to their IMDb score.

4. Recommend movie at the top of the list.

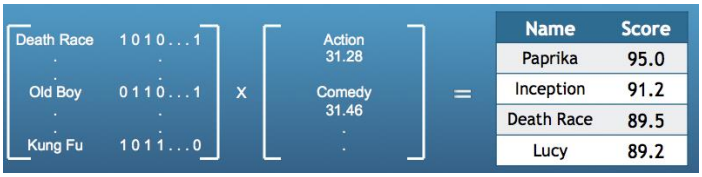

Figure 7: *Mapping Algorithm*

4. Experiment:

Using the Twitter API provided in the Twitter website, we have collected the timelines of eight users who agreed the user study and gave us permission for collecting their data. In order to check effectiveness of our method, we asked the users directly about their preference of choosing which movies to watch. We built the following website for

movie recommendation using R programming language [7], [8], [9] and R Shiny [10].

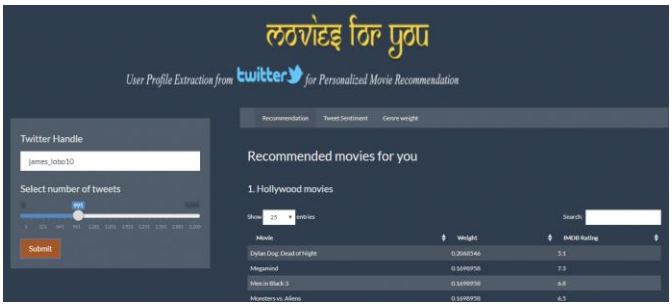

Figure 8: *User Interface of Recommendation system*

In the website, the user has to enter his twitter handle and he can choose the number of tweets, which he wants the system to consider. Upto 3200 tweets can be considered for movie recommendation.

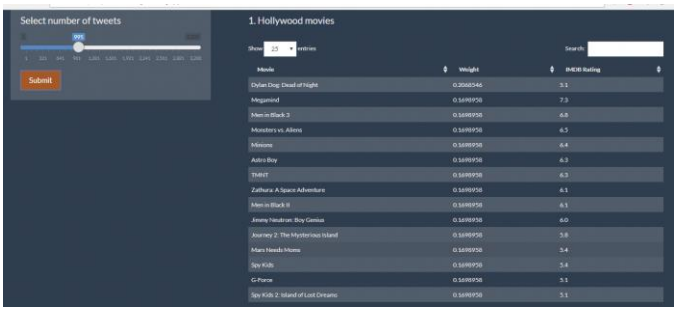

Figure 9: *Output of Recommendation system*

Movies are arranged in descending order of their weights. We have further included IMDb ratings to include collaborative filtering approach and make our system a hybrid one.

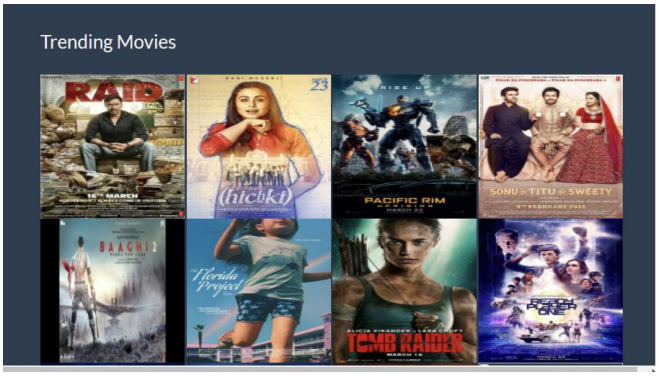

Figure 10: *Top trending movies*

To overcome the problem of cold start we have included trending movies section. This is useful is user does not have a twitter account or hasn't tweeted anything relevant for recommendation.

5. Conclusion:

Here we have proposed a movie recommendation system using Twitter profile extraction based on sentiment analysis and TF-IDF method. In this approach, first the sentiment of the tweet is analyzed, from the tweets; important keywords are extracted for which weight is assigned to each word using TF-IDF. Similarly, weights are also assigned to the keywords of movie genre dictionary. Finally, using mapping algorithm both the weights are multiplied to get a score, which is used to recommend movies. For movies getting same score, they are arranged according to their IMDb scores. The method used in this project focuses mainly on the tweets tweeted by the user and hence the problem of cold-start can be dealt only by recommending the most popular or top trending movies. The usefulness of this method is validated by a user study experimented over a prototype movie recommendation service and the accuracy is satisfactory. This project can serve a perfect tool for online marketing based on user's social media profile.

Acknowledgement:

The authors would like to thank Prof. Devendra Pandit and Prof. Pankaj Vanwari for their immense guidance in making our idea into reality.

References:

[1] Ossama H. Embarak, *A method for solving the cold start problem in recommendation systems*, in Proc. Innovations in Information Technology 2011 pp.2-6

[2] O Phelan, K McCarthy, and B Smyth, *Using twitter to recommend real-time topical news*," in Proc. ACM Recsys09, pp. 385-388, 2009.

[3] F Abel, Q Gao, GJ Houben, K Tao, *Analyzing user modeling on twitter for personalized news recommendations*, User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, pp. 1-12, 2011.

[4] Paul Resnick, Hal Varian, *Recommender System*,," Communication of the ACM, Vol. 40,

No.3, 1997.

[5] AI Schein, A Popescul, LH Ungar, and DM Pennock, *Methods and metrics for cold-start recommendations*, in Proc. ACM SIGIR02, pp. 253-260, 2002.

[6] J Wang, Z Li, J Yao, Z Sun, M Li, and W Ma, *Adaptive user profile model and collaborative filtering for personalized news*," Frontiers of WWW Research and Development-APWeb06. Springer Berlin Heidel- berg, pp. 474-485, 2006.

[7] https://tutorialspoint.com/r

[8] https://udemy.com/r-basics

[9] https://dev.twitter.com

[10] https://rshiny.com

Author Profile

Saurabh Apte is currently in final year of program B.E. in Computer Engineering from Vidyalankar Institute of Technology.

Santoshmurti Daptardar is currently in final year of program B.E. in Computer Engineering from Vidyalankar Institute of Technology.

Pratik Pednekar is currently in final year of program B.E. in Computer Engineering from Vidyalankar Institute of Technology.