News Recommendation using Sentiment Analysis

Kavya Nirmal Shah
EECS Department
(Computer Science)
Syracuse University
Syracuse
kshah28@syr.edu

Prapti Kamlesh Oza
EECS Department
(Computer Science)
Syracuse University
Syracuse
poza02@syr.edu

Spandan Manilal Patel
EECS Department
(Computer Science)
Syracuse University
Syracuse
spatel20@syr.edu

Abstract **— Different social networking platforms like Twitter are crucial for sharing knowledge, information, concepts and opinions. Finding the sentiment expressed in a text, such as a networking post, product review, or news item, is the aim of sentiment analysis over Twitter. We can get a better understanding of the user's interests by getting the sentiment of the user's tweets, so we can integrate sentiment analysis in recommendation systems. By analysing the user's preferences, the system is able to provide more personalized and accurate recommendations, effectively addressing the issue of information overload. With the growing number of individuals consuming news content online, the availability of millions of items from various sources can be overwhelming. To alleviate this, news recommender systems have been developed to suggest relevant news articles based on the user's interests, providing a curated selection of pertinent content. We indicate in our paper, a personalized news recommender system based on sentiment analysis using a hybrid deep learning method and lexicon-based model VADER. First, we collected all the data and tweets of the users from the twitter API. Next, we implemented web scraping to build a corpus of the news articles from different news channel URLs later used to be recommended to a user based on their interest. Later we pre-processed all the data using the NLP libraries and generated a vector representation for each tweet of a Twitter handler using the two lexicon-based sentiment analysis models and trained it through an ANN-LSTM hybrid deep learning method and evaluated the performance of our classification model. The model analyses the sentiment expressed in user tweets, and provides more personalized recommendations. We computed the accuracy results of the recommendation system model and found significant results using hybrid deep learning methods. We delivered another recommendation system without sentiment analysis of the user tweets. After successful retrieval of the recommended news article based on the two models mentioned individually, we calculate the performance of our model on the basis of their Jaccard similarity score. Our study showed improvement in the system's performance, demonstrating its potential to enhance the user experience and provide personalized recommendation.**

Keywords— Sentiment Analysis, Recommendation, Deep Learning, ANN-LSTM, NLP, Web Scraping, TFIDF, NLTK

I. INTRODUCTION

With the advancement of Internet and technology, various types of social-media platforms, forums, blogs like Instagram, Reddit, Twitter and more have taken their course into major public usage and activity. User engagement on such platforms like posting, commenting, liking and more is now providing vast amounts of information and content on user profiles. People are more likely to share their views, opinions, daily activities, likes, dislikes and more on such platforms causing high user engagement and further getting comments and reviews on the same. Such kinds of information about each individual user can be useful and taken into consideration for business purposes and decision making. Twitter is one such platform where users tend to engage in a lot of discussions and thus has a huge data pool. Twitter has over 100 million users, over 500 million tweets are posted every day and due to the tweet body limitation, tweets have to be very short and concise giving the exact message being conveyed very clearly[3].Users tweet and retweet on other tweets sharing knowledge and views with different kinds of sentiments. This sentimentally rich data of users can be used to get the user sentiments and make recommendations accordingly to make recommendation narrow and concise based on the sentiment of users more likely, positive. Recommendation systems are used by retail businesses, e-commerce websites, entertainment applications and more to recommend their products or information using different filtering methods like Content-based filtering method, Collaborative filtering and Hybrid filtering[6][7]. These filtering techniques are used to make recommendations in different ways to users. The main aim is to provide relevant recommendations to users based on the information retrieved from user's data in an accurate way and catering to the ever changing user interests. Sentiment Analysis is an analysis of user information like tweets, posts, reviews, ratings etc. Sentiment Analysis gives the sentiments on user information classified as positive, negative or neutral based on the user's topic of interest. This can be used to get the understanding of what the user feels about different products, news, movies, songs or any other kinds of information available. For example, tweets of a user can be used to get the user's opinion and compute sentiment scores and classify based on the scores generated. This can be done either using Lexicon-based approaches or Machine Learning based approaches[3]. Recommendation systems based on the user's past data pose a problem of recommending even the things that the user might not be interested in. As an instance, a user's tweets might also have content that he/she dislikes and has a negative outlook on that. Such content should be avoided while recommending new information to the user. This problem nowadays is being alleviated by integrating Recommendation Systems

with Sentiment Analysis to classify the user's tweets based on the sentiment labels and recommending only the information that he/she has positively tweeted about. Such a solution is used to increase the accuracy and relevancy of recommendations made to users and this can be done using various filtering techniques. This solution is discussed in different ways in the depending on the papers that we reviewed throughout our research. Finally, we are comparing the traditional content-based Recommendation System using Twitter's user's tweets data with the Recommendation System using Sentiment Analysis by hybrid learning models ANN_LST and checking the similarities using Jaccard of both concluding the latter is better in terms of similarity with user's interests.

II. Literature Review

In this paper[7], they have proposed an architecture where after preprocessing the data, classification is done using BERT which is an NLP model and text data is transformed to word embedding. It is used to map each word to have a similar representation of similar semantics. After preprocessing of data and creating feature vectors, training models for sentiment analysis come into picture using the proposed hybrid deep learning models in different hierarchies. The data is then labeled with five classes of sentiments ranging from very negative to very positive giving the output for the sentiment analysis. Furthermore, the recommendation method that is used in this paper is user based collaborative filtering to get data from all users with similar interests. Different CF methods were checked to get good accuracy and this was integrated with the sentiment analysis done previously to get the output and recommendations of Amazon Movie Reviews(dataset taken) based on that output. The main aim was to get the comparison between widely used recommendation system and a recommendation system with sentiment analysis. They conclude with the latter being more accurate. In this paper[8], they have proposed a system that uses hybrid deep learning models like CNN and LSTM on a vector of words in customer product reviews dataset. Prediction for user is done by integrating the item ratings of neighbors. They have created an architecture similar to the previous paper[7], but instead of the BERT model, they have done vectorization of words using Continuous Bag Of Words (CBOW) used to get the next word/target word based on the surrounding words. The combination of CNN and LSTM is used for extracting features and information from the past. So, the features obtained from the CNN model are taken as input into the LSTM to get a text vector. The output from the LSTM is then used for recommendation using sentiments. For recommending items to users, a matrix factorization model is used. Finally they conclude that the architecture shows the highest result for sentiment as well as recommendation.In this paper[6], movies are being recommended using sentiment analysis based on reviews. This is done using machine learning algorithms. First data preprocessing is done on the reviews dataset using NLTK and removing stopwords, tokenization and TfidfVectorizer. It gives significant findings along with in-depth analyses of both parts of the system, i.e. sentiment analysis and recommendation system. The Cosine Similarity algorithm is used to suggest films that are comparable to the user's selected film based on details from the dataset of movies.The study also includes a significant amount of sentiment analysis. The sentiment analysis part of the system uses two algorithms namely NB and SVC and these models are fitted by training and testing models. They have used a multinomial NB model to get the tags of the texts and then calculate probability. Output is the probability that is the maximum among all. Its goal is to categorize reviews based on the positive, negative and neutral sentiments. Finally the study compares all the accuracy and precision scores of NB, SVM and other existing models, finding the highest frequency of SVM followed by NB.

III. Methods

The process of developing recommendations based on sentiment analysis includes gathering user data based on their likes, dislikes, previous interactions, and behaviour with the system. The acquired data is beautified, and features are extracted. Sentiment Analysis is then conducted on the pre-processed data to assess the users' feelings based on their tweets and other associated content. After the sentiment analysis, the acquired data is clustered to group comparable data together. Finally, suggestions are created based on attitudes and user preferences. After that, the correctness is examined. However, in our research, we also demonstrated the usage of online scraping of the news articles and subsequently doing user suggestions.

A. Data Collection:

The data collection process is very important for any system. Depending on the system and type of the data requirements, various methods for data collection can be utilised. Data can be obtained utilising Social Media APIs, Web Scraping, User Profiling, User History, and other methods.

Tweepy for instance, is a social media API which has been used in our project and is also a widely used python library that enables users to extract the real time user data by fetching the user's tweets, likes, dislikes, retweets and much more.

On the other hand, Web Scraping involves the extraction of data from websites using the automated tools or scripts. This technique uses various resources to collect the data along with the websites such as online marketplaces and social media platforms. Overall, the choice of data collecting method and type of data collection depends on the individual preferences and the available resources.

B. Data Preprocessing:

Data preprocessing is another important step in data analysis and machine learning.[3] It requires the syntactically corrected tweets and helps in making the data easily readable by the machine to reduce ambiguity in feature extraction.

It involves the cleaning, transforming and preparing the raw data for analysis. We use some modules like Pandas, NLTK etc to clean and pre-process the data.

1. Pandas: Panda is an open source library used to open data files and to apply certain operations on the data. Pandas helps to clean the data which is in the tabular form and the data table is called Data Frame in pandas. The library can be imported by using ***pip install pandas*** [15]
2. NLTK: Natural Language processing Toolkit also known as Nltk is widely used for Natural Language processing library for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers and much more. [14]. It is imported by using ***import nltk***
3. Spacy: Spacy is also an open source library for Natural Language processing which helps in achieving the tasks such as tokenization, part-of-speech tagging, dependency parsing and much more. It is imported in the project by using ***import spacy.***

There are some few steps involved in preprocessing the data:

a. Data Cleaning:

Data cleaning is a necessary step in data preprocessing and data analysis that entails discovering and repairing or deleting errors, inconsistencies, and inaccuracies in raw data to assure the data's quality and trustworthiness.

- Missing values are a common issue in real-world datasets, causing difficulties with data analysis and modelling. Missing values can be addressed in data cleaning by imputing them with appropriate values, such as the data's mean or median, or by employing more advanced approaches, such as regression imputation or multiple imputation.
- Correcting Incorrect Data: Data mistakes can arise owing to a variety of circumstances, including measurement problems, data entry difficulties, and coding flaws. Data mistakes can be fixed by finding and replacing them with the right values, or by eliminating them entirely.
- Duplicates are another common issue in datasets that can cause issues with data processing and modelling. Data duplications can be deleted by recognizing and eliminating them using established criteria such as exact match or partial match.
- Data inconsistencies can emerge as a result of changes in data formats, measuring units, or data sources. Inconsistencies in data can be rectified by standardising the data to a common format or unit of measurement, or by reconciling data from many sources.
- Validation of data: Data validation is an important part of data cleaning that entails ensuring that the data is accurate and comprehensive. This can be accomplished by comparing the data to external sources, conducting statistical analyses, or employing data visualisation tools.
- Documenting data cleaning procedures: It is important to document the data cleaning procedures and decisions made during the data cleaning process to ensure transparency and reproducibility of the data analysis.

b. Data Transformation:

- Converting Upper into Lower Case: We use case sensitive analysis in this case because a text might contain the same words but because of the sentence case, we take them as two different words. Hence, in this case when we convert the texts into lower case the redundancy in the texts will reduce.[3] We can achieve lower case by using the **.lower()** method of python
- Removing links: links are not as important unless the system has specific requirements. Hence, most of the time, we tend to remove the usernames and links as they take unnecessary space in the data.
- Removing retweets: If we are using the twitter api to fetch the user data, then we tend to remove the retweets done by the user as they are not the actual tweets of the user.
- Removing usernames: Just like links, usernames are also not as important unless the system has specific requirements.
- Removing Punctuations: Punctuation is an important aspect of written language; however, it is not beneficial for NLP tasks. Because punctuation frequently contains little helpful information in the data. As a result, they are frequently eliminated during the preprocessing stage.

c. Stemming:

Stemming [13] is the process of removing the term's basis or root word. To put it simply, stemming assists in reducing the base word to its stem term. It facilitates quicker lookups and normalises the language for easier comprehension. For example, if you stutter "danc," you can get "dance," "danced," etc. As a result, the stemming algorithm reduces all of the terms.

There are various algorithms that help in performing the stemming.

1. Porter Stemmer algorithm

It is a rule-based algorithm that removes suffixes from words and transforms them into their base or root form using a collection of heuristics and rules. The Porter Stemmer class from NLTK may be used to implement the Porter Stemmer Algorithm.[13]. With the use of this class, we are able to convert a text word into its resultant stem, which results in a term that is both shorter and has the same fundamental meaning.[13]

2. Lancaster Stemming Algorithm: [16]

Its iterative strategy may result in over-stemming, resulting in linguistically inappropriate roots. It is

inefficient compared to a porter or snowball stemmer.

3. Regular Expression Stemming Algorithm: [13]: We may construct our stemmer using the Regexp stemming technique. The NLTK has a RegexpStemmer class that may be used to implement Regular Expression Stemmer algorithms. It accepts a single expression as input and eliminates any suffixes and prefixes that match the phrase.

4. Snowball Stemming Algorithm: It's fantastic stemming algorithm. Snowball Stemmer algorithms may be implemented using the NLTK Snowball Stemmer class.[13] To take advantage of this module, we must first establish a subforum with the name we intend to use, and then, after it is finished, we must use the stem () method.

d. Lemmatization:

Lemmatization is a process of grouping all of a word's inflected forms together so that they may be analysed as a single unit.

Lemmatization is the process of reducing a word to its basic or dictionary form, known as the "lemma." The lemma of the words "walking," "walked," and "walks" is, for example, "walk."

Text analysis, information retrieval, and machine learning all employ this method. It is possible to increase the accuracy and efficiency of language processing systems by reducing words to their simplest form.

e. Removing Stop words:

Stopwords are basically the common words like "is", "the", and so forth which does not add to much meaning to the text.

Removing these words helps in minimising the size of the corpus and increase the effectiveness of analysing the texts

f. Tokenization

Tokenization is a fundamental natural language processing (NLP) approach that involves breaking down a text corpus into individual components, or tokens. Depending on the job or application, these tokens might be words, phrases, sentences, or even bigger units. There are several types of tokenization such as Word Tokenization, Character tokenization, and Sub Word Tokenization.

Tokenization is a strong approach used in many NLP applications such as sentiment analysis, machine translation, and topic modelling. Researchers can obtain a greater knowledge of language use and trends by breaking text down into its basic elements, allowing them to make more accurate and insightful observations and conclusions.

C. Feature Extraction

A data analysis and machine learning technique for discovering and extracting the most relevant and informative properties from a dataset is known as feature extraction. Natural language processing (NLP) feature extraction is the process of converting a text corpus into a collection of numerical characteristics that may be used to train machine learning models.

There are several approaches for extracting features accessible today. Inverse term frequency Document frequency is an effective strategy. The TF-IDF statistic is a numerical statistic that represents the value of a word for the entire text (here, tweet).

Vectorizers in Scikit-learn convert input texts into feature vectors. We may use the library function TfidfVectorizer () to specify parameters for the types of features we wish to maintain by declaring the minimum.

TF-IDF:

TF-IDF limits the words that are too common to be of any significance while allocating each word its relevance in proportion to how frequently it appears in the corpus. The formula is tfidf = tf (t,d) * idf (t).

D. Sentiment Analysis

Sentiment Analysis is used to detect the emotional tone or mood of a words, phrases used in a text and are then classified into positive, negative or neutral. For doing sentiment analysis, the text is first pre-processed to get rid of any extra information, like stop words and punctuation using different libraries like NLTK, Spacy. The text is then tokenized, or divided into separate words or phrases. After preprocessing of data, a model is applied to the extracted features and sentiments are generated. The overall sentiment of the text is calculated using a variety of techniques after the sentiment of each token has been established. There are different types of sentiment analysis approaches:

1. Lexicon - based approach: Lexicon based approach of sentiment analysis involves preprocessing of data like tokenization, Stopwords removals etc and this pre-processed data can be labelled into sentiments using Text blob(NLTK library) or Valence Aware Dictionary and Sentiment Reader(VADER-NLTK library).[17]

2. Machine Learning Approaches - There are different Machine Learning models used in sentiment analysis like Naive Bayes, KNN, CNN, ANN, Logistic Regression, SVM and more.

- Naive Bayes: Naive Bayes algorithm in machine learning approach uses conditional probability of words. The classifier calculates the frequency of words in the text such that all the features are independent of other features. It is also used to classify text into positive, negative or neutral.
- LSTM: Long Short-Term Memory networks are a RNN type that are implemented using Keras library. LSTM are used for avoiding the long-term dependency

problems encountered by RNNs There are three gates called Forget gate, Input Gate and Output gate. In the forget gate, the model chooses the information to remember or decides to forget if irrelevant. In the Input gate, the model gets input and learns new information. In the Output gate, the model updates information. LSTM has a hidden state H(t-1) and H(t) – Short term memory, and it has cell state as C(t-1) and C(t) – Long term memory. [18]

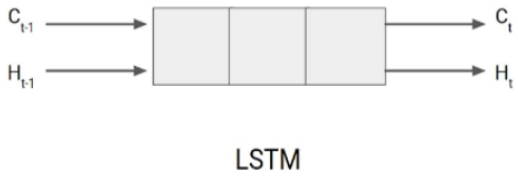
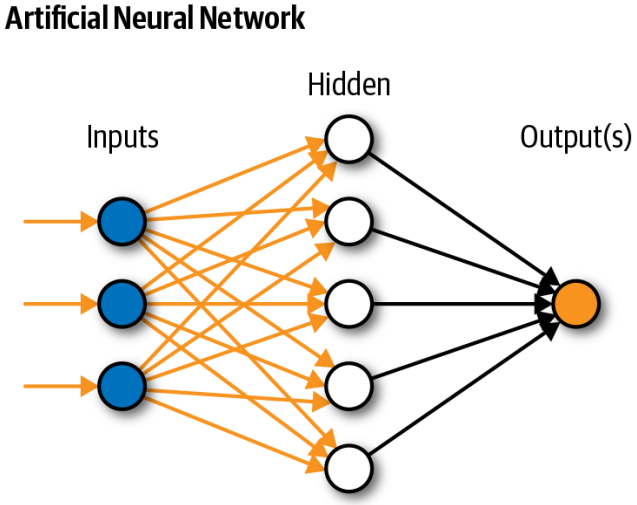


Fig. 3.1 Performed Pre-processing on the fetched data [18]

- ANN: it is a machine learning model used in NLP. It consists of interconnected nodes/neurons which are used to learn and recognize patterns to make predictions of the data that is inputted. A node/neuron takes input, gives weights to them and then it passes that to an activation function. This produces an output and this is repeated till final prediction. ANN takes a sentence of text and predicts if it is positive, negative or neutral. ANN takes features extracted after preprocessing as inputs and then passes them via a series of hidden layers in the network. ANN models can be trained on a dataset with sentiments to predict the sentiments of new data. [20]



- Hybrid Approach: Hybrid approach is when machine learning models are combined to form a system which can then be used for sentiment analysis for better performance and accuracy. It can be used to calculate the similarity measure between the set of recommendations and the set of users interests data. One such combination is:
- ANN - LSTM: ANN takes features extracted after preprocessing as inputs and then passes them via a series of hidden layers in the network. ANN models can be trained on a dataset with sentiments to predict the sentiments of new data. LSTM are used for avoiding the long-term dependency problems encountered by RNNs. When integrating ANN and LSTM together, ANN extracts the basic features from the data and LSTM captures long-term dependencies. So, after data preprocessing data, ANN extracts features using different techniques like bag-of-words and the LSTM model is then trained on this pre-processed data and it captures the long-term dependencies as input and it learns to remember the relevant information/ important features for sentiments. The gates and cells in LSTM model are used to update the features and relations between the features and sentiment labels on the text given as input. The output is then used to make predictions for sentiments.

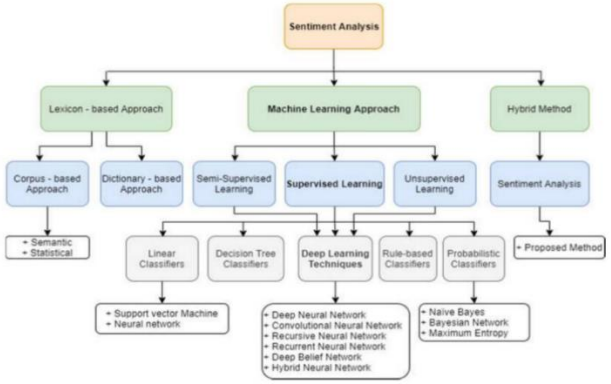


Fig. 3.2 Sentiment Analysis Tree

E. Web Scrapping

[19] Web scraping, commonly referred to as web harvesting or web data extraction, is a technique for gathering data from websites. It includes retrieving and extracting information from online pages using software that connects to the internet using the Hypertext Transfer Protocol or a web browser. The content of a page may be examined, searched for, reorganized, and copied into a spreadsheet or database to extract the needed data. Typically, web scraping is taking certain data from a website for further processing.

F. Cosine Similarity

The cosine similarity measures the similarity between two vectors generated after preprocessing the text and vectorization, are to each other. It calculates the cosine angle between two vectors and determine the similarity on the basis of that angle. The cosine similarity computes a nonnegative value ranging from 0 to 1. 0 pointing towards least similarity between two sets and 1 point towards highest similarity between two sets. In text analysis, it is frequently used to measure document similarity. Cosine similarity is computed by dot product of the vectors divided by both vector absolute length product.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Fig. 3.3 Cosine Similarity

G. Recommendation System

Recommendation systems are used by retail businesses, e-commerce websites, entertainment applications and more to recommend their products or information using different filtering methods like Content-based filtering method, Collaborative filtering and Hybrid filtering [6][7]. These filtering techniques are used to make recommendations in different ways to users. The main aim is to provide relevant recommendations to users based on the information retrieved from user's data in an accurate way and catering to the ever-changing user interests. Recommendation systems use various filtering methods:

1. Content Based: Content based filtering technique is used to recommend the users content/items based on his/her recent interests or activities/past interest and behaviours. The content-based method gets the features of the items and finds the similar items to recommend to that user. So, if a user liked an item previously, it is likely that he/she will like similar items in the future as well. The user data like interests, tweets, purchased products etc is represented in vector representation and the item data is also similarly represented in vector and the cosine similarity between them gives how similar is the item to the user's interests. [filtering-image-link]
2. Collaborative Filtering: Collaborative filtering technique is used to recommend content/items based on the between the user's interests and the other user's interests. So, if any two users have same interests then those users will be recommended the same items. There are two types: a) User-user based: In this, the vector representation of user data will have all the items of his/her interests and the ratings/rankings. It recommends items to user based on similar user's items and recommends items that those users have liked. 2) Item based collaborative filtering: In this, items are compared to the items that the user has liked in the past and similarity values are computed to recommend the items with high similarity values. [filtering-image-link]
3. Hybrid filtering: Hybrid filtering is a combination of content based and collaborative filtering techniques to recommend items to the users. It integrates the strengths of each filtering techniques thus it is the best in performance. This involves using collaborative filtering to recommend items other users have liked and also content based to recommend the items that the user have liked in the past.

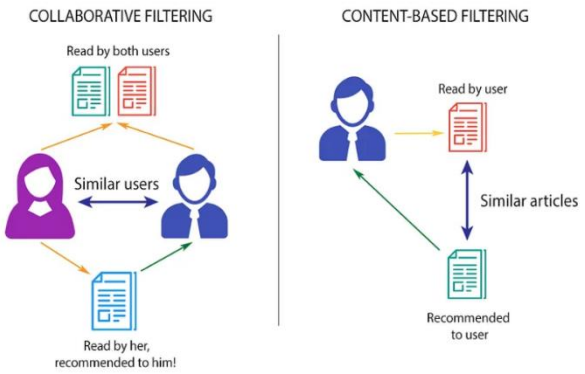


Fig. 3.4 Types of Recommendation System [21]

H. Jaccard Similarity

One can determine how similar two asymmetric binary variables are using the Jaccard Similarity. It relies on the hunch that two collections of items that have a lot of overlap might have a connection to one another. Jaccard similarity is computed as size of intersection of two sets divided by size of union of the two sets of interest. The jaccard similarity ranges from 0.0 to 1.0. The higher the jaccard similarity of the two sets, the higher the implication that they are connected to each other. Jaccard similarity is commonly used in data science applications such as text mining, E-commerce and recommendation systems.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Fig. 3.5 Jaccard Similarity

IV. RESULTS

We conducted experiments in two different environments, both without and with sentiment analysis for news article recommendation. While with the latter, the outcome of performing sentiment analysis on the news tweets is included in the recommendation process, whereas the former bases recommendations on recommender systems approaches without sentiment. We are using a hybrid deep learning model approach involving ANN-LSTM methods for training and testing of sentiment analysis which is used in the recommendation system in the latter system.

1. Fetching data using Twitter API:

The most important is to have twitter authentication customer and token keys and secret keys. The Twitter API tokens and keys are set up. To authenticate a user's request and ensure only authorized users can access the API, specific keys and tokens are needed. Later the 'OAuthHandler' class is created which is used to authenticate the user API. Then for the authenticated user we set the access token and the access token secret. Finally, we generate the tweepy library's API class .

The user data is fetched using the generated API class using user_timeline method. We than iterate over the collected tweets and retweets to fetch the ‘Users’, ‘FollowersCount’ and ‘TotalTweets’ of the users associated with the news channels ‘FoxNews’, ‘NYTimes’ and ‘NBCNews’.
We retrieve the top tweets and retweets of the users using a ‘fetchUserTweets’ definition.

2. Preprocessing:

In our code for data preprocessing we have used regular expressions, spacy, tokenization, Stopwords, lemmatization. After retrieving the tweets from twitter, the preprocessing of them is very important. Building a recommendation system requires several steps, including preprocessing, which helps to clean, transform, and get the data ready for analysis.

- Cleaning: The tweets mostly contain the garbage values like ‘https:\\’, ‘RT’ (retweet tag), ‘@’ (mentions). Thus, we clean this using Regular expressions.
- Tokenization: Based on a set of predetermined rules, such as those governing whitespace, punctuation, or linguistic features, the text corpus is divided into discrete words or tokens.
- Stopwords: Stop words can be eliminated from the tokenized text since they are common terms like "the," "and," or "a" that do not provide any useful information.
- Lemmatization and stemming: Methods for getting words back to their original forms. As a result, the vocabulary is shortened and the recommendations' correctness is increased.
- Finally, we generate pre-processed data for each user and store it in the data frame as ‘tweets as shown in (Fig.1)

	Users	FollowersCount	TotalTweets	UserTweets	ptweets
0	Charliac	1.357895	26702	['AVOID A DEBT CRISIS': The House passes Speak...	avoid debt crisis house pass speaker mccarthy ...
1	WayneBorkes	1.602446	36317	[Just in case you were wondering why Tucker wa...	case wonder tucker fire allow ask question thi...
2	FrittsJensen	204.000000	24769	[A whistleblower who worked at Biden's Health ...	work biden health human service hhs say federa...
3	RockNRollVA	1.151659	151202	[@washingtonpost Lock her lying self up. Enoug...	lock lie self special treatment guilty entrepr...
4	kenkircher1	1.994638	1245723	[🚨 A few asked for it in my DMs. Here is what ...	ask lift debt ceiling mean gop legislation pas...
...	...	...	...	...	...
144	TheVEchols	1.297002	23200	[After 10 days of brutal warfare that has kill...	day brutal warfare kill hundred tenuous ceasef...
145	RyanCla20084299	1.100000	3425	[Scott is right.\nThe GOP Establishment don't ...	scott right gop establishment not care win win...
146	coca536	3.417355	165073	[Why did Fox people tell all those lies on TV?...	fox people tell lie amen she curse criminal hu...
147	AliAdair22	1.035104	718131	[must...fit..., https://t.co/LDjBWE0ZCJ, Arka...	arkansas man sentence prison capitol riot case...
148	aintscarylarry	3.392692	19759	[🚨 Fascism continues unabated in Republican-le...	fascism continue unabated legislature state re...

149 rows × 5 columns

Fig. 4.1 Performed Pre-processing on the fetched data

3. Clustering:

In our system we have used TF-IDF vectorizer, cosine similarity and K-Means clustering to make clusters of the users based on the topic of interest. Utilizing TfidfVectorizer, text input can be pre-processed and changed into a numerical format.
For each phrase in the corpus, the phrase Frequency-Inverse Document Frequency (TF-IDF) score is calculated. Later we fit transform the tweets and generate a tf-idf matrix.
We create a features data frame from the above tf-idf matrix.
Using K-Means clustering we are separating the users and the tweets in 3 different groups which share the same interest of topics and sentiment by calculating the sentiment polarity score of the user tweets. This is useful in Collaborative filtering.

```
Top words per cluster:
Cluster 0:
 not
 amp
 biden
 like
 people
 good
 school
 point
 man
 know
Cluster 1:
 trump
 say
 year
 break
 time
 president
 not
 new
 carroll
 donald
Cluster 2:
 house
 debt
 bill
 vote
 republicans
 pass
 cut
 montana
 break
 lawmaker
```

Fig. 4.2 Clusters found after performing clustering

4. Web scraping and corpus building:

We fetch the articles from different newspaper URLs such as ‘BBC’, ‘CNN’ and ‘Washington Post’. We make a request to the website and get the HTML content. Later using Beautiful Soup, we parse the HTML content. Find and filter all the links and append into a data frame which contains the link of the article and the text content of the articles.
We reprocess the data by doing cleaning, removing Stopwords, tokenization and lemmatization. This dataframe works as a corpus for our further analysis and recommendation system.

	link	text
0	https://www.bbc.com/	Pep Guardiola says Manchester City are deliver...
1	https://www.bbc.com/news	Trump rape accuser E Jean Carroll takes the st...
2	https://www.bbc.com/sport	The Ironman unbroken by a bomb\n\nLosing blood...
3	https://www.bbc.com/reel	Why your life is probably a simulation
4	https://www.bbc.com/worklife	How we think
...	...	...
485	https://www.washingtonpost.com/privacy-policy/	Site Information Privacy Policy\n\nShare\n\nPu...
486	https://www.washingtonpost.com/cookie-policy/	Ask the Post Cookie Notice\n\nShare\n\nUpdated...
487	https://www.washingtonpost.com/discussions/202...	Site Information RSS Terms of Service\n\nShare...
488	https://www.washingtonpost.com/information/202...	Site Information Ad choices\n\nShare\n\nPublis...
489	https://www.washingtonpost.com	Speaker Kevin McCarthy (R-Calif.) aims to forc...

300 rows × 2 columns

Fig. 4.3 Texts obtained from the News Articles after Web Scrapping

5. Sentiment Analysis:

We fetch tweets of the top contributing user of news tweets and fetch more than 3000 tweets using pagination. We filter the words tweets which have only emojis as their text, since we need clean data for sentiment analysis. We reprocess the tweet text and store it in a dataframe. Then we use Sentiment Intensity Analyzer to calculate the sentiment polarity of the tweet’s texts. We label the tweets into negative or positive based on the polarity score of the VADER analyser. The hybrid deep learning ANN-LSTM method is used to train, test the results of the sentiment model and the accuracy rate of the model shows that the

system has a high level of accuracy near to **0.95** for different data extracting.

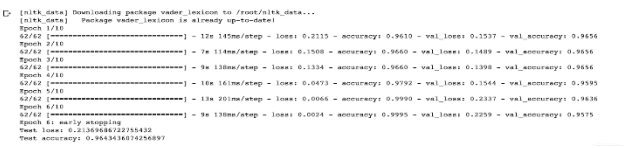


Fig. 4.4 Accuracy Level Obtained by using VADER

6. Recommendation with sentiment analysis:
We collected the tweets for which the user had positive sentiments. We then performed fit transform and cosine similarity to create a matrix between the positive sentiment tweets and the news corpus and, after reversing the scores, recommended the top 200 most similar articles to the user based on his interest.

	text	link
217	listen min comment story comment gift article ...	https://www.washingtonpost.com/nation/2023/04/...
221	listen min comment story comment gift article ...	https://www.washingtonpost.com/politics/2023/0...
86	listen min comment story comment gift article ...	https://www.washingtonpost.com/business/2023/0...
213	president biden say april take hard look age d...	https://www.washingtonpost.com/politics/2023/0...
109	listen min comment story comment gift article ...	https://www.washingtonpost.com/opinions/2023/0...
175	listen min comment story comment gift article ...	https://www.washingtonpost.com/politics/2023/0...
165	listen min comment story comment gift article ...	https://www.washingtonpost.com/politics/2023/0...
315	home single woman man buy home mean easy compe...	https://www.washingtonpost.com/home/2023/04/26...
224	style washington gambler sean young democratic...	https://www.washingtonpost.com/lifestyle/2023/...
137	montana republicans vote bar rep april critici...	https://www.washingtonpost.com/politics/2023/0...

Fig. 4.5 Accuracy Level Obtained by using VADER

7. Recommendation without sentiment analysis:
This time we vectorize all the tweets of the user without any sentiment analysis and perform cosine similarity over the tweets text and the new article corpus to generate the matrix. The matrix is then reversed to recommend the top 200 most similar articles.

	text	link
317	listen min comment story comment gift article ...	https://www.washingtonpost.com/media/2023/04/2...
388	listen min comment story comment gift article ...	https://www.washingtonpost.com/books/2023/04/2...
120	listen min comment story comment gift article ...	https://www.washingtonpost.com/opinions/2023/0...
109	listen min comment story comment gift article ...	https://www.washingtonpost.com/opinions/2023/0...
315	home single woman man buy home mean easy compe...	https://www.washingtonpost.com/home/2023/04/26...
115	listen min comment story comment gift article ...	https://www.washingtonpost.com/opinions/2023/0...
224	style washington gambler sean young democratic...	https://www.washingtonpost.com/lifestyle/2023/...
329	listen min comment story comment gift article ...	https://www.washingtonpost.com/lifestyle/2023/...
469	site information policy standard washington po...	https://www.washingtonpost.com/policies-and-st...

Fig. 4.6 Recommendation without sentiment analysis

Comparison with previous publication results:
In the [7] publication the author has experimented similarly for two recommendation systems with/without sentiment analysis on Fine food reviews and movie reviews.

- The author has implemented hybrid deep learning sentiment analysis using CNN-LSTM model.
- BERT model was used for feature extraction and two deep learning models were used to compare the experiment of the recommendation of the reviews of Amazon movies and fine food.
- The author analyses and evaluates the result accuracy of his system using the reliability of Root-Mean-Square-Error (RMSE) score.

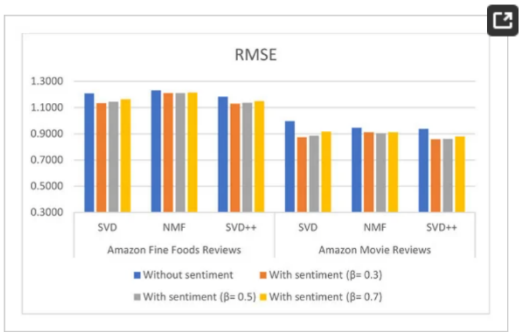


Fig. 4.7 RMSE Score of our system

We see that the error rate for recommendation systems with sentiment analysis is less than without sentiment analysis in all the cases.

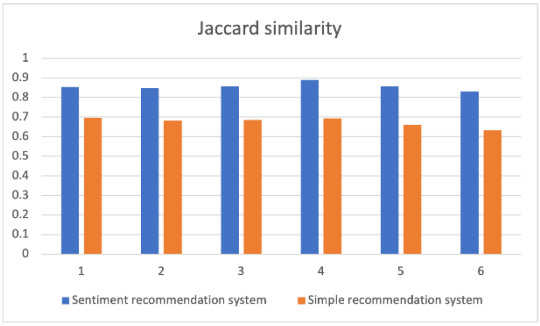


Fig. 4.8 Jaccard Similarity of our system

In our system we have calculated Jaccard similarity and which has greater value for recommendation system with sentiment analysis than without sentiment analysis. The final result output is in the results.

Overall Result:
We calculated the Jaccard similarity score of both the recommendation systems. The Jaccard similarity is a popular similarity measure in recommendation systems that evaluates how similar two collections of items are to one another. It relies on the hunch that two collections of items that have a lot of overlap might have a connection to one another. The Jaccard similarity ranges from 0.0 to 1.0. The higher the Jaccard similarity of the two sets, the higher the implication that they are connected to each other.
In our findings we see that the similarity of the recommendation system using sentiment analysis has higher value than the recommendation system without sentiment analysis. The sentiment recommendation system is having higher similarity because it is recommending users the articles based on the sentiments of the user which is closer to the actual recommendations provided.

Fig. 4.9.1 Result Comparison through Similarity Score

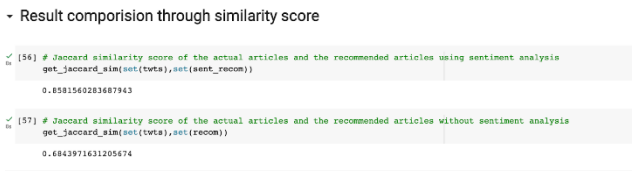


Fig. 4.9.2 Result Comparison through Similarity Score

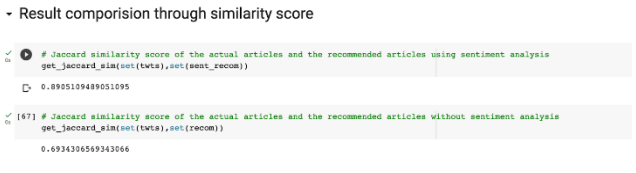


Fig. 4.9.3 Result Comparison through Similarity Score

Fig. 4.9.4 Result Comparison through Similarity Score

The (Fig 4.9.5) shows the similarity score for both the recommendation systems with/without sentiment analysis for multiple runs of the models over the command line of our system.

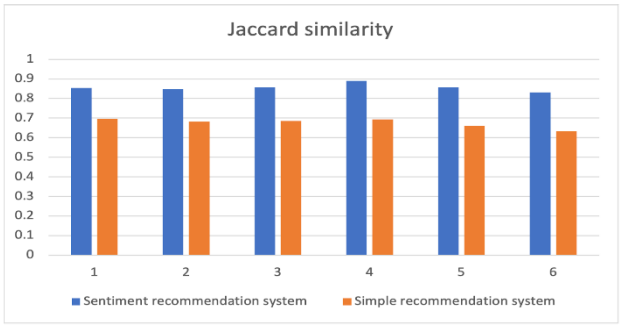


Fig. 4.9.5 Jaccard Similarity Score

V. Conclusion

In this study, we put forward a hybrid deep-learning model-based approach to sentiment analysis in a user-content based recommender system. In relation to social networks, the design can be utilized to create a recommender system that benefits from sentiment analysis performed on user input and evaluations in the network. We tested our theories using the news tweets recommender system. We show the efficacy and practicality of our methodologies in creating customized recommendations on twitter data.

The results show that using deep learning sentiment analysis in a content-based recommendation system significantly improves the performance of the model. This is accomplished by utilizing additional data from feedback from customers and comments. The recommender system is more dependable and capable of giving users better recommendations as a result of its incorporation into the established recommendation techniques.

Better recommendations will engage the users into investing more time over the social network, ultimately improving engagement and enhancement of the businesses and decision making, which is the focus of the social media platforms.

Content based filtering system is a more accuracy-centric approach. When assessing the recommendation quality, these traditional accuracy-centric methodologies might disregard other aspects of subjective user experiences (such as decision satisfaction, perceived system efficacy and exposure to diverse points of view).

As future work, we plan on implementing the hybrid deep learning sentiment analysis on the hybrid recommendation system which uses content based and collaborative filtering for recommendations. This will provide a wider scope and diversify the points for view during recommendation. This can be achieved by similar interest clustering and topic modeling using Latent Dirichlet Allocation (LDA) model. We will also consider using different sentiment analysis techniques such as graph convolutional networks, for potential improvement of the concept.

VI. Discussion

Findings demonstrate that adding deep learning sentiment analysis to a content-based recommendation system significantly improves the model's performance. Additional information from comments and user feedback is used to achieve this. After proper analysis over the results we can seamlessly reach the conclusion that by incorporation into well-known recommendation methodologies, the recommender system is more reliable and capable of offering users more accurate recommendations.
Improved recommendations would encourage users to invest more time on social networks, which will ultimately enhance engagement and improve business and decision-making, which is the primary objective of social media platforms. The output provides a clear interpretation of the result we are aiming for. The results of the other existing systems interpret the same investigation implying that incorporating sentiment analysis in recommendation systems provides better performance as compared to recommendation systems without sentiment analysis. Our system is a content-based recommendation model; thus, the articles are recommended based on the recent feedback or tweets of the user on the social media platform, providing recommendation only for specific topics, thus narrowing the scope of the topics and points of view. The system approach seems to be more accuracy-centric.

VII. Acknowledgement

We would like to thank Professor Saman Kumarwadu and Teaching Assistant David Zhang for their constant support and advise during the whole semester.
Your suggestions, opinions, and experience helped us substantially. Without their help, this work could not have been completed.
We also want to thank Syracuse University's department of engineering and computer science for establishing a research-friendly environment and providing us with the tools we needed to finish our project.

VIII. References

[1] Kharde, V. A., & Sonawane, S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. arXiv preprint arXiv:1601.06960v3.
[2] Siddharth, S., Darsini, R., & Sujithra, M. (2018). Sentiment analysis on twitter data using machine learning algorithms in python. *Int. J. Eng. Res. Comput. Sci. Eng*, *5*(2), 285-290.
[3] Gupta, B., Negi, M., Vishwakarma, K., Rawat, G., Badhani, P., & Tech, B. (2017). Study of Twitter sentiment analysis using machine learning algorithms on Python. *International Journal of Computer Applications*, *165*(9), 29-34.
[4] Khyani, Divya & B S, Siddhartha. (2021). An Interpretation of Lemmatization and Stemming in Natural Language Processing. Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology. 22. 350-357.
[5] Effrosynidis, D., Symeonidis, S., & Arampatzis, A. (2017). A comparison of pre-processing techniques for twitter sentiment analysis. In *Research and Advanced Technology for Digital Libraries: 21st International Conference on Theory and Practice of Digital Libraries, TPDL 2017, Thessaloniki, Greece, September 18-21, 2017, Proceedings 21* (pp. 394-406). Springer International Publishing.
[6] Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., Patel, A., & Shashidhar, R. (2022). Movie recommendation and sentiment analysis using machine learning. *Global Transitions Proceedings*, *3*(1), 279-284.
[7]Dang CN, Moreno-García MN, Prieta F. An Approach to Integrating Sentiment Analysis into Recommender Systems. Sensors (Basel). 2021 Aug 23;21(16):5666. doi: 10.3390/s21165666. PMID: 34451118; PMCID: PMC8402473.
[8] Hung, B. T. (2020). Integrating sentiment analysis in recommender systems. *Reliability and Statistical Computing: Modeling, Methods and Applications*, 127-137.
[9] Al-Ghuribi, S. M., & Noah, S. A. M. (2021). A comprehensive overview of recommender system and sentiment analysis. *arXiv preprint arXiv:2109.08794*.

[10] Osman, N. A., Mohd Noah, S. A., Darwich, M., & Mohd, M. (2021). Integrating contextual sentiment analysis in collaborative recommender systems. *Plos one*, *16*(3), e0248695.
[11] Raza, S., & Ding, C. (2022). News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review*, 1-52.
[12] Effrosynidis, Dimitrios & Symeonidis, Symeon & Arampatzis, Avi. (2017). A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis. 21st International Conference on Theory and Practice of Digital Libraries (TPDL 2017). 10.1007/978-3-319-67008-9_31.
[13] Khyani, Divya & B S, Siddhartha. (2021). An Interpretation of Lemmatization and Stemming in Natural Language Processing. Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology. 22. 350-357.
[14]https://www.nltk.org/
[15]https://pandas.pydata.org/docs/getting_started/index.html#getting-started
[16]https://towardsai.net/p/l/stemming-porter-vs-snowball-vs-lancaster
[17]https://www.sciencedirect.com/topics/computer-science/lexicon-based-approach
[18]https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/
[19] https://en.wikipedia.org/wiki/Web_scraping
[20]]https://www.ibm.com/topics/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another.
[21]https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd