News Recommendation Engine

Ankita Pal, Chavi Gupta, Medha Sagar
University of Washington
apal1994@uw.edu, chavig@uw.edu, sagarme@uw.edu

Abstract

The goal of this project is to build a News Recommender System. We chose to work on a News Recommender system since we were interested in addressing both the dimensionality (associated with large text content) and recommendation problems that were covered in class. Among the different domains of recommender systems, news recommendation has been explored relatively less due to the lack of structured data and features. Text mining for news articles using NLP techniques in itself is a different class of problem. In this project we aim to build a pipeline for extracting user and news article features, and eventually build a hybrid recommender system to address the problems of cold start, data sparcity and scalability.

1 Introduction

"Too many cooks spoil the broth"-Jane Austen.
Today the world is flooded and overwhelmed with news. According to our literature review, more than 2 million articles are published everyday on the web. Finding the suitable article a user wants to read, in this overwhelming amount of news, would be a very tedious job in this scenario. Therefore, we decided to take up the development of a news recommendation engine. This recommendation engine uses user side information as well as the news article information for a hybrid Collaborative Filtering and Content-Based Filtering(CF-CBF) approach, to recommend a news article to the user. We use hierarchical clustering to group the users into clusters based on the number of retweets. We find the topics preferred by user groups to get a CBF recommendation score. We use hierarchical Latent Dirichlet allocation (hLDA) to find the most relevant topics within the news articles. Combining time trends and user interests we find the time specific user interest for each topic, thus getting a CF recommendation score. Finally we take the two recommendation scores and combine them to a single score and recommend top news articles based on the choice of the individual and topic of the article.

2 Relevant Work

Amidst all the research done for recommendation systems, new recommendations have been one of the most challenging and relatively unexplored problems. In the domain of news recommender systems, there are 3 classes of recommendation algorithms- Collaborative Filtering (CF), Content-based Filtering (CBF) and hybrid algorithms. The paper (1), proposes the use of building a hybrid recommendation algorithm. They view the problem in 2 phases - Modeling phase and Recommendation phase. The objective of the modelling phase is to pre-learn user and content models that serve to reduce the dimensionality as well as group similar users/articles together. These modellings are performed offline and stored prior to the recommendation phase. The recommendation phase occurs in real-time where based on a reader query, the results of the CF and CBF approaches are combined together to produce a set of recommendations.
In paper (2), the authors propose an interesting approach to calculate the recommendation score for the content filtering algorithm. The authors propose to compute the user's interest in a category (or topic) based on the user's overall interest in a topic and also the trend of interest for a topic over time.

By combining these "long-term" and "short-term" effects, we can estimate the current interest of a user for a topic. In this project we will use a similar approach for the CBF recommendation.
The paper (3) recommends using retweet information to cluster tweets instead of tweet text. The reason was that the tweets are usually restricted to 280 characters and words are often abbreviated (modern text language) and contain special symbols like hashtags and emojis. Thus the paper proposed to calculate the similarity between tweets based on the overlap rate of users who retweeted them, creating a retweet network followed by clustering the tweets based on network clustering. In this project we decided to use a similar approach to cluster the users based on a retweet network. This user modelling will be used in the CF recommendation approach.
Eventually to combine the recommendations, we will explore the use of a weighted additive recommendation score or a multiplicative score as proposed in (2).

3 Data Collection

In order to create a news recommendation engine, we needed news as well as user data to model the user-item interactions. While there are many text corpora for news article data, our literature surveys showed that there are no readily available data sources to collect user interactions with news items.

Therefore, we decided to use Twitter data to understand how users interact with digital news entities on Twitter. The twitter data used in our project was collected using the FakeNewsNet (4) data repository. The repository contains information about tweets that contain news article information. These news articles are shared on Twitter via tweets and/or re-tweets. The repository collects news articles from two predominant websites:

1. GossipCop
2. Politifact

For the course of this project, we are focussing on news articles supplied by Politifact. Politifact is a non-profit fact checking website reporting on the accuracy of statements made by elected officials, candidates, their staffs, lobbyists, interest groups and others involved in U.S politics. The articles supplied by Politifact are predominantly related to the U.S Politics.

We will be using the news articles and its metadata collected by Politifact to extract features from the news articles. Additionally, we will be studying the tweets and the retweets of the articles to extract features that would be helpful in modeling user behavior.

3.1 Data Collection Process

: Our data collection process can be divided into two phases.

3.1.1 Phase I : Twitter Data Collection

Our data collection process starts by using the FakeNewsNet (4) repository to collect information about news articles supplied by Politifact. Twitter API keys are used for collecting data from Twitter. Each news article has the following information:

1. Article ID : unique identifier of a news text
2. Article Text: the text of the article
3. Article URL: the link to the article
4. Article Keywords: keywords from the article
5. Article Published On: date on which the article was published
6. Tweet Info:
    - Tweet ID: unique identifier for the tweet that contains the article
    - Tweet Timestamp: timestamp when tweet was tweeted
    - Tweet Retweet: no. of time the tweet was retweeted
    - Tweet favorite: no. of timed tweet was favorited

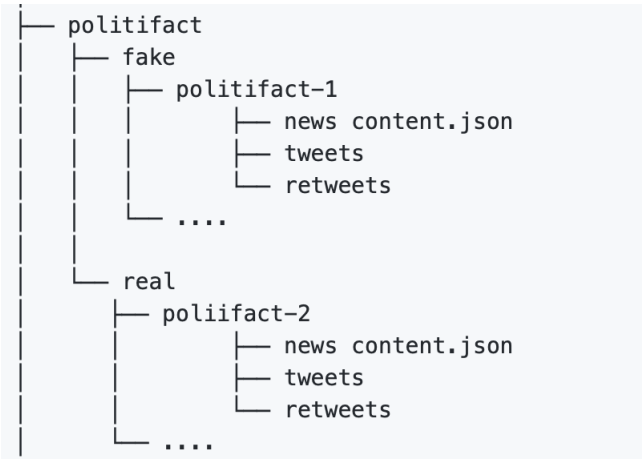

Figure 1: Data Hierarchy

- User Info:
  - User ID: unique identifier for the user who tweeted about the article
  - User Location: location of the user
  - Follower Count: no. of followers the user has
  - Friend Count: no. of friends the user has
  - User Favourite Count: no. of favorite tweets
  - User Status Count: user statuses count

7. Retweet Info:
   - Retweet ID: Unique Identifier for the retweet
   - Retweet timestamp: timestamp when tweet was tweeted
   - Retweet UserID: unique identifier of the user who retweeted the tweet
   - User Info:
     - User ID: unique identifier for the user who tweeted about the article
     - User Location: location of the user
     - Follower Count: no. of followers the user has
     - Friend Count: no. of friends the user has
     - User Favourite Count: no. of favorite tweets
     - User Status Count: user statuses count

3.1.2 Phase II : News Article Collection

In order to collect data for modeling the news article text, we used 'All the News' dataset on Kaggle (5) which contains 143,000 news articles from 15 different American publications. The motivation to use additional news data was to make a generalizable model for feature extraction on the textual article data.

Data Statistics: Since, the data downloading time and the time required to parse the tweets is exorbitant, we report the following statistics as it stands:

1. Total Number of Articles: 515
2. Total Number of Articles Processed: 285
3. Total Number of Tweets Processed: 239051
4. Total Users Processed Processed: 155353

Challenges:
1. Absence of publicly available data: As mentioned above, after rigorous literature review we realized that while there are many text corpora for news articles, there does not exist any publicly

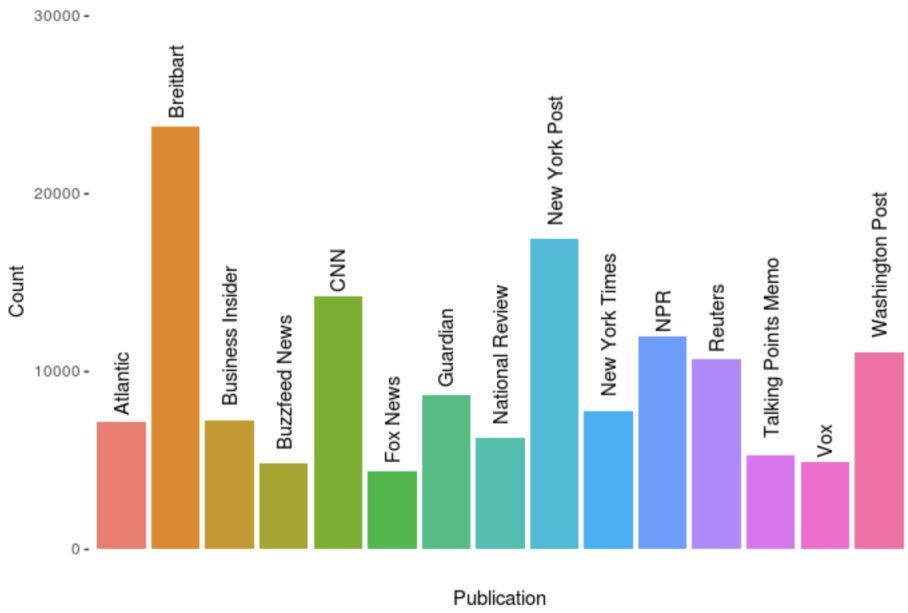

Figure 2: Distribution of News Articles and Publication

available dataset that captures the user interactions with news articles. Therefore, we had to create a dataset from scratch to collect the user interactions and news articles. We decided to leverage Twitter data to solve this problem. We have used tweets and retweets as a measure of interaction between users and digital news articles on Twitter.

2. Large Amount of Hierarchial Data: For the scope of this project, we decided to just use Politifact news articles to model our recommendation engine. For a total of 515 news articles, we realized that the due to the hierarchical nature of Twitter data, the user-item interactions are huge. For example, for one article, we can have as many as 30,000 tweets for which we need to parse both the user information and the tweet information. Each of these 30,000 tweets can further have hundreds of retweets. Those retweets would again have to be aggregated along with their author information to create dense user networks. A representation of this hierarchy is depicted in Figure 1. This leads to huge processing times to construct the data. In order to overcome the high computing cost, we are running our analysis and parsing using Google Cloud Platform.

2. Skew in News Articles: Since, we are just using Politifact articles, we realized that creating LDA models using just the Politics related articles might lead to non-generalize results. To overcome this issue, we used another dataset to train our LDA model. This article dataset contains 143,000 articles from 15 different American publications on various news topics. You can see the distribution of articles and publications in Figure 2.

4 Exploratory Data Analysis

We start our EDA with the analysis of our news articles. We have a collection of 515 articles. For our EDA, we aim to analyse the topics present in our news articles data.
For this, we train our data from an external data source(5) and then analyse how it categorises the news articles. The data process for the same is 3:

For data prepossessing we use the steps including:

- Removal of special characters
- Removal of stop words
- Making Bigrams

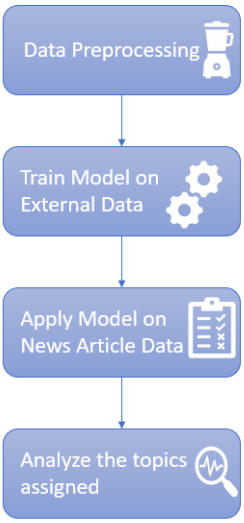

Figure 3: Workflow

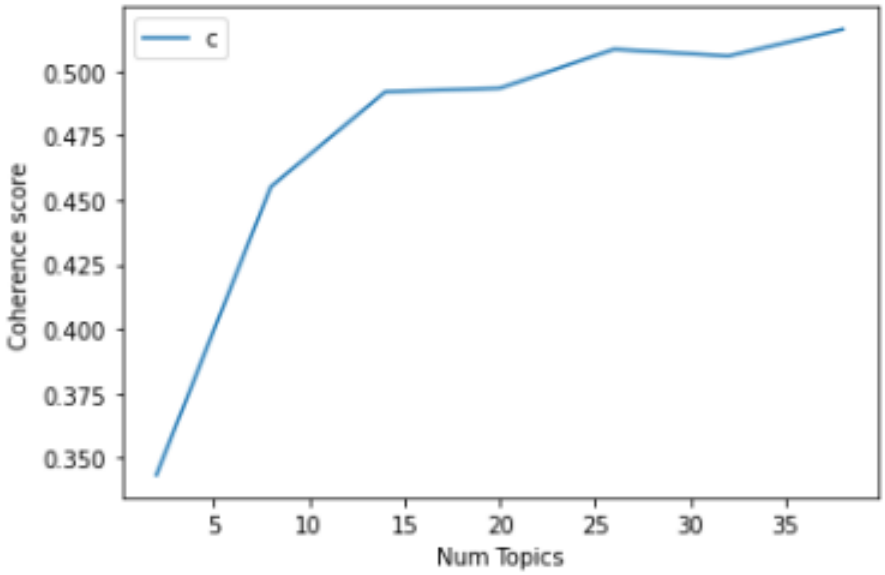

Figure 4: Topic Number vs Coherence Score

- Lemmatization

Our next step was to use this data to model the topics for the external data. We plot the coherence score vs number of topics to help us get the optimum number of topics in Figure 4.

We notice that the optimum number of topics is 15. We create a LDA model with this hyper parameter on the training data of (5). We get the topics and the corresponding word weights as shown in Figure 5.

Next we apply the model on the dataset of 515 news articles, to get the predominant topics in the dataset. Assigning the topic with the maximum score to the document and plotting the count of documents for each topic gives the Figure 6.

```
[(0,
  '0.020*"news" + 0.018*"call" + 0.016*"medium" + 0.014*"comment" + '
  '0.013*"write" + 0.012*"post" + 0.012*"show" + 0.012*"event" + 0.011*"tweet" '
  '+ 0.010*"twitter"'),
 (1,
  '0.019*"attack" + 0.013*"city" + 0.012*"isis" + 0.012*"group" + '
  '0.012*"military" + 0.011*"people" + 0.009*"kill" + 0.008*"plane" + '
  '0.008*"official" + 0.008*"terrorist"'),
 (2,
  '0.022*"people" + 0.020*"get" + 0.016*"think" + 0.015*"time" + 0.014*"know" '
  '+ 0.013*"see" + 0.013*"take" + 0.012*"want" + 0.011*"come" + 0.011*"look"'),
 (3,
  '0.019*"country" + 0.009*"government" + 0.008*"world" + 0.008*"leader" + '
  '0.008*"power" + 0.007*"american" + 0.007*"may" + 0.006*"many" + '
  '0.005*"deal" + 0.005*"policy"'),
 (4,
  '0.055*"woman" + 0.040*"student" + 0.023*"black" + 0.021*"man" + '
  '0.021*"white" + 0.016*"college" + 0.014*"group" + 0.013*"people" + '
  '0.010*"young" + 0.009*"community"'),
 (5,
  '0.027*"company" + 0.016*"business" + 0.014*"market" + 0.011*"price" + '
  '0.010*"money" + 0.009*"accord" + 0.009*"pay" + 0.008*"investor" + '
  '0.008*"cost" + 0.007*"high"'),
 (6,
  '0.019*"report" + 0.013*"official" + 0.012*"email" + 0.010*"information" + '
  '0.010*"former" + 0.008*"accord" + 0.008*"investigation" + 0.008*"case" + '
  '0.008*"intelligence" + 0.008*"release"'),
 (7,
  '0.035*"family" + 0.030*"child" + 0.024*"school" + 0.018*"drug" + '
  '0.014*"patient" + 0.013*"pain" + 0.011*"parent" + 0.011*"live" + '
  '0.010*"death" + 0.009*"teacher"'),
 (8,
  '0.052*"police" + 0.036*"officer" + 0.018*"man" + 0.016*"kill" + '
  '0.015*"shoot" + 0.014*"gun" + 0.012*"authority" + 0.011*"suspect" + '
  '0.010*"charge" + 0.010*"arrest"'),
 (9,
  '0.115*"trump" + 0.032*"campaign" + 0.031*"election" + 0.021*"vote" + '
  '0.014*"presidential" + 0.014*"state" + 0.014*"republican" + 0.013*"voter" + '
  '0.013*"candidate" + 0.012*"democratic"'),
 (10,
  '0.031*"state" + 0.025*"law" + 0.017*"federal" + 0.014*"rule" + '
  '0.013*"policy" + 0.012*"plan" + 0.011*"bill" + 0.010*"court" + '
  '0.010*"issue" + 0.009*"pass"'),
```

Figure 5: Word Weightage in Topics

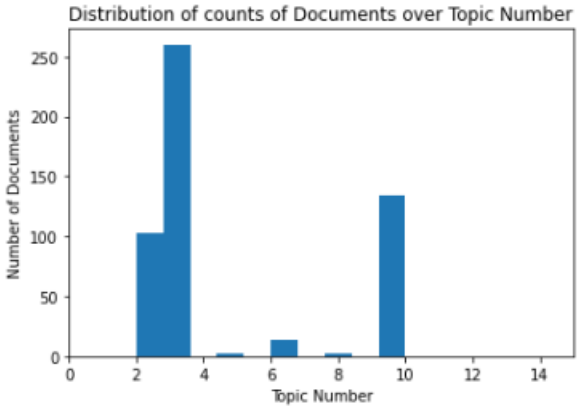

Figure 6: Topic number vs Count of documents in news articles data

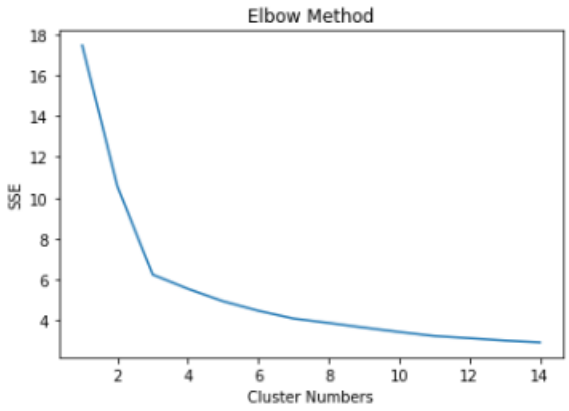

Figure 7: Number of clusters vs SSE

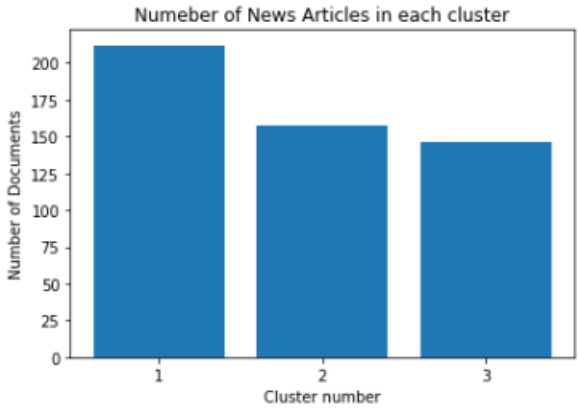

Figure 8: Number of Documents in each cluster

Thus we see that only 3 of the topics are pre-dominant in our document. By analysis of the keywords we see all three of these topics are related to politics. Also we observe that 2 and 9 are the topic numbers which are most frequently occurring.

Also, we use k-means to find the distribution of the topics in the news articles data. First to find the optimum number of clusters in the data, we plot a SSE and number of clusters plot. We get the following graph as shown in Figure 7.

We find that the number of clusters is 3 and this is consistent with the find from figure 6. The distribution of documents for among the cluster is shown in Figure 8

Thus, we conclude that we need to further divide the topics into sub-topics for analysis.

5 Algorithm

Understanding the relationship in the data: The data is composed of users, tweets and news articles. The users are linked to tweets either by directly tweeting the articles or by retweeting the articles. Each tweet refers to a news article.

The algorithm is divided into 3 major components. The description of the algorithm for each step is described in the following sections.

5.1 User Interest Modelling (Content-Based Filtering Component)

5.1.1 LDA for Topic Modelling

For this component the first step is to generate classify the news articles into different categories (or topics). The purpose of this is to reduce the dimension space of the news articles and to group similar articles based on text mining. For creating these topics we will use hierarchical topic modelling since the news articles in our dataset are already known to be related to Politics. Thus we classify each news article under a topic $\{c_1, c_2, c_3, \ldots\}$.

5.1.2 User's Genuine News Interest in Time Period t

The genuine interest of a user in a topic (categorized by LDA) is represented as follows:

$$\begin{aligned} interest_{u,t}(topic = c_i) &= p_t(article|topic = c_i) \\ &= \frac{p_t(topic = c_i|article)p_t(article)}{p_t(topic = c_i)} \end{aligned} \tag{1}$$

Where $p_t(topic = c_i|article) = \frac{\text{number of articles tweeted by user in topic } c_i}{\text{number of articles tweets by user (during time period } t)}$,

$p_t(article) = \frac{\text{number of articles tweeted by user}}{\text{total number of articles (during time period } t)}$,

$p_t(topic = c_i) = \frac{\text{number of articles in topic } c_i}{\text{number of articles during time period } t}$.

5.1.3 Combining interest with past trend

The overall interest of a user u in topic c_i is:

$$\begin{aligned} interest_u(topic = c_i) &= \frac{\sum_t (N^t \times interest_{u,t}(topic = t_i)}{\sum_t N^t} \\ &= \frac{\sum_t \left(N^t \times \frac{p_t(topic = c_i|article)p_t(article)}{p_t(topic = c_i)} \right)}{\sum_t N^t} \end{aligned} \tag{2}$$

Where N^t = Number of articles tweeted by user in time period t.

5.1.4 Predicting the User's Current News Interest

We represent the current time period with subscript 0. Thus the interest at the current time period is as follows:

$$interest_{u,0}(topic = c_i) = \frac{p_0(article|topic = c_i) \times p_0(topic = t_i)}{p_0(article)} \tag{3}$$

We estimate $p_0(article|topic = c_i)$ by $interest_u(topic = c_i)$.
Also, we assume the the probability of tweeting about any news article is constant.

$$\begin{aligned} \therefore interest_{u,0}(topic = c_i) &\propto \frac{interest_u(topic = c_i) \times p_0(topic = t_i)}{p(article)} \\ &\propto \frac{p_0(topic = t_i) \times \sum_t \left(N^t \times \frac{p_t(topic = c_i|article)}{p_t(topic = c_i)} \right)}{\sum_t N^t} \end{aligned} \tag{4}$$

5.1.5 Smoothing factor G

If the user is found to tweet very rarely or in the case of a new user (cold-start problem), we would like to approximate a user's interest based on the current trend. Thus we add a smoothing factor G that act as virtual tweets. If $\sum_t N^t$ is much larger than G, then the interest approximates to the user's real interest. However if it is very small, then the interest will approximate to the general current news trend.

$$interest_{u,0}(topic = c_i) \propto \frac{p_0(topic = t_i) \times \left(\sum_t \left(N^t \times \frac{p_t(topic = c_i|article)}{p_t(topic = c_i)} \right) + G \right)}{\sum_t N^t + G} \quad (5)$$

Thus the Content-Based Filtering Score for a user u and topic c_i is $interest_{u,0}(topic = c_i)$.

Note that offline we can precompute N^t and $\frac{p_t(topic = c_i|article)}{p_t(topic = c_i)}$. Thus in real time we only have to calculate the $p_0(topic = c_i)$ i.e. the probability of a topic currently being in trend.

5.2 User clustering (Collaborative Filtering Component)

The following sections explain the algorithm for clustering users based on retweet networks.

5.2.1 Create User Retweet Network

1. Create a User-Tweet matrix W where $W_{u,t} = \frac{1}{\text{number of retweets on the tweet } t \text{ by user } u}$
2. Get user weights by finding the sum of each row in the W matrix. Let this user weight matrix be V.
3. Create a User-User matrix where $U_{u1,u2}$ = (Sum of $\left(\frac{1}{\text{number of retweets}} \right)$ for each common retweet between user $u1$ and $u2$)
4. Calculate the cosine similarity of each user link as $\frac{U_{u1,u2}}{V_{u1} \times V_{u2}}$.
5. Create a graph G based with each user as a node and each edge is the link between each user with edge weight as the cosine similarity and edge value as the number of common retweets.
6. Cluster the graph using the fast greedy algorithm.

Thus we get groups of users. Using the results of the LDA from the previous section, we can create an interaction matrix I where I_{g,c_i} = average of probabilities of category c_i read by the users in user group g

We finally get the recommendation score based on Collaborative Filtering by using I_{g,c_i} where user u belongs to user group g and the article belongs to topic c_i.

5.3 Recommendation Generation Component

From the above two sections we saw how we can compute two recommendation scores for each news article for a user. Let the recommendation scores from the Content-Based Filtering component be CF and Collaborative Filtering component be CBF. We can calculate a single recommendation score for each article as $CF \times CBF$.

Finally using the recommendation scores for each article for each user, we can recommend the top news articles for any user.

6 Conclusion

In the current news recommendation systems, the major drawback lies in the absence of user interactions with news articles. Our literature survey shows most of the current systems are content based. This was a major motivation behind our project. We wanted to create a system that leverages both user and news data to generate recommendations.

References

[1] Ismail, Walaa & Nasr, Mona & Saied, Mohamed. (2018). A HYBRID NEWS RECOMMENDER SYSTEM.

[2] Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behavior. In Proceedings of the 15th international conference on Intelligent user interfaces (pp. 31-40).

[3] Uchida, K., Toriumi, F., & Sakaki, T. (2017, August). Evaluation of retweet clustering method classification method using retweets on Twitter without text data. In Proceedings of the International Conference on Web Intelligence (pp. 187-194).

[4] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286.

[5] Thompson, A. (2017, August 20). All the news. Retrieved from https://www.kaggle.com/snapcrack/all-the-news