News Article Recommendation System

Keerthana Sundaresan
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
2017103058@student.annauniv.edu

***Abstract*—This project aims to create a recommendation system that would predict or filter preferences according to the user's choices. The goal is to recommend news articles which are similar to the already read article by using attributes like article headline, category, author and publishing date. The method of implementation is that of a content-based recommendation system, which recommends items based on the user's past preferences. Two methods of content filtering are implemented and their results are compared.**

***Index Terms*—news recommender, content-based filtering, natural language processing, feature extraction, information retrieval**

I. Introduction

The amount of data and services available, and our dependence on them, grows every day. With such magnitude of data, it can become very time consuming to find the service you desire without hassle. This is where recommender systems come into play. A recommender system is an algorithm used to suggest items relevant to a particular user. Recommender systems are crucial in creating a satisfying user experience, and can generate huge profits when used by companies in a way that benefits their users and stands out from the competition.

One such area where recommender systems can play a critical role is in the recommendation of news articles. With the large number of news articles available at our fingertips, a successful recommendation system can enhance the user experience by presenting the reader with the articles they would be most likely be interested in. In this project, the aim is to explore different methods of information retrieval and create a recommendation system for news articles.

II. Background

Content-based recommendation works using similarity between users and items. Users and items have profiles describing their characteristics and the system would recommend an item to a user if the two profiles match. Content-based recommendation uses metadata about the users or items to determine similarity.

To determine similarity, the concept of distance is usually used. If distance is low, similarity is high and if distance is high, similarity is low. To calculate the distance, the headline must be represented as a d-dimensional vector. The two approaches that have been explored here are the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) algorithms.

A. Bag of Words

Bag of Words is a Natural Language Processing model used to extract features from text. As many algorithms cannot work directly with raw text, the text must be converted into numbers. This is the role of algorithms like Bag of Words. Bag of Words model is used to pre-process text by converting it into a bag of words, which keeps a measure of the total occurrences of most frequently used words. It is called a bag of words because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document they occur.

A bag-of-words involves (i) a vocabulary of known words and (ii) a measure of the presence of known words. The measure may be represented in the following ways:

- *Binary Scoring*: Checking whether each word is present or absent – marking an absent word as 0 and an occurring word as 1
- *Count*: Counting the amount of times each word is present in the document
- *Frequency*: Calculating the frequency that each word appears in a document out of all the words in the document.

The first step in using Bag of Words is to clean the data (corpus) and extract the unique relevant words from it. This forms the vocabulary. After the vocabulary has been chosen, we score words using one of the measurement techniques mentioned above. In this project, each headline can be considered as a document and set of all headlines form a corpus. Using the Bag of Words approach to score each word in a document, each document is represented by a d-dimensional vector, where d is total number of unique words in the corpus.

B. TF-IDF

One disadvantage of the Bag of Words method is that it does not give importance to low frequency words that are important but not often occurring in the documents of the corpus. To overcome this disadvantage, we use the TF-IDF method for feature representation. It is a ranking factor that focuses more on term frequency rather than on counting keywords. TF-IDF checks how relevant the keyword is throughout the corpus.

TF-IDF stands for term frequency-inverse document frequency. TF-IDF weighs a term's frequency (TF) and its inverse document frequency (IDF).

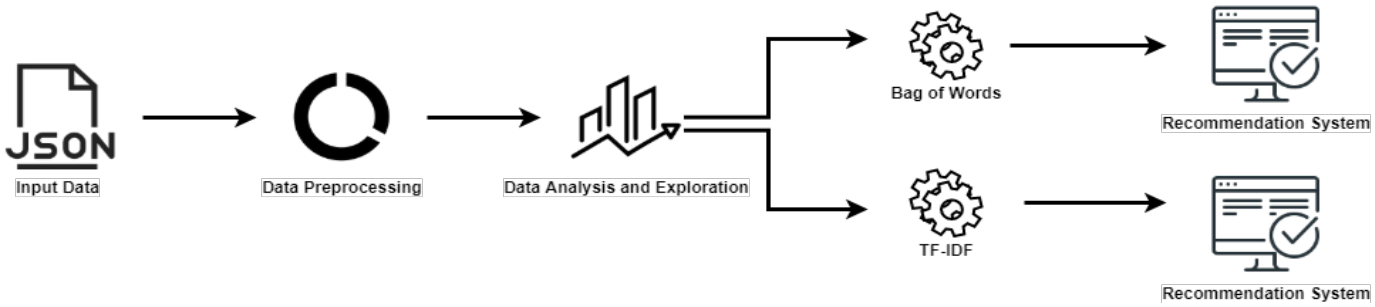

Fig. 1. Architecture Diagram

TF and IDF are measured as follows:

- *TF*: The term frequency of a word in a document
- *TF(i,j)* = (no. of times word i appears in document j)/(no. of words in document j)
- *IDF*: The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. IDF(i,D) = log_e(no. of documents in the corpus D)/(no. of documents containing word i)

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.

$$weight(i,j) = TF(i,j) \times IDF(i,D) \tag{1}$$

If a word occurs a greater number of times in a document but a smaller number of times in all other documents in the corpus, then its TF-IDF value will be high, as IDF value will be higher when a word is less common in the entire corpus.

III. Experimental Results

A. Dataset

The News Category dataset from Kaggle [1] was used to build the recommender system. This dataset contains approximately 200k news headlines from the year 2012 to 2018, and has been obtained from HuffPost [2]. Each news headline in this dataset has a corresponding category such as 'Politics', 'Wellness', and 'Entertainment'. Every record is defined by the category, headline, author, link, short description and date of publishing.

B. Procedure

The main packages which are imported for text processing are NLTK (Natural Language Toolkit) and sklearn.feature_extraction.

NLTK [4] is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Scikit-learn [5] is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. The sklearn.feature_extraction module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image.

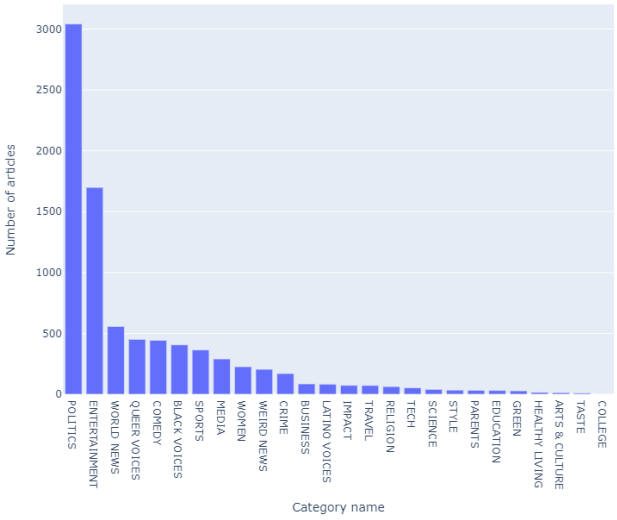

Fig. 2. Distribution of articles category-wise

Fig. 1 shows the architecture diagram of the recommender system. The input data is accepted as a JSON file from Kaggle, which then undergoes data pre-processing in order to clean the large dataset. To obtain a better understanding of the data being handled, certain analysis and data exploration is performed in the form of bar charts and graph functions. The data is then sent through the various algorithms - here, Bag of Words and TF-IDF - and finally multiple recommender systems are developed based on the given algorithms. Output of recommendations are displayed.

The data pre-processing is performed on a copy of the dataset to obtain the data in the desired format and shape. The size of the dataset is reduced by considering news articles from only 2018. The text pre-processing involves the removal of stop words such as 'a', 'an' and 'the', which is performed by making use of the stopwords module of the NLTK corpus.

Stop words and headlines which are less than 5 words in length are removed.

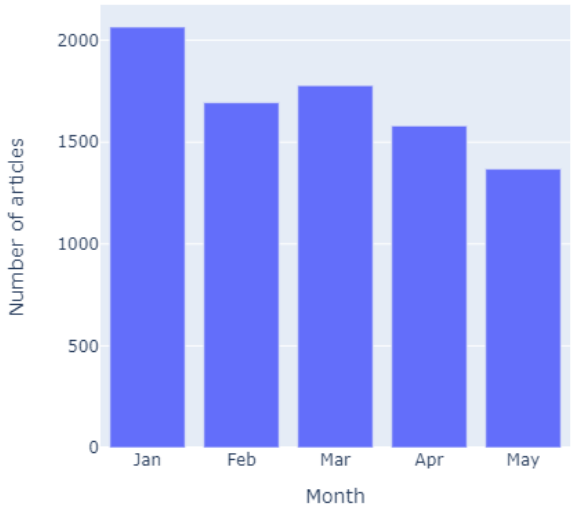

Fig. 3. Distribution of articles month-wise

Lemmatization is then performed by using the WordNetLemmatizer method imported from the NLTK corpus. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Finally, it is made sure that there are no duplicate headline appearances by removing the duplicate items, and a dataset of size (8485, 6) is obtained.

Before building the models, basic data analysis and exploration is performed to learn the distributions and characteristics of the dataset - distribution of articles across categories and dates. The number of samples (articles) per category is captured in a bar graph and displayed as in Fig. 2. It is observed that the 'Politics' category has the highest number of articles (3042), followed by 'Entertainment'. The graph shows that the news distribution is imbalanced – first three most well represented categories, 'Politics', 'Entertainment' and 'World News', if combined, make up around 44% of all data samples.

From Fig. 3 the result of grouping the samples on a monthly basis is captured in a bar graph. The data shows that the month of January has the highest number of articles (2065). The probability distribution function (PDF) of the length of the headlines is obtained, as displayed in Fig. 4. It is similar to a Gaussian distribution, in which most of the headlines are 58 to 80 words in length.

Finally, recommendation systems for Headline-based similarity of news articles are built using two different approaches:

- Bag of Words approach
- TF - IDF approach

In both approaches, the function recommends 10 similar articles to the queried article based on the headline. It accepts two arguments - index of already read articles and the total

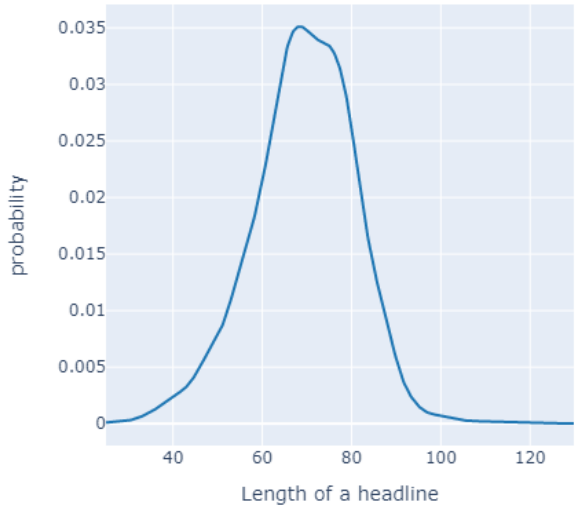

Fig. 4. PDF

number of articles to be recommended. Based on the Euclidean distance it finds out 10 nearest neighbours and makes recommendations. Here, the headline is "Woman Fired After Flipping Off Trump's Motorcade Sues Former Employer". The 10 recommended articles are displayed with details of the published date, headline, and the Euclidean similarity with the queried article.

It is observed that in comparison to the Bag of Words approach, the TF-IDF method recommends articles with headlines containing words such as "employer", "fire" and "flip" in top five recommendations, and these words occur less frequently in the corpus.

IV. Future Areas of Study

A disadvantage in both the approaches implemented, i.e. Bag of Words and TF-IDF, is that they do not capture the semantic and syntactic similarity of a given word with other words. For example, there is an association between "trump" and "white house", "office" and "employee", "tiger" and "leopard", and so on. Such similarities may be captured using word embedding techniques, which leverage semantic similarity between words.

Further, the model currently makes recommendations based only on one feature, i.e. the headline. In order to develop a robust recommendation system, multiple features at a time may be considered from the given dataset, including category and author.

V. Conclusion

In this project, a content-based recommendation system was successfully developed in order to filter news articles based on the given headline. Various approaches were implemented and their performances compared. The top 10 articles of closest

```
============================== Queried article details ==============================
headline :  Woman Fired After Flipping Off Trump's Motorcade Sues Former Employer
```

```
 ======================== Recommended articles :  ======================
```

	publish_date	headline	Euclidean similarity with the queried article
1	2018-02-21	All They Will Call You Will Be Deportees	3.162278
2	2018-02-12	What You Should Know About Trump's Nihilist Budget	3.162278
3	2018-03-09	Trump's Abstinence-Only Pamphlet Is Quite Educational	3.316625
4	2018-01-12	People Are Suggesting How To #FixTrumpIn5Words	3.316625
5	2018-03-18	This Democrat Just Became The Longest-Serving Woman In The House	3.316625
6	2018-01-13	A Deconstruction Of The Alt-Right Movement	3.316625
7	2018-01-11	Why The California Mudslides Have Been So Deadly	3.316625
8	2018-02-12	Trump's Deficit Is Fine. His Budget Is Terrible.	3.316625
9	2018-01-25	Where The Work-For-Welfare Movement Is Heading	3.316625
10	2018-02-01	The Next Financial Crisis -- Not If, But When	3.316625

```
============================== Queried article details ==============================
headline :  Woman Fired After Flipping Off Trump's Motorcade Sues Former Employer
```

```
 ======================== Recommended articles :  ======================
```

	publish_date	headline	Euclidean similarity with the queried article
1	2018-04-26	Cardi B's Former Manager Sues Her For $10 Million	1.263328
2	2018-05-21	The Supreme Court Just Made It A Lot Harder For You To Sue Your Employer	1.276155
3	2018-03-17	Former FBI Deputy Director Andrew McCabe Is Fired 2 Days Before Retirement	1.280543
4	2018-05-12	Der Spiegel Cover Portrays Trump As A Finger Flipping Off Europe	1.287640
5	2018-02-14	Former Employee Sues Vice Media For Allegedly Underpaying Female Staffers	1.287641
6	2018-05-03	Former American University Student President Sues Andrew Anglin For Racist 'Troll Storm'	1.290762
7	2018-01-09	Big Tax Game Hunting: Employer Side Payroll Taxes	1.291920
8	2018-01-24	Garrison Keillor's Former Station Reports He Was Fired For More Than Touching A Woman's Back	1.293396
9	2018-01-16	State Employer Side Payroll Taxes And Loser Liberalism	1.293850
10	2018-05-01	California Sues Trump's EPA Over Weakened Clean Car Rules	1.297878

Fig. 5. (a) Bag of Words method (b) TF-IDF method

relation were displayed. Further improvements for the model were also explored.

VI. Acknowledgements

I would like to thank Mr. G. Manikandan for his guidance throughout this project, as a part of the Big Data Analytics course taken up in the 6th semester of my undergraduate program.

References

[1] News Category Dataset, https://www.kaggle.com/rmisra/news-category-dataset
[2] The Huffington Post, https://www.huffingtonpost.com/
[3] What is TF-IDF?, https://www.onely.com/blog/what-is-tf-idf/
[4] NLTK, https://www.nltk.org/
[5] Scikit-learn, https://scikit-learn.org/stable/