|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  | | PROJECT REPORT | | |  | |
|  | COMPARISON OF BAG OF WORDS AND TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY FOR TEXT CLASSIFICATION ON NEWS CATEGORY DATASET | | | | |  |
|  | | |  |  | | |
|  | | | TWINKLE BHIMANI - 17BCP010YASH BOKIL - 17BCP012TANMEET BUTANI - 17BCP01314th May 2020—NATURAL LANGUAGE PROCESSING—**DR. SANTOSH KUMAR BHARTI** |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | |  |  | |
|  | | INTRODUCTION | | | | |  | |
|  | |  |  | | |  |  | |
|  | TEXT CLASSIFICATION USING NLP Natural Language Processing can be used to classify text based upon the query as required by the user. This has many applications in real world like suggestions from Google, News Classification based on category, Title suggestion for movies, songs etc. All of these applications can be implemented using many datasets as training modules, differently developed NLP pipelines, classification methods. | | | | | | |  |
|  | OUR IMPLEMENTATION | | |  |  | | |  |
|  | We have performed Text Classification based upon the News Category Dataset available from Kaggle to show other related articles from the dataset. We have used two methods of text classifications to show the difference in recommended articles so as to measure the difference between the methods used for text classification. To keep the differences stable, we have kept the rest of the parameters almost as same as possible for both of the methods. We also discuss advantages and disadvantages of both the methods while comparing them. | | | | | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | |
|  | | LITERATURE SURVEY | | |  | |
|  | |  |  |  |  | |
|  | The Modern technology era has brought a change in lifestyle in many domains. The medium to publish the news has become faster, Smarter and more advanced due to information Technology. It has been flooded with immense amount of data, which being published every minute of every day in the form off comments, news sharing, articles, comments, social media platforms, websites, applications, and many more. There are numerous genres for publishing of news nowadays such as Sports, Politics, trending news, Crime news, Entertainment News, Health News, Science, Technology. A person reads different types of news articles depending upon his or her interests. Huge number of news articles are published every-day by various news agencies, independent bloggers and reporters. Hence in this vast sea of news articles in order to provide a reader what he wants to read, is a challenging and important task. For this, the news recommendation system is implemented so that the person is able to read different news in his or her field of interest.  Even with monstrous information on the Internet, clients regularly ''lost themselves''. Customized suggestion innovation has made advancements in territories, for example, web based business, promoting, sound and video suggestion as of late. Because of the intrinsic attributes of system news, for example, the huge information, heterogeneity, update and change quick, practicality and solid topographical mindfulness thus on, the advancement of customized suggestion innovation in the field of news falls behind the above zones. Also, it can't meet the necessities in news field totally. In this manner, it is the primary assignment of the present news suggestion framework to coordinate the current customized proposal advancements into the news suggestion field, to concentrate how to deal with monstrous heterogeneous news information, to develop an ideal client inclination model, and to improve the general execution of the customized news suggestion.  This project presents customized news suggestion according to the previous read news. | | | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | |
|  | | PROPOSED SYSTEM | | |  | |
|  | |  |  |  |  | |
|  | THE DATASET The dataset consists of more than 200,000 rows of news data dating from 2012-2018 from Huffington Post. It is in json format and contains various columns like category, headline, authors, links, short\_description, and date. The only columns used as features for actual Model Training is category and headline. The rest of the columns are used for data cleaning, preprocessing and warehousing. The dataset has a usage rating of 10.0 from Kaggle and has the open public CC0 usage license. The website also suggests various kernels on the website some of which have been cited for use. DATA CLEANING AND PREPROCESSING From the dataset, we first clean the data to produce most relevant results. Making sure that maximum accuracy is achieved for comparison. And the dataset remains neutral to both the methods used for text classification. The first step of cleaning filtering the news to date only from 2018 to the end, making sure that the news is updated according to recent times. Next we move on to remove the duplicate news headlines, the dataset mostly does not have any duplicate rows, but to adhere to the rules of data cleaning, this step has been taken care of. Another step to be taken care of is to look for missing data if any, again, the dataset is very compliant with ML techniques to take care of this itself, but we have taken care of it as well. Next, we learn a few statistical things about the dataset so as to make sure that all the features we require for extraction are available readily by plotting statistical graphs of the data. We complete the step by creating the required fields and making a temporary copy of the dataset to work on. This dataset will work as the raw data for preprocessing pipeline. TOKENIZATION The query string is first tokenized using the tokenization module from nltk library which helps to convert every word in the query string into a separate token which can then be sent forward for token-wise cleaning as well as stop-word removal and also lemmatization STOP-WORD REMOVAL This process is carried out using the nltk library in python which as a stopwords module for various languages. We tokenize the query requested by the user and compare if the word is in the stop words list. If it is there, then it is remove, else kept. LEMMATIZATION The Lemmatization is performed using nltk library again which translates the branch words to root words so that it becomes easy to compare them to the existent dataset. FEATURE EXTRACTIONCOUNT VECTORIZATION Count Vectorization is used for normalization of the required features to vectors (or simply converting text values to numeric values with multiple columns in one row) so that it becomes easy to calculate the similarity between two vectors using numeric distance calculation methods.  We have used the Count Vectorizer module from sklearn feature extraction library which uses the required feature to be translated to an appropriate vector so that it can further be used for classification.  This method of vectorization can be used to classify text based upon bag of words method because it returns integer based vectors. TF-IDF VECTORIZER Text Frequency – Inverse Document Frequency Vectorizer is also used for the normalization of the text in the training dataset to be converted to numeric vectors for easy numeric similarity calculation.  We have used the TF-IDF Vectorizer module available from sklearn feature extraction library which again uses the required feature to an appropriate vector.  This method of vectorization can be used to classify text based upon tf-idf based model because it returns float based vectors. CLASSIFICATION OF TEXTBAG OF WORDS MODEL Bag of Words creates a yes-no presentation of words that are present in a string to translate it into a binary vector which can then be used for calculating the distance between two strings so that the query string can be classified according to the rest of the strings in the dataset. The Count Vectorizer mentioned above helps create this vector and train a model based on various of these vectors so that it creates a set of all the words present in the dataset, this set can be referred to as the bag of words. All the tokens present in the query string are then compared to the words present in the set as to find a numerical distance (or angular distance) between two of those strings so as to classify it as a string in the same or different class. TF-IDF MODEL Term Frequency – TF calculates the score of a word in a query string by taking the logarithm of the ratio of the frequency of a word in the string and the total no of words in the string. Inverse Document Frequency – IDF calculates the score of a word in the dataset by taking the logarithm of the ratio of the total no of rows containing the word and the number of rows the word in it. These two scores are multiplied to get a TF-IDF score for every word in the query string. Making a float vector for the query string. This vector is then used for calculating the distance between the strings in the dataset. | | | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | IMPLEMENTATION AND RESULTS | | |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | CODE IMPLEMENTATION Reading the dataset as a pandas dataframe    Cleaning the dataset  Setting the dataset to use articles after 1st January 2018.    Removing duplicate entries    Filling in blank cells    Statistical analysis of the data  Plot 1: Category vs Articles      Plot 2: Month-wise distribution of articles      Indexing, refactoring and copying data to work on respectively    Using tokenization and stop-word removal for pre-processing    Using lemmatizer to reduce to the root word    Using count vectorizer to vectorize, fit and transform dataset to train model and Bag of Words model to classify query string      Using TF-IDF vectorizer to vectorize, fit and transform the dataset to train model and TF-IDF model to classify query string.     RESULTS Bag of Words Results    TF-IDF Results |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | Conclusion | | |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | Thus, the TF-IDF model is able to give more accurate classifications than Bag of Words, due to its implementation word wise query matching from the dataset.  Given that the training dataset (therefore supervised learning) is properly curated and parameters of training are taken care of ADVANTAGES OF BOW  * The BoW method is simple to implement and is easily studied through simple observation. * It uses Euclidean distance to calculate similarity between vectors so faster implementation is possible.  DISADVANTAGES OF BOW  * It gives lesser importance to words no occurring in dataset. * It does not follow the words in order.  ADVANTAGES OF TF-IDF  * It is weighted so it gives importance to lesser frequent words in the dataset as well.  DISADVANTAGE OF TF-IDF  * It does not observe semantic properties of a query or the dataset. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | REFERENCES | | |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | * <https://www.kaggle.com/rmisra/news-category-dataset/metadata> * <https://www.kaggle.com/vikashrajluhaniwal/recommending-news-articles-based-on-read-articles> * <https://stats.stackexchange.com/questions/153069/bag-of-words-for-text-classification-why-not-just-use-word-frequencies-instead><https://machinelearningmastery.com/gentle-introduction-bag-words-model/> |  |