AIM

The main focus of this project is to do topic modelling and sentiment analysis on data related to ‘**Aadhaar- Digital Identity for India’** gathered from various news archive of Hindu, NDTV and Times of India from 2009

BY:

Anirudh Syal

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **S.no** | **Content** |
| 1. | Steps Followed |
| 2. | Functions Defined |
| 3. | List of Documents Uploaded on Github |

**STEPS FOLLOWED:**

**1. PREPARE CORPUS**

The corpus was collected by scrapping live news websites like –“NDTV” and “Times of India” and archives of “The Hindu”.

Approach followed is mentioned below:

1.1 News articles from all three website were searched for keyword “Aadhaar” from archives dating from 2009 till 2018.

1.2 “Selenium” web driver was used to scrape news from “NDTV” and “The Hindu” since both website had dynamic content on the pages.

1.3 Beautiful Soup and requests libraries were used to extract news heading, date, year and body.

* 1. The extracted corpus was stored in form of dataframes, later saved in form of a pickle file.

**2. CLEAN CORPUS**

The corpus collected by web-scrapping has to be pre-processed to remove unwanted data like HTML junk, stop-words, unwanted digits, punctuation characters, etc.

Approach followed is mentioned below:

2.1 All the functions to clean the corpus are applied to rows of dataframe containing text body of news using ‘map’ and ‘apply’ functions only.

2.2 Specific English language patterns like won’t, can’t , I’m etc. have been replaced to their full forms like ‘would not’, ‘cannot’ and ‘I am’ respectively before removing punctuation marks from entire corpus and preventing these English short words to lose their significance.

2.3 Entire corpus is converted to lower case post which html tags, individual digits, special characters like ‘?’ ‘#’ etc are removed from corpus

2.4 Significant symbols like percent or USD are converted to text ‘percent’ and ‘dollar’ respectively.

2.5 Multiple spellings of Aadhaar like aadhar or aaadhaar in corpus are replaced by single term ‘aadhaar’

2.6 Patterns like e-mail or i-phone are converted to email and iphone respectively before removal of special characters like hyphen or underscore.

2.7 Abbreviations like PD – public distribution are replaced by their full forms since they are significant terms for aadhaar topic modelling

2.8 Location of every news is removed from news body corpus since it is at the beginning of each news body.

2.9 The entire corpus is split into individual tokens and regex is used to remove multiple spaces, special words like ‘.com’, ‘.net’ etc.

2.10 The tokens are lemmatized using WordNetLemmatizer after assigning part of speech tag to each token and rejoined

2.11 Other corpus specific stop-word are collected in a file and removed for text corpus

2.12 Bi-grams and Tri-grams are generated and all unigrams that are less significant to their bigrams or trigrams are converted to unigram using underscore for eg: Narendra Modi replaced by narendra\_modi.

2.13 The final corpus after all clean-up is then tokenized using a simple tokenizer function.

**3. TFIDF ANALYSIS**

Term frequency and inverse document frequency (tf-idf), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document (tf), but is often offset by the frequency of the word in the whole corpus (idf), which helps to adjust for the fact that some words appear more frequently in general.

Approach followed is mentioned below:

3.1 Scikit learn tfidf vectorizer function is used to find the tfidf of all the tokens in the corpus.

3.2 More Irrelevant terms(that don’t contribute to theme of article) found in tfidf matrix are again removed from the corpus to get a better set of tokens to be used for topic modelling.

**4. TOPIC MODELLING USING LDA**

We have used an unsupervised machine learning technique, Latent Dirichlet Allocation (LDA), for automatically finding the mixture of similar words together, thus forming the topic or theme. This forms an unsupervised problem since we do not know the labels/classes/categories of the data and aim to find the groups or the clusters within the population of tokens.

Approach followed is mentioned below:

4.1 We have used feature\_extraction of scikit learn to perform the LDA modelling.

4.2 pyLDAvis is used for visualization of the LDA topics.

**5. DOCUMENT CLUSTERING USING K-MEANS**

K-means clustering is a type of unsupervised learning, which is used to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

Approach followed is mentioned below:

5.1 We have used Scikit-learn MiniBatchKMeans package of python to find the k document clusters.

**6. SENTIMENT ANALYSIS USING TEXTBLOB**

Sentiment analysis is the process of categorizing opinions expressed in a piece of text. A basic form of such analysis would be to predict whether the opinion about something is positive or negative (polarity).

Approach followed is mentioned below:

6.1 We have used TextBlob package of python to calculate polarity and subjectivity of text.

**FUNCTIONS DEFINED:**

**LIST OF DOCUMENTS UPLOADED:**

1. Source Code for Web – Scrapping NDTV, The Hindu, Times of India
2. Source code for text cleaning, TFIDF, LDA, KMEANS
3. Supporting files as input:
4. location\_stop\_words.txt: List of locations in body of news
5. more\_stop\_words.txt: List of stop words
6. bigrams\_1.xlsx: List of bigrams
7. bigrams\_2.xlsx: List of bigrams continued
8. trigrams\_1.xlsx: List of trigrams
9. trigrams\_2.xlsx: List of trigrams continued
10. hindu\_non-relevant\_entries.xlsx: List of stop words removed after observing TFIDF/topic modelling results
11. Output files:
12. Toi\_df.pkl: Pickle file for dataframe created post scrapping times of India
13. Hindu\_df.pkl: Pickle file for dataframe created post scrapping times of India
14. Final\_clean\_toi\_corpus.pkl: Pickle file of TOI dataframe after cleaning complete corpus
15. Final\_clean\_hindu\_corpus.pkl: Pickle file of Hindu dataframe after cleaning complete corpus
16. Hindu\_tfidf\_001.xlsx: Matrix of TFIDF of various tokens
17. Pyldadavis\_x.html: HTML files for LDA plot for x number of topics