PROJECT Y

BY:

PATHFINDERS

JUNE – AUGUST 2021

1. ANUVA GOYAL
2. SHABD SWAROOP
3. SUMATI

PROBLEM STATEMENT

This project consists of two components -

Component 1 - Web Scraping module

You are required to scrape two weeks (Feb 1 , 2021 to Feb 14 , 2021) of web news files from The Hindu archive site : https://www.thehindu.com/archive/web/2021/02/ . As per the Robots.txt at Hindu website(https://www.thehindu.com/robots.txt) , the archives section is NOT in disallow , thus it implies it is within legal norms to scrape the archive section.

From each of the page , you need to scrape out the web content and create an independent JSON file that will capture all the text details as values and store them in a key named "text" within that file.

Component 2 - Topic Model

Once you've scraped the two weeks' worth of data , the idea is to perform Topic Model on this scraped data set. You can keep 90% of the web scraped data in training of the model and 10% as a holdout set on which you can assign topics based on the trained model.

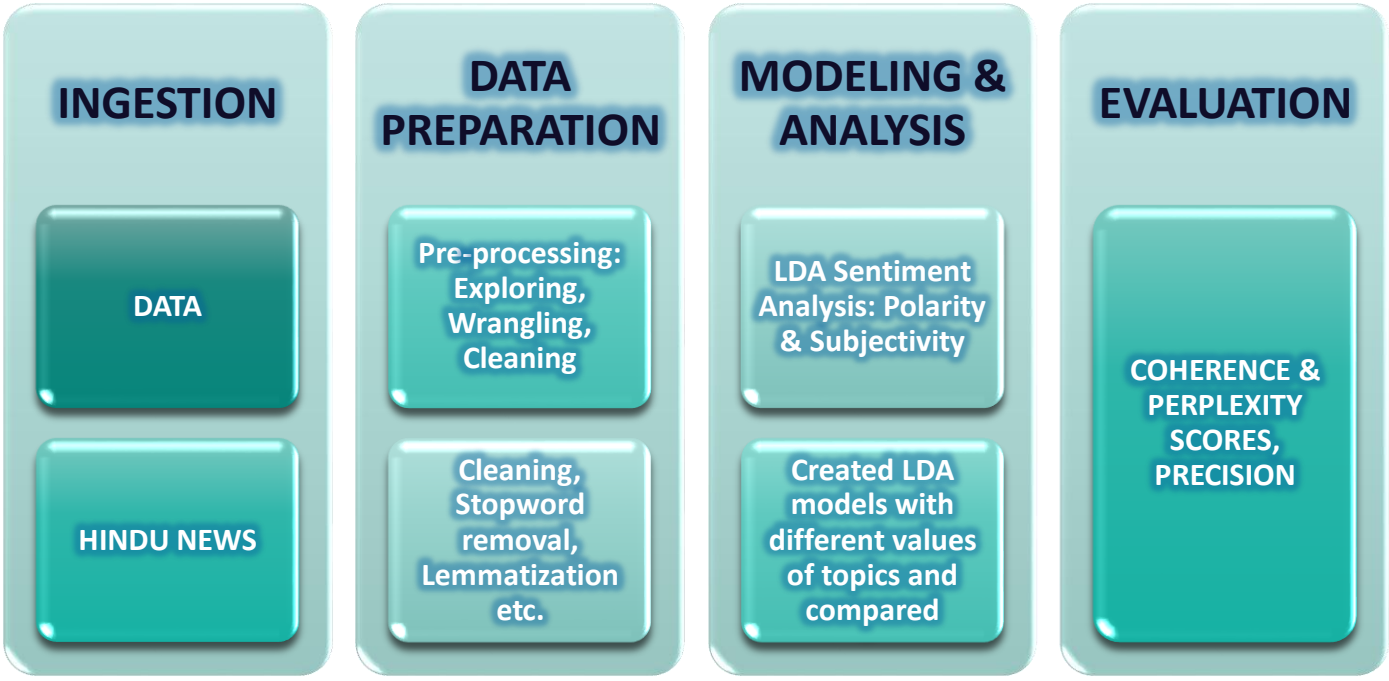
Find out the optimal topics number based on Topic Coherence score along with manual review of Topic - words quality. Iterate until you can get the diverse and best topics with words that closely define a theme/topic

For each of the Topic that have been identified - give the Topic Name based on IAB Taxonomy attached. IAB Taxonomy is a standard followed across web to provide web page content a consistent Topic naming convention.

Based on the above Topic names assignment , save the model, and then assign the topics to 10% test set. Try doing manual review for few sets of links i.e., whether the assigned topics make sense to what URL web page is about ? Come up with metrics around Accuracy or any other that you feel relevant to measure the quality of Topic Model

1. METHOD DESCRIPTION

Following flowchart describes the methodology adopted for tackling the problem statement. Each module in the flowchart will be further decomposed in subsequent sections of the report.



2. MODULE DESCRIPTION

2.1. WEB SCRAPING(INGESTION/DATA)

Before working on the scraper or writing code for scraper, we analysed the weblinks first which were to be scrapped. Following were some results:

- A total of 5565 links of Hindu news articles was required to be scrapped.
- Since each webpage contained some irrelevant data which were not required for scaping, the html tags which contained useful content was required to be analysed.
- HTML tags that contained useful content were:

h1 : Title of the News article.
h2 : Subtitle of the article.
p : Body of the article.

Now since there was a total of 5565 links, it was difficult to scrape them in one go as we got blocked by the site after scraping almost 2000 files. We tried to use proxy IPs as well for scraping but it didn't work as a single IP took almost 25-30 min to establish connection and scrape a link.

For scraping we created two separate lists of weblinks for 7 days each and then scraped them separately.

2872 weblinks : Took 1 hour and 45 minutes to scrape.
2683 weblinks : Took 1 hour and 37 minutes to scrape.

For detecting the efficiency of our scraper, we did an outlier detection as to how many files contained number of words that were below the lower threshold. Following were its results:

Upper bound = 1786.00000
Lower bound = -897.2000
Files below lower bound = 0
Files above upper bound = 11

These 11 files above the upper threshold were removed and a total text of 5554 links were used for further preprocessing.

2.2 DATA PREPROCESSING

Raw data scraped from the webpages was not feasible for analysis and topic modeling as the data must have contained extra white spaces and contained those

extra words which could have disturbed topic building. Hence data preprocessing is a key methodology that is needed to be adapted.

- Data Cleansing : In this particular process all extra white spaces, numbers (which is not necessary in topic building as it will give no-sense), and other html tags such as /w,/s etc.
- Stop Words Removal: It is basically removal of set of commonly used words in any language. In this we had increased the set of stopwords so as to create a cleaner data set for topic modeling. The set of stopwords contained all conjunctions, articles (a/an/the), and other unnecessary words which we came across while analysing the data set that were not required in topic modeling. The set was made with the help of frequency distribution plot for each word. It helped in extending the stopwords set.

```
['crore', 'first', 'said', 'work', 'also', 'case', 'would', 'take', 'time',
 'last', 'year', 'three', 'make', 'nthe', 'need', 'even', 'issu', 'well', '
come', 'made', 'come', 'howev', 'work', 'februari', 'like']
```

- Tokenization : In this process, text is splitted into smaller pieces of words known as tokens. These tokens help in understanding the context or developing the model for the NLP. It is basically creating a vocabulary for machine learning model.

- Stemming : Stemming basically breaks the words to its root word that contains only the suffix and prefix of the word. It is important as all additional forms of word are reduced to their root word. For ex: Words like “minister”, “ministry”, “ministers” are all reduced to their root word “minist”. It is important as it may happen that each form of word may take a separate topic in topic modelling.

- Lemmatization : In Lemmatization as well the words are broken into its root word called ‘lemma’. Although both stemming and lemmatization perform same task there is a difference between them as in lemmatization the algorithm chops the word in a way that makes sense in the language. However, stemming just chops out the word without knowing the actual meaning.

- Bigram and Trigrams: Bigram and trigram simply makes a sequence of either two or three words, respectively.

2.3 SPLITTING DATA TO TRAIN AND TEST SET

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Training dataset trains the algorithm for machine learning model whereas test dataset acts as an input so the ML algorithm can make predictions. The model is first to fit on the available data with known inputs and outputs. It is then run to make predictions on the rest of the data subset to learn from it. This can be used to make predictions on future data sets where the expected input and output values are non-existent.

In this 90% of the dataset is used for training the model and remaining 10% of data is used for testing the model. 4998 datasets were under training dataset whereas 556 datasets were under test dataset. The scikit-learn Machine learning library was used as it specifically contains train_test_split() function.

2.4 TOPIC MODELING

Topic Modeling is basically a technique that provides method for organizing, understanding, searching, and summarizing large chunk of data by discovering hidden themes and classifying them. We have used LDA (Latent Dirichlet Allocation) Model in our project.

LDA model is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words. LDA assumes that every chunk of text we feed into it will contain words that are somehow related. Therefore, choosing the right corpus of data is crucial. It also assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution.

Firstly, we used gensim LDA model on the train dataset to train the module. Before building the model we first did the hyperparameter tuning by hit and trial method to obtain the best possible hyperparameters. Following were its results:

Num_topics	random_state	iterations	passes	alpha	chunksize	Coherence Score	Perplexity
25	42	50	20	auto	1000	0.566	-7.166
20	42	100	150	auto	1000	0.587	-7.206
15	42	50	100	auto	1000	0.595	-7.202
15	42	50	120	auto	1000	0.595	-7.202
15	342	50	120	auto	1000	0.597	-7.222
14	563	50	120	auto	1000	0.590	-7.226
14	342	50	120	auto	1000	0.610	-7.225

As seen the hyperparameters with Number of topics = 14 and random state =342 , was most suitable as it had highest coherence score among all. Hence the same hyperparameters were used and lowest perplexity score. However, we tried another model of LDA mallet as well but the coherence score were not as good as required, so we proceeded with the same LDA model itself.

Further, after training the model, we tried to visualize the topics prepared by the model and since LDA model doesn't give a topic name to topics and it is for us humans to interpret them, we named the topics based on the IAB taxonomy file provided to us. Following were the 14 topics interpreted by us:

0. Medical Health
1. Political Issues
2. Sports – Cricket
3. News and Politics – Elections
4. New and Politics – Law
5. News and Politics – National News
6. Business and Finance
7. Region/State
8. Technology and Computing
9. Education
10. Region/State
11. Movies
12. International News
13. National News - Projects

Since Training part of the module was done, we further went on to test our model on the test dataset. Using LDA model, visualization of topic distribution on test set was done and coherence score was calculated which came out to be 0.50897. For checking

the topic distribution on test set we did a reverse processing on almost 100 files of the test set and obtained following results:

TOTAL FILES : 100

TRUE POSITIVES : 43

FALSE POSITIVES : 57

PRECISION : 0.43

CONCLUSION

Following is the gist of the entire report with the results obtained:

WEB SCRAPING: 1) Total 5565 links scraped.
2) Proxy rotation failed.
3) Time taken : 3 hours and 22 minutes

DATA PREPROCESSING: 1) Performed all the steps of data preprocessing.
2) Increased the size of stopwords corpus.
3) Removed the 11 outlier files.
4) No problems/blockers faced as such.

SPLITTING DATA TO TRAIN AND TEST: 1) 4998 datasets kept in Train, 556 in test.
2) No problems/blockers faced as such.

TOPIC MODELING: 1) LDA MODEL USED.
2) Hyperparameter tuning done by Hit and trial method.

```
(corpus=corpus,id2word=id2word,num_topics=14,random_state= 342,iterations
= 50,chunksize=1000,passes=120,alpha='auto',per_word_topics=True)
```

3) Coherence Score of 0.61051 on train set.
4) Coherence Score of 0.50897 on test set.
5) Precision of 0.43 on 100 files of test data set.
6) No Problems/blockers faced as such.