News Article Analysis using Data Mining and Natural Language Processing

Presented by
Group 12

Dharmang Solanki (013723931)
Kunika Mittal (014529086)
Prachal Patel (014529216)

Under supervision of
Professor Gheorghi Guzun

Github Repository:
https://github.com/dharmang007/project255

1. Introduction

The explosion in the amount of news and journalistic content being generated across the globe, coupled with extended and instantaneous access to information through online media, makes it difficult and time-consuming to monitor news developments and opinion formation in real-time. There is an increasing need for tools that can pre-process, analyze, and classify raw text to extract interpretable content; specifically, identifying topics and content-driven groupings of articles.

News articles on websites, blogs, or newspapers heavily rely on text mining and data mining techniques in order to improve their customer service and search performance. Data mining can be leveraged to extract important information that can be further used based on specific needs. For instance, clustering and classification can be used to automate tag generation in blog sites and websites. Tags are very important for blogs to improve search performance and find relevant topics easily.

In this project, we aim to implement various clustering techniques, like k-means, and topic modelling techniques Latent Dirichlet Allocation like to automate the tag generation based on the results. Analysis and comparison of various techniques are important to check which method works best in the domain of Natural Language Processing. Moreover, based on the error, content, and nature of raw data, we have used different data cleaning techniques to get proper input data.

2. Design and Implementation

2.1 Algorithms Implemented

The algorithms implemented for tag generation in the project are K-Means Clustering and Latent Dirichlet Allocation (LDA).

Latent Dirichlet Allocation

LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics.

K-Means

K- means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points

and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The K-Means clustering algorithm has been used to cluster the news articles with a similar topic or content context. The extraction of nouns and pronouns is done while naming the clusters. Text data being unpredictable we cannot randomly assume the cluster size. We tested the K-means using **Davies–Bouldin** index and Sum of Square for error.

MiniBatchKMeans method was compared with normal Kmeans. Due to the large dataset and for getting the results faster we used MiniBatchKMeans which uses less computation time with trade-off in accuracy. This variation of K-means uses mini-batches to reduce the computation time, while still attempting to optimise the same objective function.

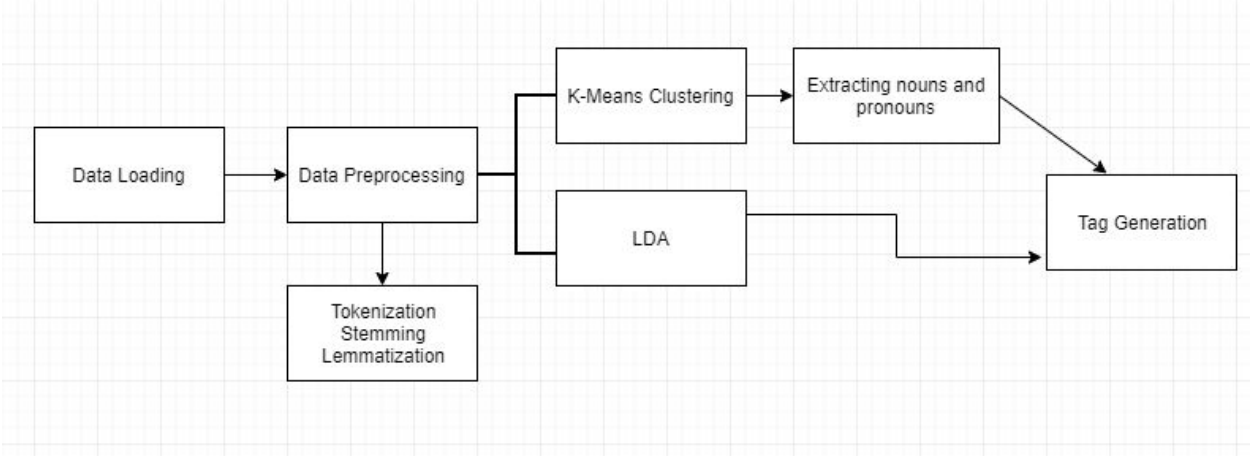
2.2 Technologies and Tools Used

The following tools and technologies have been used extensively for the execution of the project.

- Jupyter Notebook
- Python 3.8
- Libraries Used: sklearn, nltk, spacy

2.3 Data Flow

The data is first loaded in data frames and then preprocessed using different feature selection methodologies which will be discussed in detail later in the report. The filtered data goes through different tag generation algorithms to obtain the final tags for the news articles.



Data Flow Diagram

3. Proof of Evaluation/ Experiments

3.1 Data Set

The dataset used in this project is “All The News”, which consists of 15 news publishers in the US. The article’s content is in the English language and articles are dated from the year 2013 to 2018.

The publications include the New York Times, Breitbart, CNN, Business Insider, the Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, the Guardian, NPR, Reuters, Vox, and the Washington Post.

The data has been chopped into three different CSV files because of the large amount of data in them.

articles1.csv - 50,000 news articles (Articles 1-50,000)
articles2.csv - 49,999 news articles (Articles 50,001-100,00)
articles3.csv - Articles 100,001+

Link: https://www.kaggle.com/snapcrack/all-the-news

3.2 Data Preprocessing

The data has been preprocessed using the following text normalization methods.

- **Stemming**: Stemming algorithms are typically rule-based. With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or smaller form of the word.
- **Removing Noisy tokens:** Noisy data can add many problems like increasing dimensions and giving incorrect results. In our text data there are many forms of noise:
    - Non English words: Non-English words add no value if we just want to cluster based on English language.

Fig. Some non English words found in our data

- Stop words: Words like ‘a’, ‘an’, ‘the’, and many those english articles can unnecessarily increase the computation and add very little value to the clustering process.
- Removing alphanumeric tokens: In understanding the core abstract of a text, alphanumeric tokens will not add any value to the clustering processing.

Fig. Some of the Alphanumeric Noise in our data

- Remove the words with length 1: This is the assumption that single letter tokens will serve no purpose in adding the meaning of a document or in the clustering process.

- **Tokenization**: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation.

Fig. Result of tokenization

- **Lemmatization**: Lemmatization, unlike Stemming, reduces the inflected words properly ensuring that the root word belongs to the language. In Lemmatization the root word is called Lemma. A lemma (plural lemmas or lemmata) is the canonical form, dictionary form, or citation form of a set of words.
  E.g.

am, are, is ⇒ be

car, cars, car's, cars' ⇒ car

This preprocessing helps reduce the semantic duplicates and adjust the frequencies of similar words.

- **Count Vectorization**: Creating vectors that have a dimensionality equal to the size of our vocabulary, and if the text data features that vocab word, we will put a one in that dimension. Every time we encounter that word again, we will increase the count, leaving 0s everywhere we did not find the word even once.

- **TF - IDF Vectorization:** A Tf-Idf (term frequency-inverse document frequency) vectorizer, gives a value for each word in each article weighted by that word frequency in the whole corpus. The inverse-document frequency is a denominator derived from the word's frequency in the entire dataset.

The data after text normalization is fed to both of the following algorithms.

K - Means Clustering

The k-means problem is to find cluster centers that minimize the intra-class variance, i.e. the sum of squared distances from each data point being clustered to its cluster center (the center that is closest to it).

The input given to k means was reduced dimensions using Truncated SVD with 100 components.

3.3 Statistical Comparisons and results

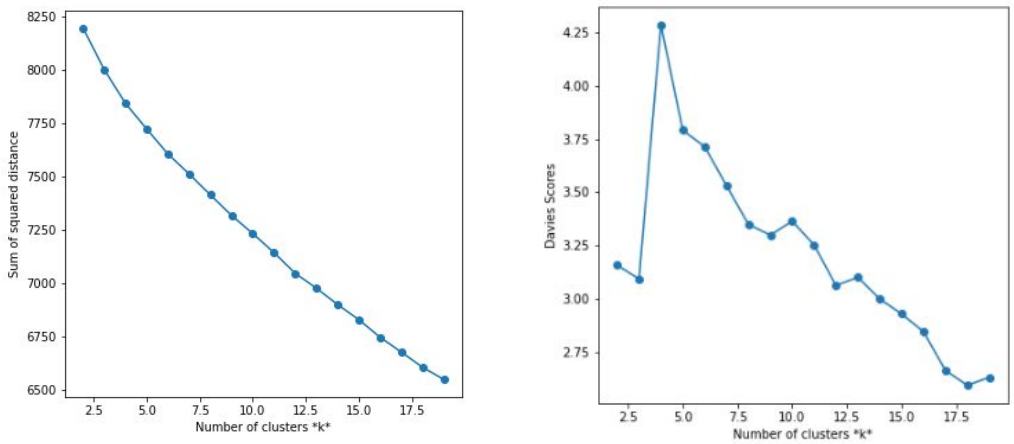


Fig. Sum of Squared error(left) and Davies Bouldin Score(right) for Number of Clusters using simple K means

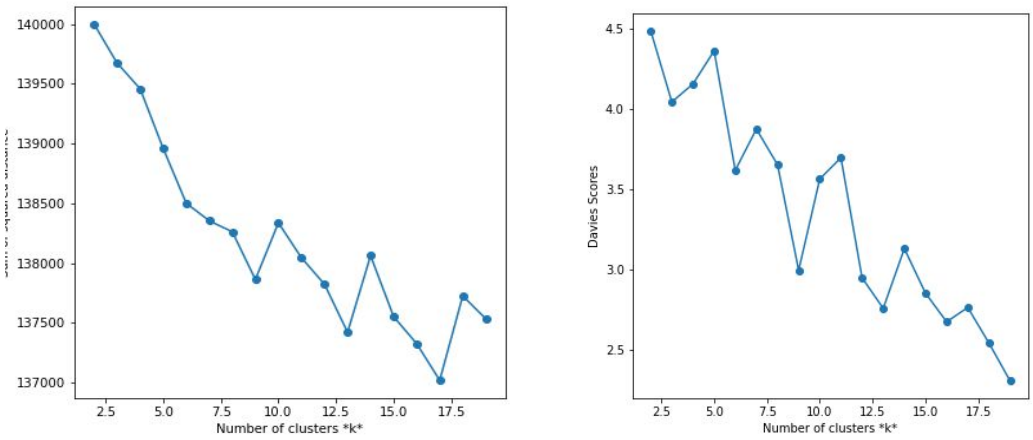


Fig. Davies-Bouldin Score vs Number of Cluster Using Mini Batch Kmeans

Based on the above results you can see that 20 gave the least Davies-Bouldin Score of 2.55.

Clustering Method	**Optimized Number of clusters**	**Davies Bouldin Score**
K-means	18	2.2333
MiniBatch Kmeans	20	2.5644

3.4 Post Processing and Analysis

Tag Generation

Giving the names to the clusters is the final step of this process. There are many ways we can assign names to a specific cluster. Giving a single name to a cluster is a very restrictive way. We fetch the best possible 10 tags based on which the user can name the tag.

Based on the below results, we can analyze that documents which were clusters had similar abstract and had similar frequent words.

Cluster 0 had the words like ‘disappearance ’, ‘mystery’, ‘jury’, ‘trail’ which gives the idea that we can name that cluster as “Crime”.

Cluster 16: Have too many words related to Donald Trump, which represents that we can name that cluster as Donald trump.

Cluster

```
Cluster ID 0:
['convicti', 'patz', 'disappearance', 'mystery', 'jury', 'trial', 'manhattan', 'tuesday', 'year', 'foota']
Cluster ID 1:
['kenya', 'blackout', 'monkey', 'elect', 'knee', 'infrastructure', 'hour', 'tuesday', 'footag', 'foota']
Cluster ID 2:
['application', 'gawker', 'ability', 'launch', 'concern', 'crime', 'apple', 'time', 'week', 'football']
Cluster ID 3:
['deba', 'houston', 'season', 'gop', 'night', 'thursday', 'donald', 'trump', 'forbi', 'forbiddin']
Cluster ID 4:
['change', 'penny', 'mike', 'vice', 'climate', 'president', 'trump', 'indiana', 'gov', 'donald']
Cluster ID 5:
['year', 'cnn', 'people', 'day', 'time', 'new', 'police', 'city', 'woman', 'tuesday']
Cluster ID 6:
['ashe', 'semifinal', 'serena', 'arthur', 'stadium', 'williams', 'sister', 'career', 'foodporn', 'fontana']
Cluster ID 7:
['trump', 'donald', 'president', 'campaign', 'washington', 'republican', 'nominee', 'cnn', 'house', 'white']
Cluster ID 8:
['breslaw', 'anna', 'alcohol', 'drink', 'commitment', 'writer', 'manhattan', 'foot', 'footcare', 'footballer']
Cluster ID 9:
['news', 'breitbart', 'fox', 'host', 'channel', 'siriusxm', 'daily', 'trump', 'donald', 'am']
Cluster ID 10:
['photo', 'hardship', 'siena', 'indication', 'beauty', 'answer', 'international', 'award', 'life', 'month']
Cluster ID 11:
['spicer', 'press', 'scrutiny', 'sean', 'briefing', 'secretary', 'white', 'house', 'thursday', 'trump']
Cluster ID 12:
['state', 'united', 'president', 'nation', 'trump', 'cnn', 'donald', 'secretary', 'washington', 'year']
Cluster ID 13:
['ad', 'extension', 'fol', 'story', 'food', 'foot', 'footcare', 'footballer', 'football', 'footbal']
Cluster ID 14:
['fedna', 'hermosante', 'chantal', 'creek', 'haiti', 'wind', 'hurricane', 'bank', 'wall', 'home']
Cluster ID 15:
['week', 'trump', 'cnn', 'president', 'donald', 'time', 'day', 'election', 'house', 'new']
Cluster ID 16:
['obama', 'president', 'barack', 'administration', 'trump', 'washington', 'house', 'white', 'donald', 'cnn']
Cluster ID 17:
['softbank', 'telecom', 'conglomerate', 'internet', 'united', 'tuesday', 'state', 'donald', 'trump', 'foodie']
Cluster ID 18:
['clinton', 'hillary', 'donald', 'trump', 'campaign', 'sander', 'bernie', 'email', 'nominee', 'candidate']
Cluster ID 19:
['annibale', 'gasparis', 'asteroid', 'italian', 'jupiter', 'astronomer', 'mar', 'march', 'de', 'footbal']
```

Latent Dirichlet Allocation

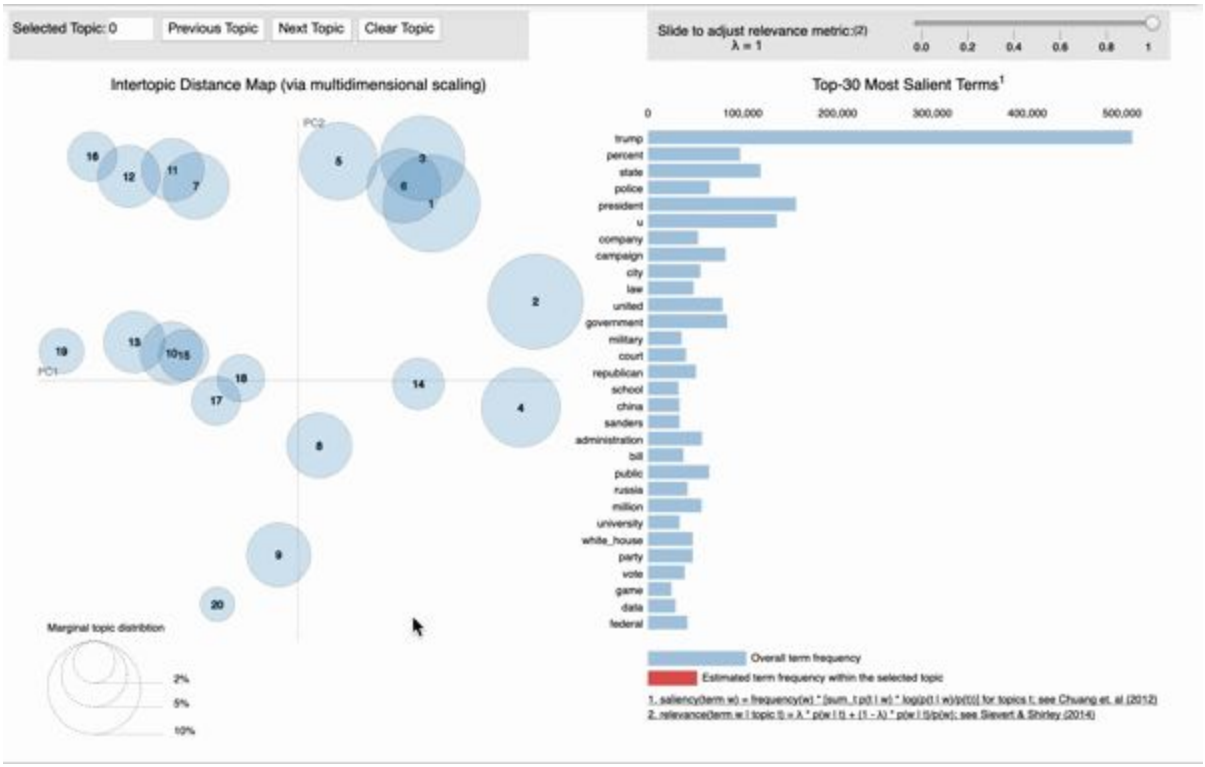
LDA uses the bag-of-words feature representation to represent a document. It makes sense, because, if I take a document, jumble the words and give it to you, you still can guess what sort of topics are discussed in the document.

It is like PCA but it mainly focuses on maximizing the distinctiveness from the known classes. The primary goal of LDA is to provide such features from the data that are more distinct features not the most representative features from the data. This will give us the topics from the data and provide the tags in our case from the articles.

We had a pool of 1,42,570 news articles which is a very large amount of data to process for LDA. So, after applying basic preprocessing techniques, we decided to convert the articles into bigram phrases that can be helpful for the topics to correlate to each other.

The data however was very large so it took about 38 minutes in total to perform LDA using gensim library which is an open source library for topic modeling and Natural language processing. The result was 20 topics from the model based on their similarity of topics discussed.

Results:



4. Discussion and Conclusions

Based on the results and type of the tags names generated in K-means and LDA, more relevant names were found in the LDA method for more flexible and reliable results. The bottlenecks we faced were the memory management and execution which further improved using cuda parallel computing. In terms of accuracy we saw a great impact of text preprocessing and can further be improved by leveraging text summarizer and semantic analysis of each sentence in the document.

The results obtained from these methods can be reliably used for many applications like to find trending topics in a given year, find which news publications public which kind of topics in a given period.

5. Task Distribution

Tasks have been equally divided and executed by all the group members.
Code: The coding for the implementation of the project was taken care of by the following member.

- Data Study and Preprocessing : Kunika Mittal
- Preprocessing, Clustering and Postprocessing Tag Generation: Dharmang Solanki
- LDA, Topic modeling and related Preprocessing: Prachal Patel

Report: The report was collaboratively written by all the team members.

6. References

- K-means Clustering
  https://en.wikipedia.org/wiki/K-means_clustering
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Enhancing Text Clustering Using Concept-based Mining Model
  Shady Shehata, Fakhri Karray, Mohamed Kamel
- https://www.kaggle.com/nmud19/topic-modelling-doc2vec-kmeans-nmf
- https://medium.com/datadriveninvestor/automatic-topic-labeling-in-2018-history-and-trends-29c128cec17
- An Evaluatoion on feature selection for text clustering
  Tao Liu, Shengping Liu, Zheng Chen
- Moe R.E. (2014) Clustering in a News Corpus. In: Sojka P., Horák A., Kopeček I., Pala K. (eds) Text, Speech, and Dialogue. TSD 2014. Lecture Notes in Computer Science, vol 8655. Springer, Cham
- Austin L.E Kraus, News Articles Clustering Using Unsupervised Learning, Medium article, August 2, 2019
- Andrew Thompson, "All the News",https://www.kaggle.com/snapcrack/all-the-news/
- Evaluation Metrics, https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation
- https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
- LDA topic modeling : https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0