Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

# Topic Modeling in Natural Language Processing

Subhashini Gupta
Department of computer engineering
K. J. Somaiya college of engineering
Mumbai, India

Prof. Grishma Sharma
Department of computer engineering
K. J. Somaiya college of engineering
Mumbai, India

***Abstract*— Every day large quantities of data are collected. As more information is available, the access to what we are seeking gets challenging. We, therefore, require processes and techniques for organizing, searching, and understanding massive amounts of information. The task of topic modeling is to analyze the whole document to learn the meaningful pattern that exists in the document. It is a supervised strategy used to identify and monitor words in clusters of texts (known as the "topics"). Through the use of topic analysis models, companies can load tasks on machines rather than burden employees with too much data. In this paper, we have used Word embedding for Topic Modelling to learn the meaningful pattern of words, and k-means clustering is used to group the words that belong to one group. In this paper, we have created the nine clusters of words from the headline dataset. One of the applications of topic modeling i.e sentiment analysis using the VADER algorithm is also demonstrated in this paper.**

***Keywords—Topic Modeling, Word Embedding, K-means, Sentiment Analysis,VADER.***

## I. INTRODUCTION

Topic modeling is a simple way to capture the sense of what a document or a collection of documents is about. Documents are any coherent collection of words, which could be as short as a tweet or as long as an encyclopedia. Topic modeling, in a summary, is a text-mining methodology for identifying topics in documents. Topic modeling is frequently used as a first step in analyzing textual data to acquire a sense of the text's content. This is especially relevant when abstracts/summaries are unavailable and the text is too voluminous to manually examine within the time span provided.

Topic modeling is a form of unsupervised machine learning, as opposed to supervised learning algorithms. This indicates that in order to train the model, we do not need to provide labels (that is, subject names matching to each document) during training. This not only helps us discover interesting topics that might exist but also reduces the manual effort spent in labeling texts. On the flip side, it can be a lot more challenging to evaluate the output of a topic model.

Words with comparable meanings can be linked together using topic modeling and discriminate between the usage of words with different meanings by using a cluster of phrases that regularly occur together. The goal of the analytics industry is to extract "Information" from data. It's challenging to get relevant and required information with the increasing volume of data in recent years, most of it is unstructured. However, technology has created some strong tools for mining through data and retrieving the information that we want.

The topic modeling technique can be used in a different application of natural language processing. So in this special topic seminar, I'm proposing the different analysis and comparison methods of topic modeling used in natural language processing.

## II. LITERATURE SURVEY

In paper[1] Word Embedding for topic modeling. They compared and assessed two distinct frameworks for unsupervised topic modeling of the CompWHoB Corpus, essentially the political-linguistic dataset, in this study. The first approach leverages the Latent Dirichlet Allocation technique, while the second framework uses the Word2Vec methodology to learn word vector representations that will later be utilized for topic modeling. The linguistic preprocessing stage was given specific attention in order to increase the quality of textual data. NLTK library is used for word tokenization, POS-tagging, and To refine the data, lemmatization was used. In the first experiment, LDA is utilized, which is a generative probabilistic model for inferring latent topics from a collection of documents. The LDA is trained on training corpus by the use of the Gensim library following the pre-processing stage Word2Vec is utilized in the second experiment. Based on the hypothesis that words that appear in similar settings have similar meanings. The Word2Vec model can be used to learn word embeddings, which are vector representations of words. The CBOW technique was used to train the model because it is better suitable for larger datasets.

In paper [2] Topic Modeling in Embedding Spaces, An embedded topic model is employed for the generative document model which combines conventional topic with word embedding. Even with huge vocabularies containing unusual words and stop words, the ETM finds interpretable topics. The ETM probably employs an embedding word matrix, a representation of the vocabulary in a smaller space. They can either employ pre-fitted embeddings or learn them in practice as part of their entire strategy. When the ETM learns embedding as part of the fitting technique, it simultaneously discovers topics and integration space. Like LDA, the ETM is a generative model of probability, in which every document is a mix of subjects, and a particular topic is given to each word observed. Each sentence is represented by an integration; each subject is a point within the space of the integration, and the distribution of the subject over terms is proportionate to the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

internal product of the embedding of the subject and the embedding of every term. One of the objectives in the ETM was to integrate word similarity into the topic model. The ETM model is the combination of LDA and word embeddings. They have employed the variation of word integration of a continuous bag of words (CBOW). The idea underlying consistency is that a well-organized theme emphasizes terms that often appear in similar writing. In other words, a high level of mutual information should most likely be found in a coherent topic. Topic models with greater cohesion are more understandable. The ETM learns a corpus in a specific embedding space when used previously fitted embeddings. This is particularly useful when the insertion comprises terms not found in the corpus. The ETM can determine the size of the words in the subjects.ETM gives better predictions and topics than LDA and the Neural Document Model. Both consistent language patterns and the accurate distribution of words should be provided by a good document model, which will require both predictability and topical interpretability to measure performance.

In paper[3] Semantic Augmented Topic Model over Short Text, The shorter text was proposed for a latent semantic augmented bi-term topic model(LS-BTM). The popularities of mobile equipment make short texts an important element of the information carrier. For many natural language tasks like detecting emerging topics, content analysis, question answering, sentimental analysis, automatic summary systems, recommendation systems, etc. discovering the prospective topics from the short text are significant. However, the lack of short texts results in insufficient information within the context, and it is impossible to analyze the diversity of language expression by general approaches. The classical subject models like LDA and PLSA were quite successful in a long text. However, because of the lack of word patterns, they operate badly on short texts. Topic models were explored for numerous years and utilized successfully in numerous areas. In short text topic models, however, there are significant challenges. In brief texts, for instance, a tweet message comprises a maximum of 140 characters, a maximum of more than 90% is less than 10 words and nearly half of the text-only includes one or two words. Secondly, short messages are always flexibly expressed in a language that leads to ambiguity.

## III. PROPOSED METHODOLOGY

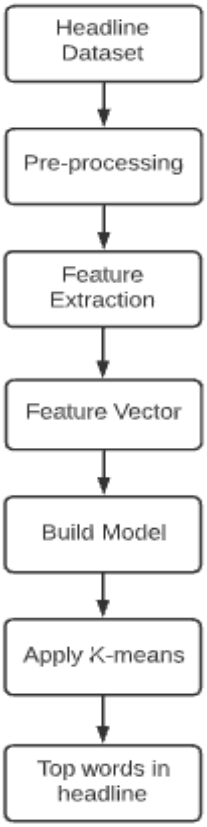


Fig. 1 Proposed Methodology

### A. *Word Embedding*

Word Embeddings are numerical representations of texts that have been translated. The same text may be represented numerically in multiple ways. Individual words in word embeddings are represented as real-valued vectors in a vector space. It can recognize a word's context in a document, as well as semantic and grammatical similarities, relationships with other words, and so on. It is a method of providing a dense representation of words. Due to its training speed and performance, Word2Vec is one of the most common strategies for learning word embeddings. The word2vec tool generates vector representations of words by employing the continuous bag-of-words (CBOW) and skip-gram models. These representations can then be employed in a variety of natural language processing applications.

### B. *How* does word2vec work

Word vectors are generated from a text corpus using the word2vec tool. It learns the vector representation of words after creating a vocabulary from the training text input. The generated word vector file has the advantage of being usable in a wide range of (NLP)natural language processing and (ML)machine learning applications.

Finding the closest terms to a user-specified term is a simple method for investigating the learned representations. This is what the distance tool is for. If we type in 'Iraq,' for example, the distances between them and 'Iraq' will be displayed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

It has recently been demonstrated that word vectors capture numerous linguistic regularities, such as vector operations. **vector('Ronaldo') - vector('Soccer') + vector('Tennis')** results in a vector that is very close to [('Nadal', 0.6514425277709961), ('Safin', 0.6181677579879761), ('Federer', 0.6156208515167236)], and **vector('king') - vector('man') + vector('woman')** is close to [('queen', 0.7118192911148071), ('monarch', 0.6189674139022827), ('princess', 0.5902431011199951)].

### *C. Continuous Bag of words*

The way CBOW works is that it predicts the probability of a word given a context. A single word or several words could be a context.

An input layer, a hidden layer, and an output layer comprise a three-layer neural network.

The flow is as follows:

1. The target layer and the input layer, both single-hot encoded of size [1 X V].
2. Two weights are available. One is between the input and the layer that has been hidden, and the second between the layer that has been hidden and the output.
3. The size of the input-hidden layer matrix is [V X N], and the size of the hidden-output layer matrix is [N X V].

Where N denotes the number of dimensions in which we want our word to be represented. It is an arbitrary hyper-parameter for a Neural Network. In addition, N represents the number of neurons in the hidden layer.

4. The input is multiplied by the input-hidden weights and called hidden activation.
5. The output is calculated after multiplying the hidden input by the hidden-output weights.
6. It determines and re-adjusts the differences between the output and the target error.
7. As a vector representation of the word, the weight between the hidden layer and the output layer will be assumed.

### *D. Skip – Gram model*

It simply flips the architecture of CBOW on its head. Skip-gram is a technique for predicting the context of a word. Skip-gram input vectors are comparable to a 1-context CBOW. The differences will be found in the target variable. Two single-hot coded target variables are available and two corresponding outputs because both sides have a context window of one. For each target variables, two independent errors are created, resulting in an element-by-element error vector that is propagated back to updating weights. Element by element. After training, the input and hidden layers weights are used to create a word vector representation. The objective or loss function is the same as in the CBOW model.

### *E. K means Clustering Algorithm*

Clustering is one of the most often exploratory data analysis approaches for gaining a sense of the data's structure. The k-means algorithm identifies clusters in the dataset so that data points in the same cluster are quite identical while data points in other clusters are quite dissimilar.

K-means Algorithm segment the dataset into k pre-defined clusters(subgroups) with every data point belonging to a single category.

K-.means algorithm is given below:

1) the first step is to select k no. of the cluster.
2) As centroids, pick k random no. of points from the given data.
3) Allocate all points to the closest cluster centroid.
4) Determine again the centroids of the newly generated clusters.
5) Repeat steps 3 and 4.

To stop the K-means algorithm, three conditions can be used:
1.Newly created cluster centers do not alter.
2.The points stay in the same group.
3.Iterations shall reach the maximum number.

## IV. APPLICATION OF TOPIC MODELING IN SENTIMENT ANALYSIS USING VADER

### *A. VADER Algorithm*

Valence Aware Dictionary for Sentiment Reasoning(VADER) is responsible for the analysis of sentiment analysis of the text that takes both positive or negative polarity and the intensity of words.

A lexicon that maps lexical items to emotional intensities, which are referred to as sentiments, is used to analyze the sentiment. We may calculate the sentiment of a text by summing the intensity of every word in the text. Words such as love, enjoyment, happiness, and like all mean good emotions, for instance. It also acknowledges the value of capitalization and punctuation, as in "ENJOY."

| **Word** | **Sentiment** |
|---|---|
| good | 0.5 |
| great | 0.8 |
| terrible | -0.8 |
| alright | 0.1 |

Fig. 2 lexicon dictionary

### *B. Sentiment Analysis*

Sentiment analysis has become a significant research subject in information retrieval and web data analysis as the World Wide Web has grown in popularity and acceptability. Because of the massive amount of user-generated information on blogs, forums, social media, and other platforms. Because it deals with the extraction of thoughts and attitudes, researchers from academia and industry have been drawn to sentiment analysis.

Sentiment Analysis can be done in 2 ways:-
Either by using just the words(i.e. topic modeling). As topic modeling is unsupervised we don't need to labeled data. Words

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

like 'love,' 'enjoy,' 'happy,' and 'like', for example, all communicate a positive sentiment. There is a corpus of words and their sentiments available. Based on the words in each topic, we may compute a sentiment score. This is the approach if we plan to use VADER's sentiment analyzer class, LDA for sentiment extraction. The other option is Text Classification, in which we must tag a corpus as positive, negative, or neutral and then use it to train a sentiment classifier using word features. Topic classification is a supervised learning technique, which means it requires training before it can be used to analyse texts automatically.

*C. Process flow for sentiment analysis using VADER*

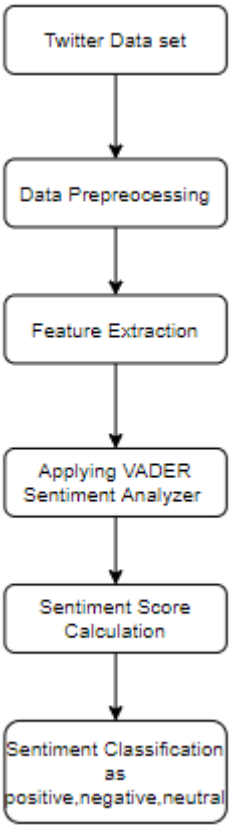


Fig. 3 Sentiment classification

1. Data Pre-preocessing:
In order to transform data in a useable format for future processing, data processing is performed. In this we first remove the duplicate tweet if any exist, then we delete the extra spaces from the text data. After this, we remove the hashtag and '@' from the tweet data. Then delete all the links and the text whose length is less than 3 from the tweet as it doesn't carry any information. At the end we remove all the stop words, punctuations then the tweet is converted into tokens.

2. Feature Extraction:
In this step we extract the all the useful feature that is needed for next step.

3. Applying VADER Sentiment Analyzer and Sentiment Score Calculation:
In this, we import the pre-trained VADER algorithm of Sentiment Analyzer. The sentiment value is derived on the basis of every word's intensity.

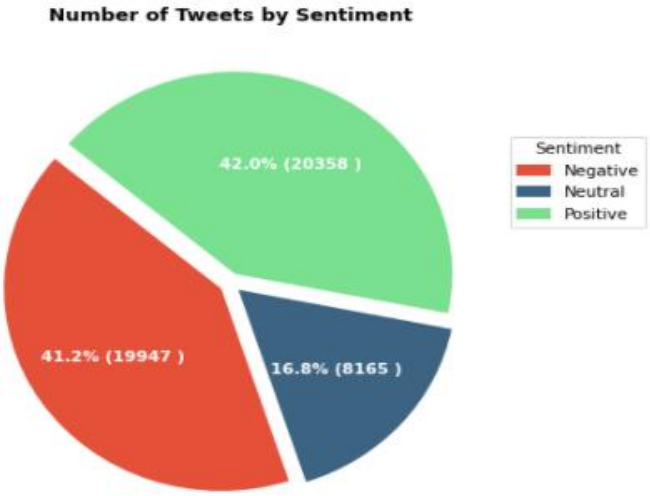


Fig. 4 Number of Tweets by sentiment

4. Sentiment Classification:
Once score is calculated the sentiment is classify as positive,negative,neutral.


Fig. 5 positive tweets


Fig. 6 Negative tweets

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

## V. EXPERIMENTS AND EVALUATIONS

1. Dataset:
We'll extract topics from a million news headlines collected from the respectable Australian news outlet ABC(Australian Broadcasting Corp.) in this exercise. The dataset can be found on Kaggle. Dataset content three column: publish_date , headline_text and Start Date.

2. Data Analysis:
We'll start exploratory data analysis now that we've imported our data to get a better idea of what's in it.

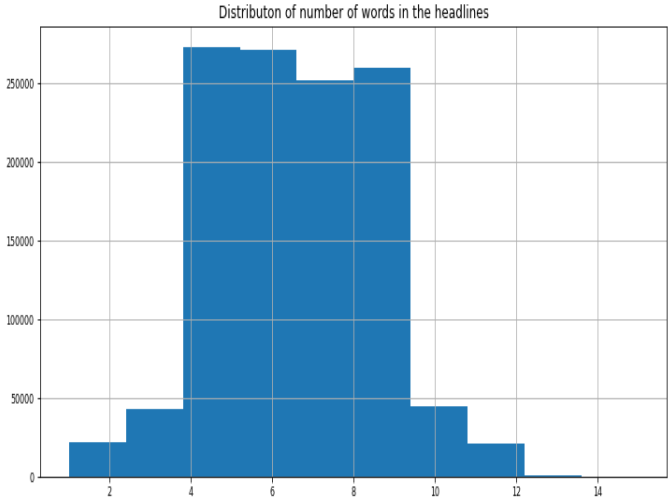


Fig. 7. N umber of words in the headline

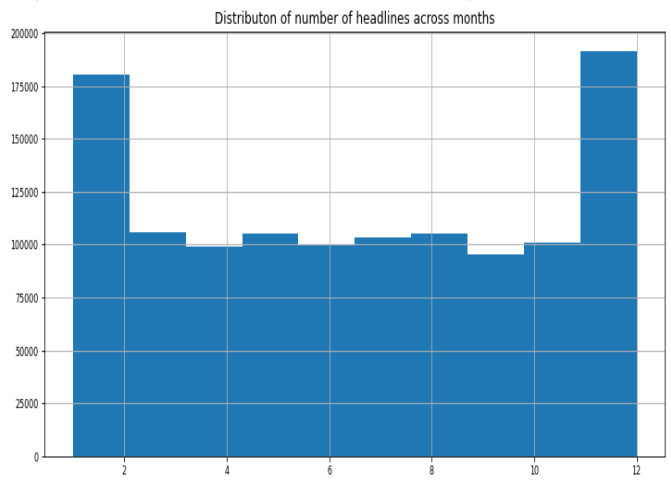


Fig. 8 Number of headline across months

3. Data Cleaning:
Let's continue to clean up the headlines by converting each word to lowercase, deleting punctuation, and deleting non-ASCII characters that aren't important to modeling topics.

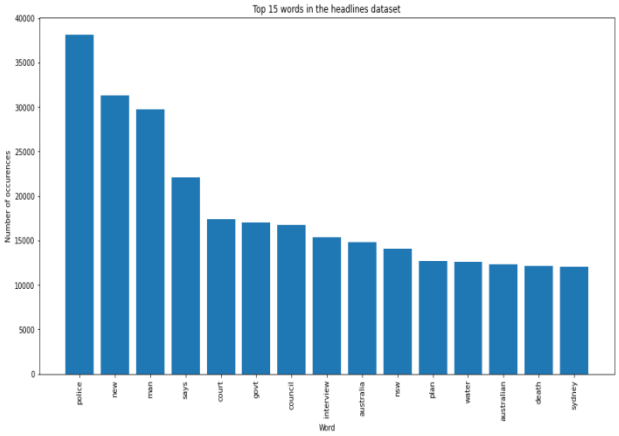


Fig. 8. Top 15 words in the headlines

4. Clustering using 'wordtovec' embeddings:
We will import the word embeddings from the pre-trained deep NN on google news and then represent each headline with the mean of word embeddings for each word in that headline. Now, we will randomly sample 20% of the data because of the memory constraints and then build the clustering model using the word embeddings we just imported. Now we have 22343 headlines and each headline has 300 features. Let us use KMeans CLustering to cluster them into 8 clusters.

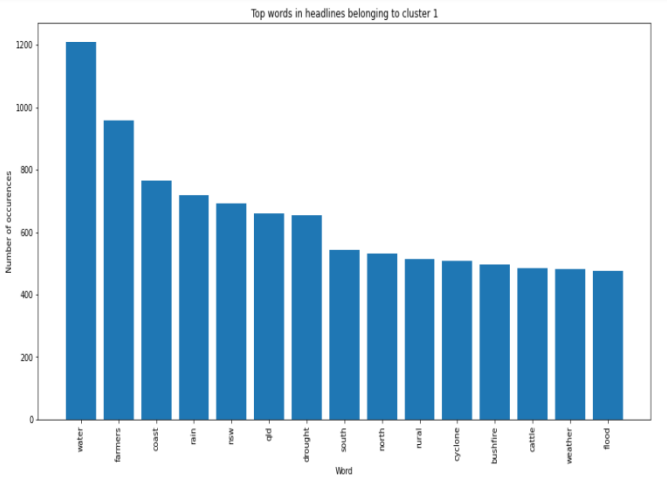


Fig. 10. T opic1 of Cluster

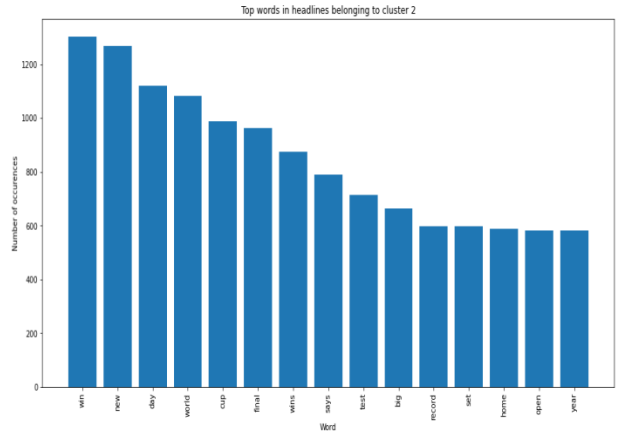


Fig. 11. Topic 2 of Cluster

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

## VI. CONCLUSIONS

In this work, we have used the word embedding model for analysis of different topics present in new headlines which is capable of capturing syntactic similarity and semantic relation, the context of a word in a document, relation with other words. we have created the group of 8 different clusters of the present in the document. And also one of the applications is implemented which is sentiment analysis using the VADER Rule-based Model for Sentiment Analysis of Social Media Text .

## ACKNOWLEDGMENT

I would like to thank my mentor Prof. Grishma Sharma. through her guidance and vigilance, I've accomplished this, thank you mam.

## REFERENCES

[1] F. Esposito, A. Corazza, F. Cutugno, "Topic Modelling with Word Embeddings," in IEEE Transactions on Content Mining, Vol.7, April 2018.

[2] Adji B. Dieng, Francisco J. R. Ruiz, David M. Blei, "Topic Modeling in Embedding Spaces," in ACL Information Retrieval (cs.IR); Computation and Language (cs.CL); Machine Learning (cs.LG); Machine Learning (stat.ML), Vol. 6, 8 Jul 2019.

[3] Lingyun Li; Yawei Sun; Cong Wang,"Semantic Augmented Topic Model over Short Text.", 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS).Nov. 2018.

[4] C.J. Hutto, Eric Gilbert,"VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text", in IEEE 8th Eighth International Conference on Weblogs, Nov2014.

[5] Bhagyashree Vyankatrao Barde, Anant Madhavrao Bainwad,"An overview of topic modeling methods and tools", in IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), Jan 2018.

[6] P. Anupriya, S. Karpagavalli,"LDA based topic modeling of journal abstracts", in IEEE International Conference on Advanced Computing and Communication Systems, Nov 2015.

[7] Dandan Song; Jingwen Gao, Jinhui Pang, Lejian Liao, Lifei Qin, "Knowledge Base Enhanced Topic Modeling",in IEEE International Conference on Knowledge Graph (ICKG), Sept 2020.

[8] Yang Gao, Yue Xu; Yuefeng Li, "Pattern-Based Topic Models for Information Filtering,in IEEE 13th International Conference on Data Mining Workshops,March 2014.

[9] Biao Wang, Yang Liu, Zelong Liu, Maozhen Li, Man Qi,"Topic selection in latent dirichlet allocation",in 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),Dec 2014.

[10] Zhenzhong Li, Wenqian Shang, Menghan Yan ,"News text classification model based on topic model",in IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),June 2016.

[11] David Alfred Ostrowski,"Using latent dirichlet allocation for topic modelling in twitter",in Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015),March 2015.