Approach the use of Bert & Word2vec in Text Segmentation of New York Times Data

Prudhvi Paruchuri, Nimish Kulkarni

Abstract: -

It has grown to be accepted practice in machine learning to represent words as numerical vectors based on the contexts in which they appear. We offer instructions for training these representations on headlines from New York Times articles in this publication. We go into detail about the various word representations, pre-trained data word vector embeddings that are accessible, intrinsic, and extrinsic evaluation, applications, and limits of these methods.

Keywords: - Natural Language Processing (NLP), Bert, Word2vec, Text Analysis, Machine Learning,

Introduction: -

The New York Times has a large readership and is influential in determining the public conversation, particularly in the USA, as well as how people view current events. The articles' comment sections are frequently updated and provide an insight into the opinions of readers on the subjects covered in the articles.

Both writers and readers are heavily involved due to the contemporary environment's growing demand for internet news articles.
News articles have evolved into a forum for addressing current geopolitical unrest, the environment, the media, etc. Newspaper firms are rapidly using machine learning (ML) and artificial intelligence (AI) techniques to analyse text data for factors such as customer sentiment, customer evaluations, future issues prediction, and information discovery and understanding. Free-text articles can provide useful information. The vocabulary used in articles and comments, however, is unstructured, filled with unique characters, and is extremely particular to the genre. One of the biggest issues is still turning free text into an unbiased form that NLP can use.

In this paper, we examine word embeddings, a well-liked technique for capturing the semantics of text data, with an emphasis on how this technique might be usefully used to describe text data.

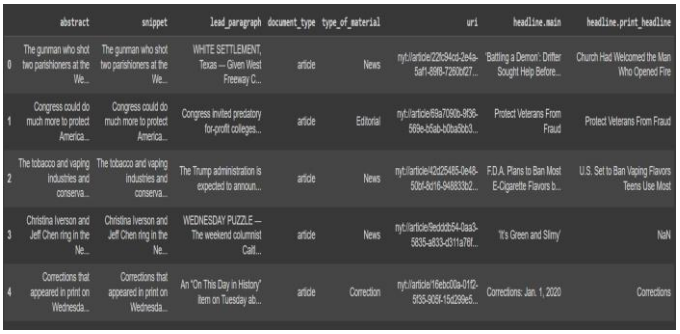
we go into detail about model training techniques (including information on the data needed to learn these representations), evaluation processes, applications, and evaluation.

Our main goal is to analyse the given New York times which would help us to determine the textual meaning of the given data, understanding the relevance of the information, building an effective model in tackling the user's point of view, as well as evaluating the metrics.

In this paper, we try to implement Bert and Word2vec models for text analysis on the pre-processed data using Python as code language. We also try to find their relations among different attributes using Exploratory Data Analysis, split the data into train and test set to validate our model's performance and visualizing our given measurements in a meaningful way.

Data: -

The dataset that we have worked on is of 154 articles of New York Times published from January 2020 to October 2022. The data consists of 8 Attributes and 153062 rows which is quite sufficient for our analysis.



	abstract	snippet	lead_paragraph	document_type	type_of_material	uri	headline.main	headline.print_headline
0	The gunman who shot two parishioners at the We...	The gunman who shot two parishioners at the We...	WHITE SETTLEMENT, Texas — Given West Freeway C...	article	News	nyt://article/22b94cd-2e4a-5af1-89f8-7260fd27...	'Battling a Demon': Drifter Sought Help Before...	Church Had Welcomed the Man Who Opened Fire
1	Congress could do much more to protect America...	Congress could do much more to protect America...	Congress invited predatory for-profit colleges...	article	Editorial	nyt://article/69a7090b-9f36-569e-b5ab-b0ba5bb3...	Protect Veterans From Fraud	Protect Veterans From Fraud
2	The tobacco and vaping industries and conserva...	The tobacco and vaping industries and conserva...	The Trump administration is expected to announ...	article	News	nyt://article/42d25485-0e48-50b4-8d16-948833b2...	F.D.A. Plans to Ban Most E-Cigarette Flavors b...	U.S. Set to Ban Vaping Flavors Teens Use Most
3	Christina Iverson and Jeff Chen ring in the Ne...	Christina Iverson and Jeff Chen ring in the Ne...	WEDNESDAY PUZZLE — The weekend columnist Calif...	article	News	nyt://article/9eddcb54-0aa3-5835-a833-d311a76f...	'It's Green and Slimy'	NaN
4	Corrections that appeared in print on Wednesda...	Corrections that appeared in print on Wednesda...	An "On This Day in History" item on Tuesday ab...	article	Correction	nyt://article/16ebc00a-0122-5f35-905f-15d299e5...	Corrections: Jan. 1, 2020	Corrections

The attributes consist of: -

- Abstract (Objective of the topic)
- Snippet (Summary of the topic)
- Lean_paragraph (First paragraph of news)
- Document_type (article, audio, media)
- Type_of_material (News, Editorial, etc.)
- Uri (article web url)
- Headline.main (headline of article[uncleaned])
- Headline.print (headline of article[cleaned])

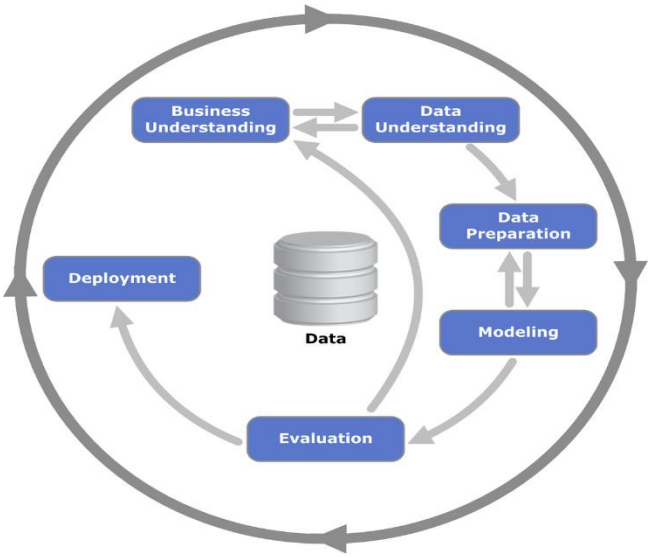
Framework: -

The experiment's objective is to identify a correlation between Article Abstract, Paragraph with Document Type, Material Type. The complete dataset is not viable because it contains several attributes and, more crucially, numerous null and special characters. We also try to determine the number of unique words generated, followed by Topic modelling using Latent Dirichlet Allocation technique. To overcome, we propose following outcomes: -

- Organize articles according to their article type.
- Generating word cloud for the given text.
- Generating basic text statistics with Topic modelling.
- Using Word2vec in classifying Paragraph text with the type of material.
- Using Bert Model in predicting the model classification with respect to document type.

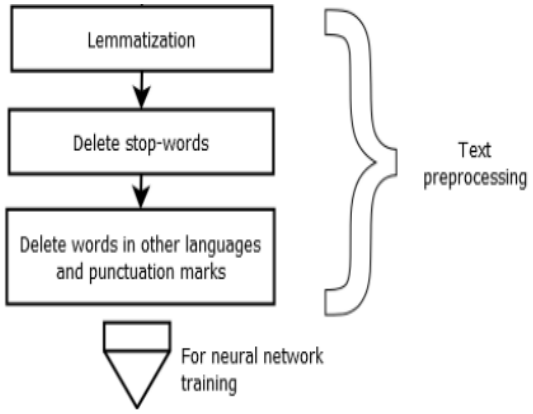
Analysis: -

For most of the Data Science concepts studied, we tried to use CRISP-DM approach in determining the business needs.



1. Text Pre-processing: -

   Our results may not be accurate or efficient and may be challenging to comprehend and evaluate if the raw text data contains undesirable or unimportant text. Raw data must therefore undergo proper pre-processing. Firstly, we select all the data that needs to be analysed. Secondly, we try to remove all the alphanumeric characters followed by Lemmatization of characters. Thirdly, we remove the stop words from the existing dataset as it doesn't convey any relevance for the analysis.



2. Statistics: -

   To proceed further with the analysis, we also tried to find out the total number of word count with respect to its unique words present within the dataset. Additionally, we calculated the entropy of the text data.

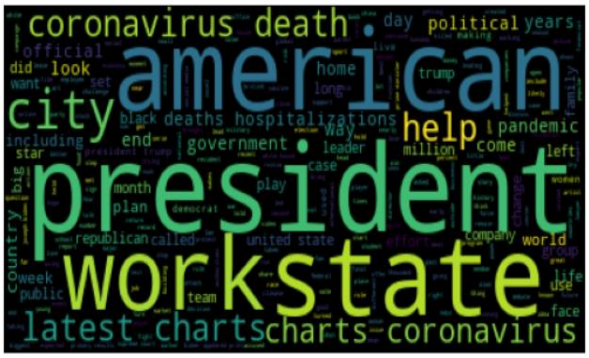
```
the following are some basic statistics on the input text
  total number of words: 1226905
  total number of unique words: 52035
  total entropy in the text: 0.5876650732456625
(entropy is a measure of information rate)
```

The concept of entropy in information theory is rather simple. Let's say you have a procedure (like a language L that generates words). There is a chance p that the thing that happened (the event) was going to happen at each stage of the process. The quantity of surprise is equal to -log(p), where p is the logarithm in the base of your choice (equivalent to changing units). Events with low probability often surprise people. Events that were expected to occur (p=1) have no unexpected outcomes. Events with p=0 impossibility surprise by infinity.

From the above result, our text has achieved the total count of words to be 122,6905 while the unique words are 52035. Our entropy has achieved the rate of 0.587 which is a reasonable value.

3. Word cloud: -

Word Clouds are used to visualize words in a corpus with their size representing their occurrences in the text we want to analyse. It is a fun and intuitive way to understand text. The word cloud is trained with the "abstract" column of the data which represents the outline of the article. This is a valuable source of information rather than the "lead_paragraph" column which only has the first paragraph.

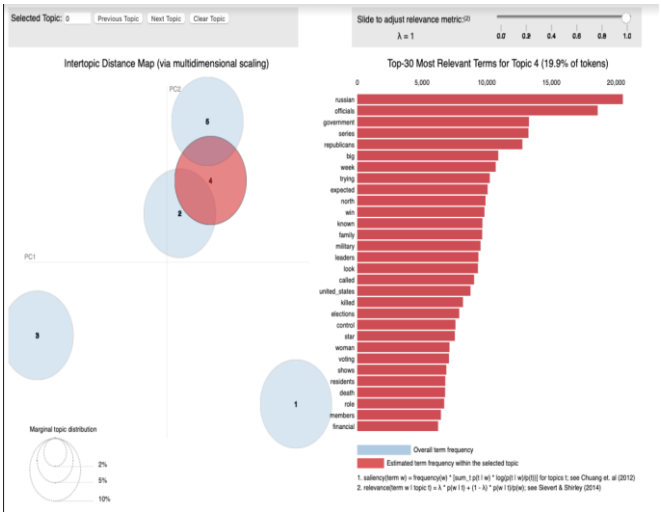


Now, in the Word Cloud we obtained, we can see the words with bolder and bigger size which are the most repeated. Words like 'president' refer to Donald Trump because the dataset was from 2020. We know the story of what Donald Trump did during COVID when he was in office. Due to the rise in COVID-19 cases all over the world, we can see that people are reading about latest charts and charts coronavirus, deaths, hospitalizations. Since more articles are concerned about coronavirus, we have the state word because each state had their own set of rules for masks. Regarding masks, we can see terms like use face and mask. Work is one more most important term, due to covid workplaces have been affected, people were switching from in-person to at-home work. So, we can certainly see that all the articles were mostly related to corona virus in USA.

4. Topic – modelling: -

Topic modelling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives. It helps us to understand hidden meaning behind the different hints of collection, helps us classifying articles based on topics.

Latent Dirichlet Allocation (LDA) is the methodology used for our analysis. Based on the Dirichlet distributions and processes, classifies, or categorizes the text within a document and the words per topic.



In this we try to understand the top topics in the data, it is mostly relatable to the word cloud that we generate.

In the topic modelling, we have five topics from the data, the first one talks about all the cities in the world. The second cluster or topic talks about politics, all the terms related to topics are clustered into the 2nd cluster. The third cluster talks about results (the presidential results), now this one is closely related to federal, republican, democrat, elections. This one is widely separated from the 2nd one which is close to 4th and 5th clusters. The 4th cluster talks mostly about Russia, Ukraine, military. The 5th cluster speaks about state, trump, republican, California, and topics as such.

Adding to the topic analysis, we generated top bigrams of words along with its top topics.

Word1	'Coronavirus deaths'
Word2	'Latest charts'
Word3	'Charts coronavirus'
Word4	'Deaths hospitalizations'
Word5	'United states'
Word6	'President trump'
Word7	'Appeared print'
Word8	'Primary results'
Word9	White house
Word10	Supreme court

The top 5 topics are: -

- City,world,election,help,Russia,country,times,end,Donald,set.
- Political,house,use,company,biden,florida,work,public,party,way.
- Results,president,republican,American,climate, Ukraine,home,social,Ukrainian,midterm.

- Russian,officials,government,series,republicans,big,week,trying,expected,north.
- State,years,California,day,federal,war,long,young,democrats,trump.

Text Vectorization: -

Two methods were used to represent words in a vector space: word2vec and BERT in the framework of this study.

Word2Vec: -

The concept behind Word2Vec is essential in solving many natural language processing problems. It helps a computer understand how a human understands language, just like, we can figure out "You need to bear with his". But a computer might think we were talking about an animal, which is why we need models like this to train the computer. In Word2Vec we make embedding vectors of a specified length for each word. For this, we have two main architectures which support Word2vec, they are CBOW and Skip-gram Models. Skip gram helps predict the context while CBOW will take predict missing words from the input.

Two important factors must be taken into consideration when training word2vec embeddings: 1) the number of embedding dimensions (typically between 50 and 500, tuned experimentally); and 2) the length of the context window (i.e., how many words should be used as context for training the word embeddings before and after the target word, typically 5 or 10 words). The embeddings must be large enough to distinguish between words even though training embeddings with more dimensions often necessitates more training data because each dimension should capture some aspect of meaning.

In this dataset, the lead_paragraph is compared with the material type which consists of Letters, video, news, editorial, etc. In the model that we trained had an accuracy of nearly 77.3%. The nine words with displayed in the output refer to the words like 'trump'. In Word2Vec models, we can find the words closer to our preferred word. This will change depending on the parameter we use to develop the model. To train the model we use the 'lead paragraphs' as inputs and 'type of material' as the target variable. To predict an input, we need to first convert each word into its corresponding vector in the Word2Vec model we developed and then we can use any classification algorithm to predict the target. In our case, the target variable in multi-class, we need to be mindful of this.

words	accuracy
talked	0.888
crises	0.84
scott	0.834
cost	0.834
guests	0.833
reducing	0.833
renters	0.832
coffee	0.832
helicopter	0.832
decline	0.83

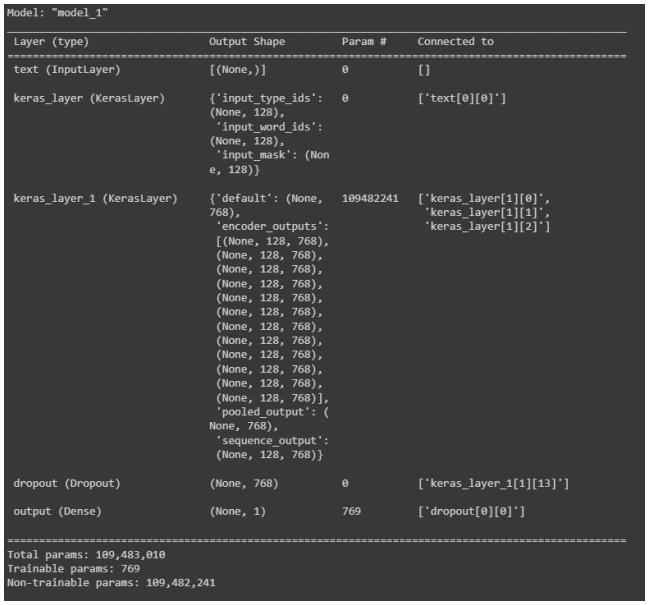
Bert Model: -

Bidirectional Encoder Representations from Transformers is known as BERT. BERT models assist computers in deciphering and interpreting text. To grasp the context, it refers to the paragraph that comes right before it. To determine the true meaning of words, it also examines the links between words inside a sentence.

The given sentence will subsequently be transformed into an embedding vector by BERT. The distinctive words in each manuscript are represented by an embedding vector. BERT guarantees that words with the same meaning will be represented similarly.

While machine learning is effective with numbers, it is ineffective with language. BERT transforms the input text into embedding vectors as a result. The model can readily operate with the numbers that make up the embedding vectors.

For the given text data, we compare the snippet and abstract of the text with the given document type. We convert the document type from string to numerical values assigned to them with variables for ease in classification. All the given operations are performed using TensorFlow. We use uncased Bert model. In this approach, the model learns an inner representation of the English language that can subsequently be utilized to extract features helpful for downstream tasks.



```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
==================================================================================================
 text (InputLayer)           [(None,)]                 0           []

 keras_layer (KerasLayer)    {'input_type_ids':        0           ['text[0][0]']
                              (None, 128),
                              'input_word_ids':
                             (None, 128),
                              'input_mask': (Non
                             e, 128)}

 keras_layer_1 (KerasLayer)  {'default': (None,        109482241   ['keras_layer[1][0]',
                             768),                                  'keras_layer[1][1]',
                              'encoder_outputs':                    'keras_layer[1][2]']
                              [(None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768),
                              (None, 128, 768)],
                              'pooled_output': (
                             None, 768),
                              'sequence_output':
                              (None, 128, 768)}

 dropout (Dropout)           (None, 768)               0           ['keras_layer_1[1][13]']

 output (Dense)              (None, 1)                 769         ['dropout[0][0]']

==================================================================================================
Total params: 109,483,010
Trainable params: 769
Non-trainable params: 109,482,241
```

The prediction probabilities will range from 0.0 to 1.0 because we employed a sigmoid activation function. Therefore, the output should be 1 if the forecast result is greater than 0.5, and 0 if it is less than 0.5. The model is later run on 15 epochs which achieves a 98% accuracy.

Results: -

Word2Vec	77.3%
Bert uncased	98.7%

As you can see the difference in accuracies, Word2vec is able to generate same results for each word embedding irrespective of its meaning. While on the other hand, Bert model captures different word embeddings with respect to its context of words.

Given the wide variation in word embedding performance across tasks, values are frequently inconclusive. Because of this, it's crucial to compare word embeddings using both intrinsic and extrinsic metrics.

Conclusion: -

There aren't many publicly accessible pre-trained word embeddings models now for generic situations. The fact that data privacy laws frequently forbid the distribution of any models developed using those data is one of the data's key problems. More investigation into the potential bias, interpretability, and privacy effects of these word embedding models would help us learn more about the models. Because of the engineering accomplishments and results obtained, BERT has an impact. Theoretically speaking, BERT represents a very, better results than Word2vec.

References: -

1. Dataset https://www.kaggle.com/datasets/cascaschatz/new-york-times-articles-metadata-2020-2022
2. https://medium.com/@dilip.voleti/classification-using-word2vec-b1d79d375381
3. https://towardsdatascience.com/simple-wordcloud-in-python-2ae54a9f58e5
4. https://www.kdnuggets.com/2019/02/word-embeddings-nlp-applications.html
5. https://www.saltdatalabs.com/blog/word2vec-vs-bert
6. https://www.tensorflow.org/text/tutorials/classify_text_with_bert
7. https://ieeexplore.ieee.org/document/8996183
8. https://towardsdatascience.com/nlp-extracting-the-main-topics-from-your-dataset-using-lda-in-minutes-21486f5aa925