Recommending News Articles Based on Read Articles by Using Natural Language Processing (NLP) in Machine language

Course Name & Code:

Research Design and Applications for Data and Analysis (SWE – 686)

Project Report

Prepared By (Name with ID)

Md Rakib Hasan (3002)

Md. Ashraful Alam (3011)

Students of M. Sc in SWE
Department of Software Engineering
Daffodil International University

Submitted to

Mr. Fazly Rabbi

Faculty of Department of Software Engineering
Daffodil International University

Date: May 29, 2023

Table of Contents

Chapter 01: Abstract

This project focuses on the classification of news articles using Natural Language Processing (NLP) techniques, specifically employing word embeddings and Recurrent Neural Networks (RNN). The aim is to develop an effective model that can accurately categorize news articles into predefined classes based on their content.

The project utilizes a dataset containing a collection of news articles from various sources. Preprocessing techniques are applied to clean the text data, including tokenization, removing stopwords, and stemming.

To capture the semantic meaning of words and their contextual relationships, word embeddings are employed. Word embeddings are dense vector representations that preserve semantic information. These embeddings are trained using techniques like Word2Vec or GloVe on a large corpus of text.

A Recurrent Neural Network architecture, specifically a Long Short-Term Memory (LSTM) model, is then implemented. LSTM is well-suited for sequence data processing due to its ability to capture long-term dependencies. The model takes the word embeddings as input and learns to classify the news articles into different categories.

The training process involves splitting the dataset into training and validation sets. The model is trained on the training set and optimized using techniques like backpropagation and gradient descent. Hyperparameter tuning is performed to find the optimal configuration for the model.

Finally, the performance of the model is evaluated on a separate test set, using metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of the proposed approach in accurately classifying news articles into their respective categories.

This project showcases the potential of combining NLP techniques, such as word embeddings and RNNs, to solve the challenging task of news article classification. The developed model can be valuable for various applications, including automated content tagging, recommendation systems, and information retrieval in the domain of news and journalism.

Keywords- NLP (Natural Language Processing), News articles, Classification, Word embeddings, RNN (Recurrent Neural Network), Tokenization, Stopwords, Stemming, Word2Vec, GloVe, LSTM (Long Short-Term Memory), Backpropagation, Performance evaluation, Accuracy.

Chapter 02: Introduction

The classification of news articles plays a vital role in organizing and retrieving information from vast amounts of textual data. With the exponential growth of digital content, there is a need for automated systems that can accurately categorize news articles based on their content. This project focuses on addressing this challenge by leveraging Natural Language Processing (NLP) techniques, specifically word embeddings and Recurrent Neural Networks (RNNs), to classify news articles into predefined categories.

The primary goal of this project is to develop an effective model capable of accurately categorizing news articles by learning the semantic meaning and contextual relationships between words in the text. Traditional approaches, such as bag-of-words representations, often fail to capture the nuanced meanings and contextual information present in natural language. Therefore, this project explores the use of word embeddings, which provide dense vector representations that encode semantic information.

Word embeddings are trained using techniques like Word2Vec or GloVe on large corpora of text data. These embeddings allow the model to capture the relationships between words, such as synonyms or related concepts, by representing them as vectors in a high-dimensional space. By utilizing word embeddings, the model can better understand the underlying semantic structure of the news articles.

In addition to word embeddings, this project employs Recurrent Neural Networks, specifically Long Short-Term Memory (LSTM) models, which are well-suited for processing sequential data. LSTM models can capture long-term dependencies and effectively model the sequential nature of text. The LSTM model takes the word embeddings as input and learns to classify the news articles into their respective categories.

The project follows a systematic workflow, starting with data preprocessing techniques such as tokenization, removing stopwords, and stemming to clean the text data. The dataset is then divided into training, validation, and testing sets to train and evaluate the model's performance. The model is trained using backpropagation and gradient descent, with hyperparameter tuning to optimize its configuration.

The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of the proposed approach in accurately classifying news articles, paving the way for various applications such as automated content tagging, recommendation systems, and information retrieval in the field of news and journalism.

This project combines NLP techniques, including word embeddings and RNNs, to develop a robust model for news article classification. The project aims to improve the efficiency and accuracy of categorizing news articles, contributing to the organization and retrieval of information in the digital era.

Chapter 03: Literature Review

The classification of news articles using NLP techniques has been a topic of extensive research in recent years. Several studies have explored various approaches to improve the accuracy and efficiency of news article classification. The following literature review highlights some relevant studies in this domain:

Mikolov et al. (2013) introduced the Word2Vec model, which revolutionized the field of word embeddings. Their work demonstrated the effectiveness of learning distributed representations of words that capture semantic relationships. Word2Vec has since been widely adopted in NLP tasks, including news article classification.[1]

Pennington et al. (2014) presented the GloVe (Global Vectors for Word Representation) model, which also focuses on learning word embeddings. GloVe utilizes global statistics of word co-occurrence to generate word vectors that capture semantic relationships. GloVe embeddings have been successfully applied in various NLP tasks, including news article classification.[2]

Zhang et al. (2015) proposed a CNN-based (Convolutional Neural Network) approach for text classification. Their model leveraged multiple convolutional filters to capture different levels of n-gram features in the text. This study demonstrated the effectiveness of CNNs in capturing local and compositional features, leading to improved news article classification performance.[3]

Hochreiter and Schmidhuber (1997) introduced the Long Short-Term Memory (LSTM) model, a type of RNN architecture specifically designed to capture long-term dependencies in sequential data. LSTMs have been widely adopted in NLP tasks due to their ability to model the context and sequential nature of text. LSTM-based models have achieved notable success in news article classification.[4]

Yang et al. (2016) proposed a Hierarchical Attention Network (HAN) for document classification. Their model incorporated attention mechanisms at both word and sentence levels to capture the most important information for classification. The HAN model demonstrated improved performance in news article classification by attending to relevant parts of the text.[5]

Xu et al. (2019) explored the use of pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers), for news article classification. By leveraging the contextualized word representations learned from large-scale language modeling, the BERT-based model achieved state-of-the-art performance in various NLP tasks, including news article classification.[6]

The reviewed literature demonstrates the importance of word embeddings and RNN-based architectures in news article classification. Techniques like Word2Vec, GloVe, CNNs, LSTMs, and attention mechanisms have significantly contributed to improving the accuracy and

efficiency of classification models. Moreover, the emergence of pre-trained language models, such as BERT, has further pushed the boundaries of performance in this field.

Building upon the existing literature, this project aims to leverage word embeddings and RNNs, specifically LSTM models, to develop an effective classification model for news articles. By incorporating these techniques, the project seeks to improve the understanding and categorization of news articles, contributing to the field of information organization and retrieval in the context of news and journalism.

Chapter 03: Methodology

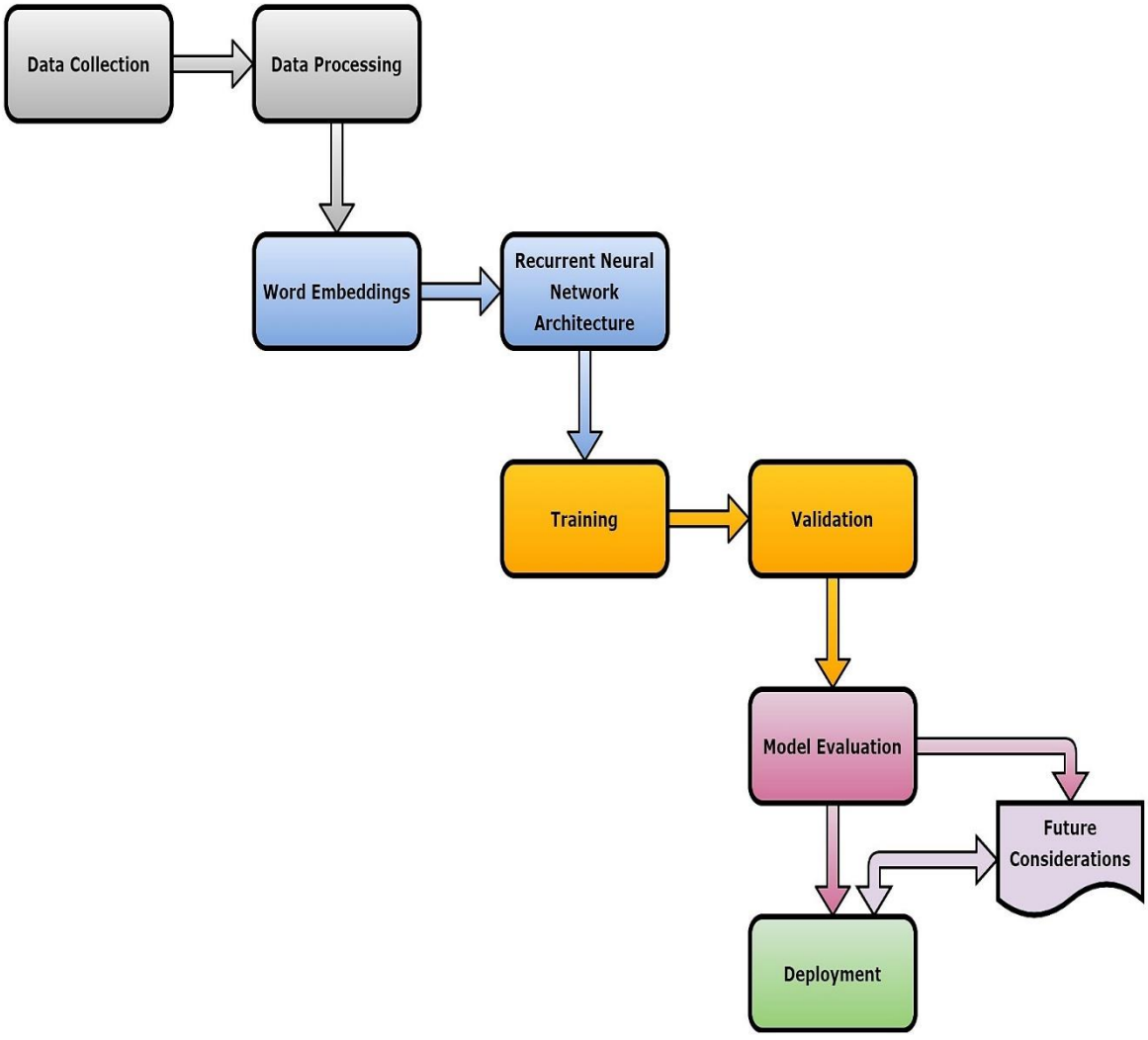


Figure-01: Methodology

The methodology of this project involves several steps to develop an effective news article classification model using word embeddings and Recurrent Neural Networks (RNNs). The following outlines the key stages of the methodology:

1. Data Collection and Preprocessing:

- Collect a dataset of news articles from various sources, ensuring it covers multiple categories.
- Perform data preprocessing, including text cleaning and normalization.
- Tokenize the text into individual words or subword units.
- Remove stopwords and perform stemming or lemmatization to reduce noise in the data.

2. Word Embeddings:

- Train word embeddings using techniques like Word2Vec or GloVe on a large corpus of text data.
- Generate dense vector representations for each word, capturing semantic relationships and contextual information.
- Assign the trained word embeddings to the words in the news articles.

3. Recurrent Neural Network Architecture:

- Utilize a Recurrent Neural Network (RNN) architecture, specifically a Long Short-Term Memory (LSTM) model.
- Design the LSTM model to take the word embeddings as input and learn to classify news articles into predefined categories.
- Configure the LSTM model with appropriate hyperparameters, such as the number of LSTM layers, hidden units, and dropout rates.

4. Training and Validation:

- Split the dataset into training and validation sets.
- Train the LSTM model on the training set using backpropagation and gradient descent algorithms.
- Optimize the model's performance by tuning hyperparameters, such as learning rate and batch size, using techniques like grid search or random search.
- Monitor the model's performance on the validation set to prevent overfitting and select the best-performing configuration.

5. Model Evaluation:

- Evaluate the trained LSTM model on a separate test set that was not used during training or hyperparameter tuning.

- Measure performance using metrics such as accuracy, precision, recall, and F1-score.
- Analyze the model's ability to correctly classify news articles across different categories.
- Compare the results with baseline models or existing approaches to assess the improvement achieved.

6. Model Deployment and Future Considerations:

- Once the model has demonstrated satisfactory performance, deploy it for practical use in classifying news articles into relevant categories.
- Consider future enhancements, such as incorporating ensemble techniques or exploring other NLP models like BERT or Transformer-based architectures.
- Continuously update and retrain the model as new data becomes available to improve its accuracy and adaptability.

By following this methodology, the project aims to develop a robust and accurate news article classification model that can effectively categorize articles based on their content. The utilization of word embeddings and LSTM models allows for better understanding and representation of the semantic relationships within the text, enabling more accurate classification results.

Chapter 04: News Articles Classification and Modeling

1. Research Problem Discussion:

Text is one of the most widespread forms of sequence data. It can be understood as either a sequence of characters or a sequence of words, but it's most common to work at level of words. Text-sequence processing includes following applications:

Applications of deep learning for text data:

1. Document classification
2. Articles labeling
3. Sentiment analysis
4. Author identification
5. Question-answering
6. Language detection
7. Translation Tasks

In true sense deep learning models map the statistical structure of written language, which is sufficient to solve many simple textual tasks and problems.

Deep learning for natural-language processing is pattern recognition applied to words, sequence, and paragraphs, in much similar way that computer vision is pattern recognition applied to pixels.
Deep-learning models don't take input as text like other models they only work with numeric tensors

Three techniques to vectorize the text data:

1. Segment text into words, and convert word into a vector
2. Segment text into characters, and transform each character into a vector.
3. Extract n-grams of words, and transform each n-grams into a vector.

- There are many ways one can convert text to vector and it depends on what models one is using along with time or resources utilization.

Typical workflow to prepare text data for machine learning models:

1. Tokenization
2. One-Hot encoding or word indexing
3. Pad sequencing
4. Embedding layer (Word2Vec)
5. Corresponding word vector

In this notebook, we are going to explore and solve news classification problem to classify 42 types of news headlines and news descriptions.

Use-case: Such text classification models are used in News Apps or by reporter to classify news topics for better reach to right audience.

Problem-statement: Build news classification model using deep learning techniques and deploy model for reporters to classify and label news articles.

2. About Dataset

2.1. Context

This dataset contains around 210k news headlines from 2012 to 2022 from HuffPost. This is one of the biggest news datasets and can serve as a benchmark for a variety of computational linguistic tasks. HuffPost stopped maintaining an extensive archive of news articles sometime after this dataset was first collected in 2018, so it is not possible to collect such a dataset in the present day. Due to changes in the website, there are about 200k headlines between 2012 and May 2018 and 10k headlines between May 2018 and 2022.

2.2. Content

Each record in the dataset consists of the following attributes:

category: category in which the article was published.

headline: the headline of the news article.

authors: list of authors who contributed to the article.

link: link to the original news article.

short_description: Abstract of the news article.

date: publication date of the article.

There are a total of 42 news categories in the dataset. The top-15 categories and corresponding article counts are as follows:

POLITICS: 35602

WELLNESS: 17945

ENTERTAINMENT: 17362

TRAVEL: 9900

STYLE & BEAUTY: 9814

PARENTING: 8791

HEALTHY LIVING: 6694

QUEER VOICES: 6347

FOOD & DRINK: 6340

BUSINESS: 5992

COMEDY: 5400

SPORTS: 5077

BLACK VOICES: 4583

HOME & LIVING: 4320

PARENTS: 3955

2.3. View of Dataset

	link	headline	category	short_description	authors	date
0	https://www.huffpost.com/entry/covid-boosters-...	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...	Carla K. Johnson, AP	2022-09-23
1	https://www.huffpost.com/entry/american-airlin...	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...	Mary Papenfuss	2022-09-23
2	https://www.huffpost.com/entry/funniest-tweets...	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...	Elyse Wanshel	2022-09-23
3	https://www.huffpost.com/entry/funniest-parent...	The Funniest Tweets From Parents This Week (Se...	PARENTING	"Accidentally put grown-up toothpaste on my to...	Caroline Bologna	2022-09-23
4	https://www.huffpost.com/entry/amy-cooper-lose...	Woman Who Called Cops On Black Bird-Watcher Lo...	U.S. NEWS	Amy Cooper accused investment firm Franklin Te...	Nina Golgowski	2022-09-22

Figure-02: Dataset of News Category

2.4. Dataset information & Descriptive Statistics

```
(209527, 6)
Unique categories: 42
--------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209527 entries, 0 to 209526
Data columns (total 6 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   link               209527 non-null  object
 1   headline           209527 non-null  object
 2   category           209527 non-null  object
 3   short_description  209527 non-null  object
 4   authors            209527 non-null  object
 5   date               209527 non-null  datetime64[ns]
dtypes: datetime64[ns](1), object(5)
memory usage: 9.6+ MB
```

Figure-03: data info

	link	headline	category	short_description	authors	date
count	209527	209527	209527	209527	209527	209527
unique	209486	207996	42	187022	29169	3890
top	https://www.huffingtonpost.comhttps://www.washingtonpost.com/politics/divisions-within-gop-over-trumps-candidacy-are-growing/2016/02/28/97b16010-de3a-11e5-8d98-4b3d9215ade1_story.html	Sunday Roundup	POLITICS			2014-03-25 00:00:00
freq	2	90	35602	19712	37418	100
first	nan	nan	nan	nan	nan	2012-01-28 00:00:00
last	nan	nan	nan	nan	nan	2022-09-23 00:00:00

Figure-04: Data Description

3. Profiling Report of Dataset

Panda Libraries provide a report dashboard by using this command function *df.profile_report()*

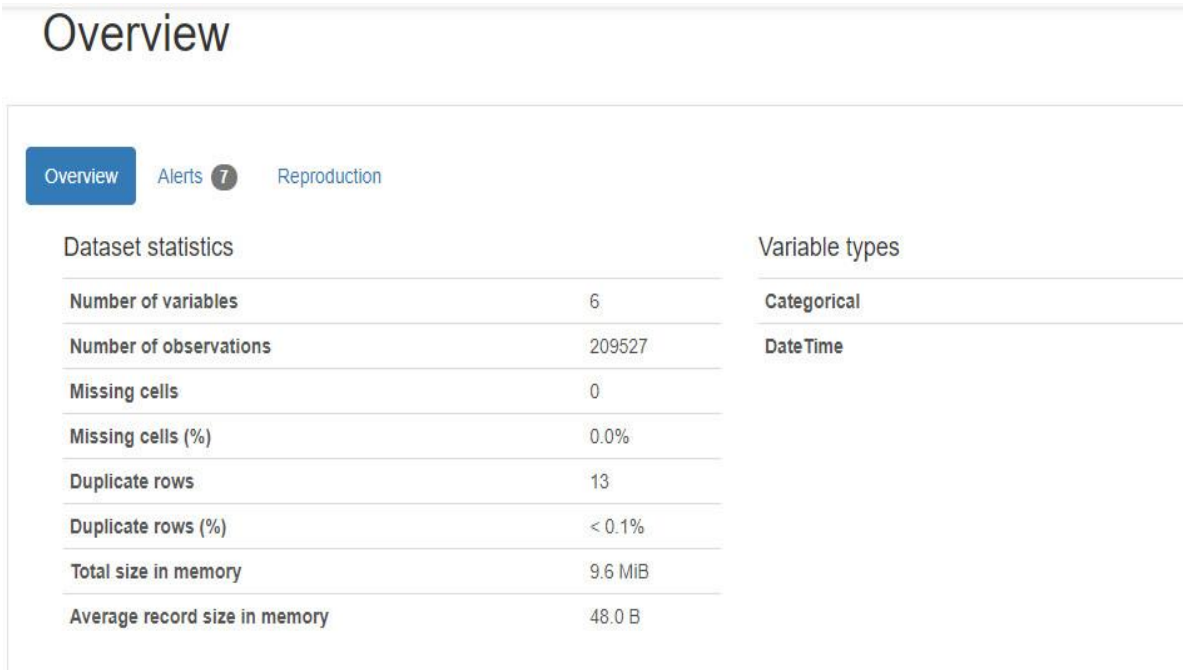


Figure-05: Report overview

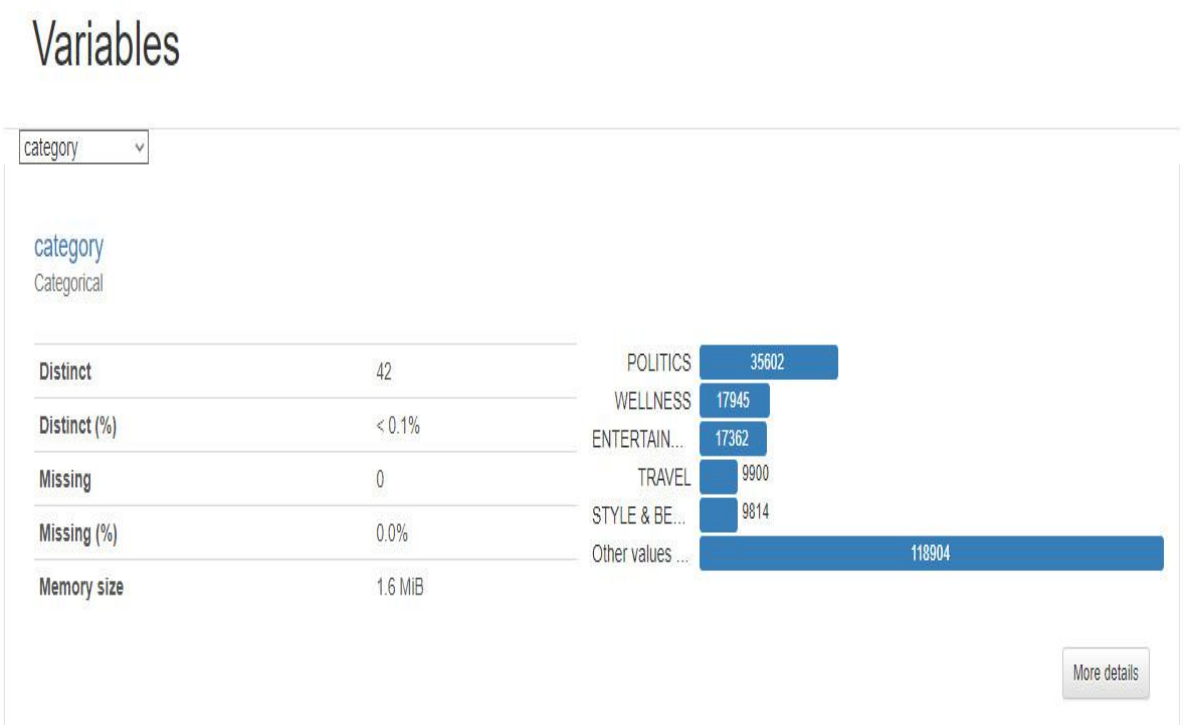


Figure-06: Report Variables

Figure-07: Report Missing Values

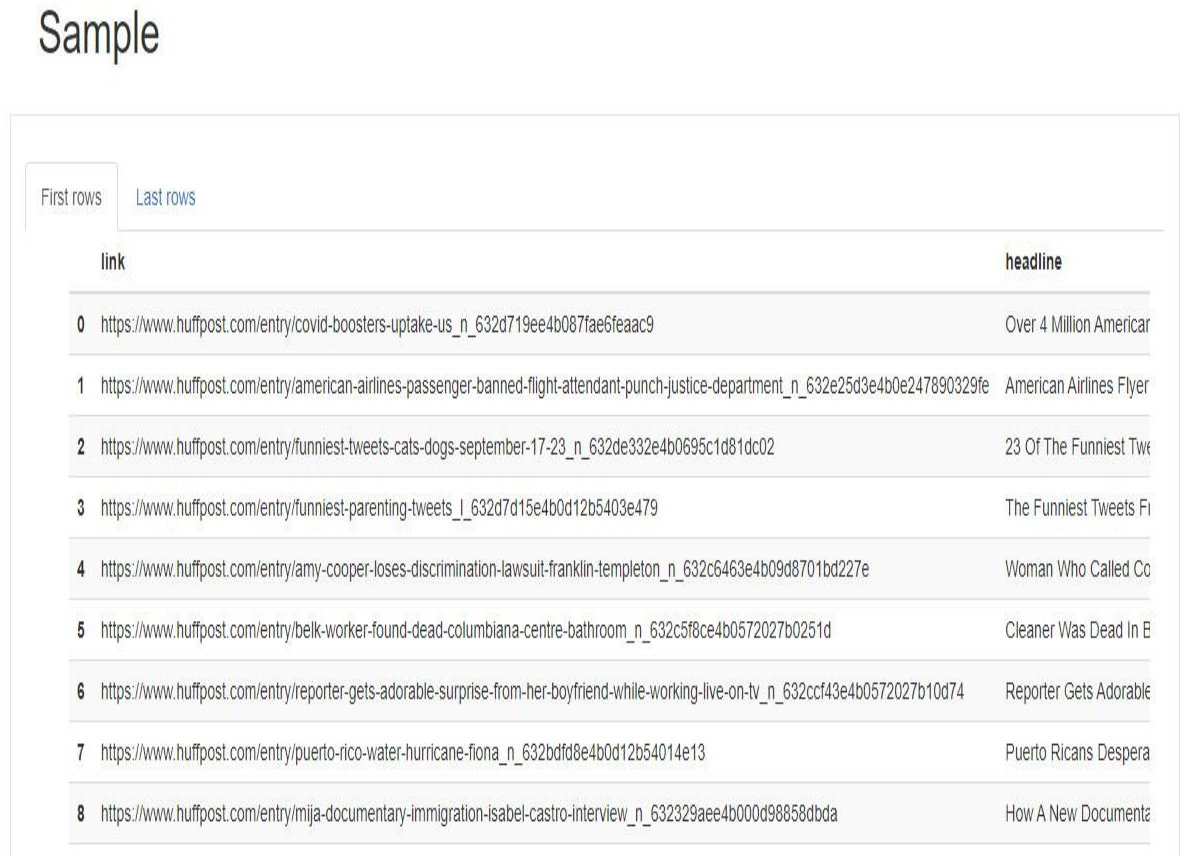


Figure-08: Report Sample

Duplicate rows

Most frequently occurring

	link	headline
0	https://www.huffingtonpost.comhttp://blogs.wsj.com/cio/2012/07/06/apple-removes-green-electronics-certification-from-products/	Apple Removes Green EPEAT
1	https://www.huffingtonpost.comhttp://d.repubblica.it/english/fashion/2012/02/21/video/video_frida_giannini_gucci-862602/1/	Gucci's Frida Giannini Reveals
2	https://www.huffingtonpost.comhttp://d.repubblica.it/english/fashion/2012/02/22/video/prada_video-865968/1/	Behind-The-Scenes Look At Pra
3	https://www.huffingtonpost.comhttp://d.repubblica.it/english/fashion/2012/02/23/video/video_prada-869060/1/	Versace Atelier Worker Reflects
4	https://www.huffingtonpost.comhttp://gizmodo.com/former-facebook-workers-we-routinely-suppressed-conser-1775461006	Former Facebook Workers: We
5	https://www.huffingtonpost.comhttp://www.businessinsider.com/google-is-attacking-apple-from-the-inside-out-and-its-working-2012-12?op=1	Google Is Attacking Apple From
6	https://www.huffingtonpost.comhttp://www.cnbc.com/2016/04/12/on-equal-pay-day-the-gap-is-still-too-wide-commentary.html	On Equal Pay Day, The Gap Is
7	https://www.huffingtonpost.comhttp://www.cnn.com/video/data/2.0/video/health/2014/02/07/crossfit-defends-crossfit-orig-jtb.cnn.html	The World's Most Dangerous W
8	https://www.huffingtonpost.comhttp://www.motherjones.com/politics/2016/05/trump-butler-anthony-senecal-facebook-kill-obama	On Facebook, Trump's Longtim
9	https://www.huffingtonpost.comhttp://www.nytimes.com/2012/09/23/opinion/sunday/the-optimal-diet.html?hp	Eating For Health, Not Weight

Figure-09: Report Duplicate Rows

Key findings:

Dataset has total 42 distinct categories of news articles 'Politics' is the most common category of news in our dataset. We have total of 29169 unique authors who have written various news articles Maximum length of headline is 320 while median length is around 152. Maximum length of description is 1472 while median length is around 120.

4. Text Data Visualization

Drop columns like authors, links and date as they are irrelevant to our problem.

	headline	category	short_description
0	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...
1	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...
2	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...
3	The Funniest Tweets From Parents This Week (Se...	PARENTING	"Accidentally put grown-up toothpaste on my to...
4	Woman Who Called Cops On Black Bird-Watcher Lo...	U.S. NEWS	Amy Cooper accused investment firm Franklin Te...

Figure-10: Dataset after drop some columns

5. TOP 10 Categories of News Articles

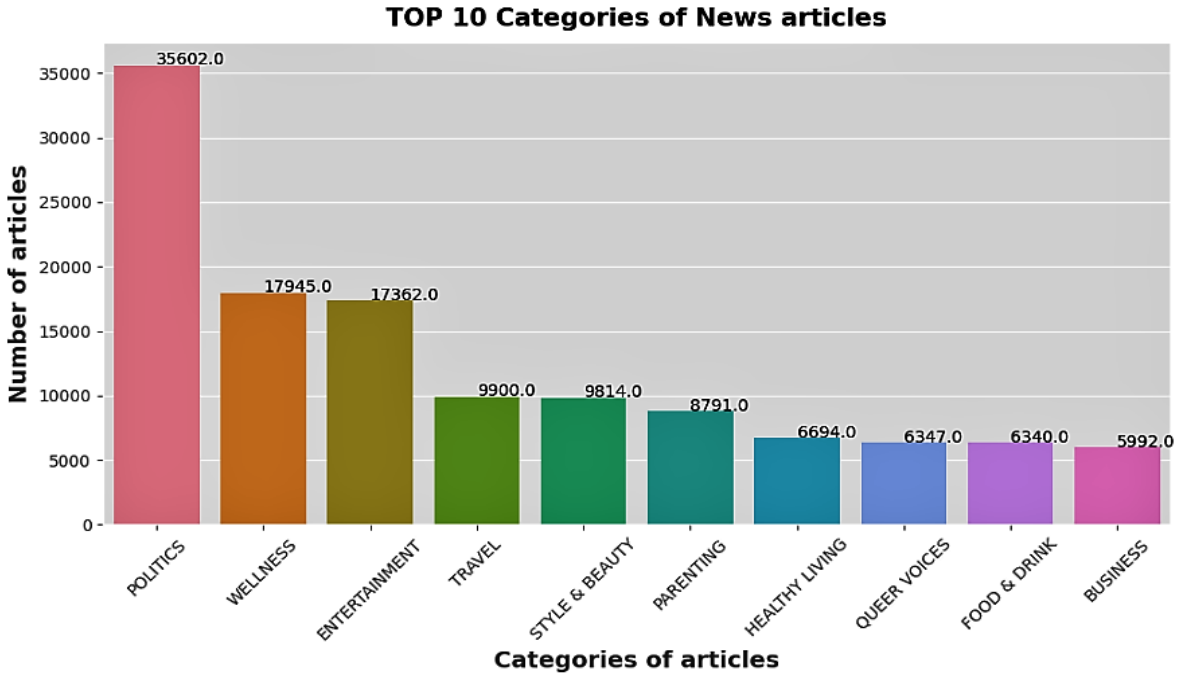


Figure-11: Top Ten News Articles wise bar chart

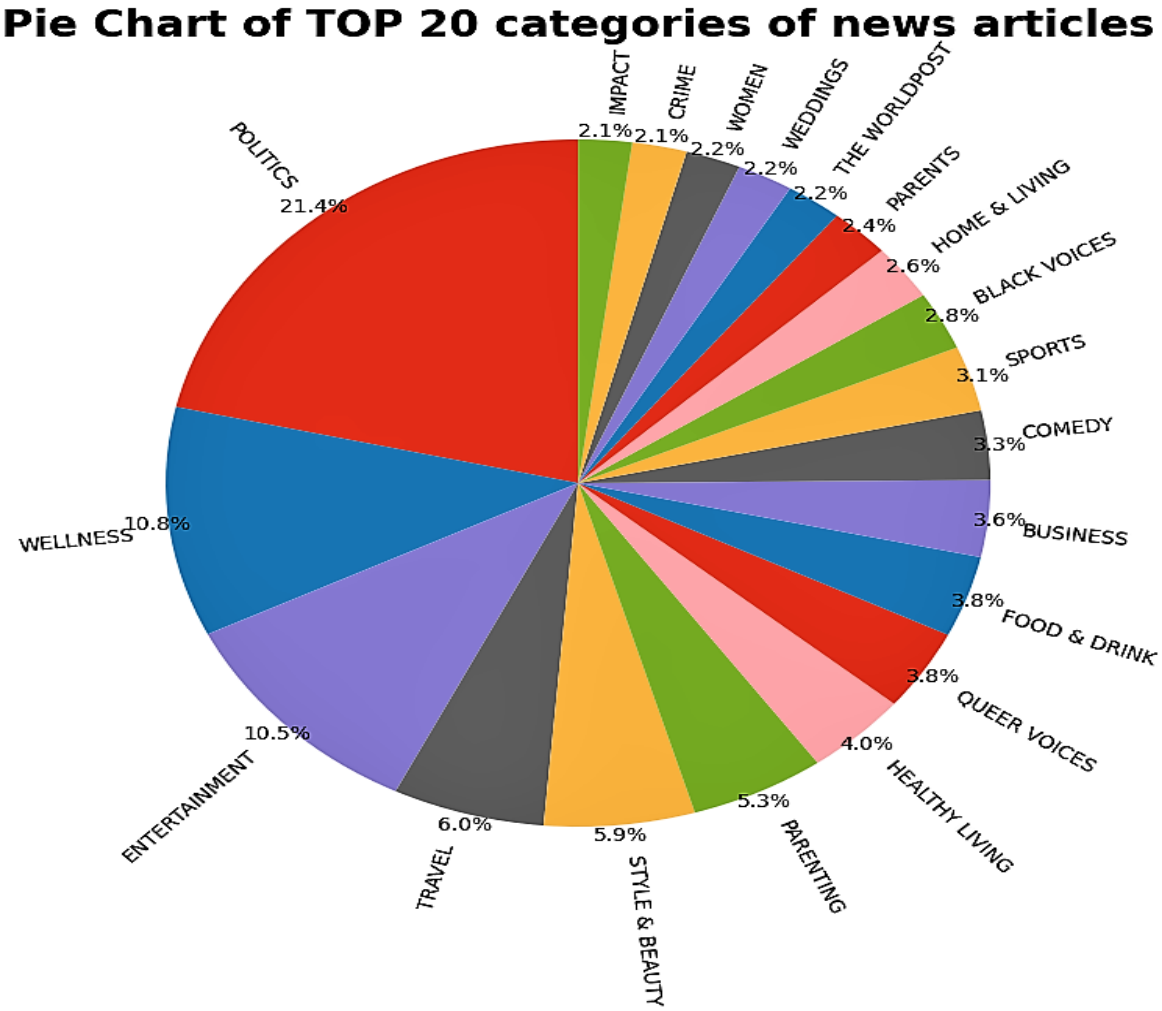


Figure-11: News wise pie chart

6. Lengths of 'headline' and 'short_description' of each category

6.1. List for maximum length of news in each category:

	category	len_news
24	POLITICS	1486
41	WORLDPOST	1424
4	COLLEGE	1225
16	HEALTHY LIVING	1073
38	WELLNESS	1036

Figure-12: News of Maximum Length

6.2. Minimum length of news in each category:

	category	len_news
35	U.S. NEWS	73
8	DIVORCE	66
33	THE WORLDPOST	56
36	WEDDINGS	54
21	MONEY	51

Figure-13: News of Minimum Length

6.3. The bar plots of max and min length of news articles:

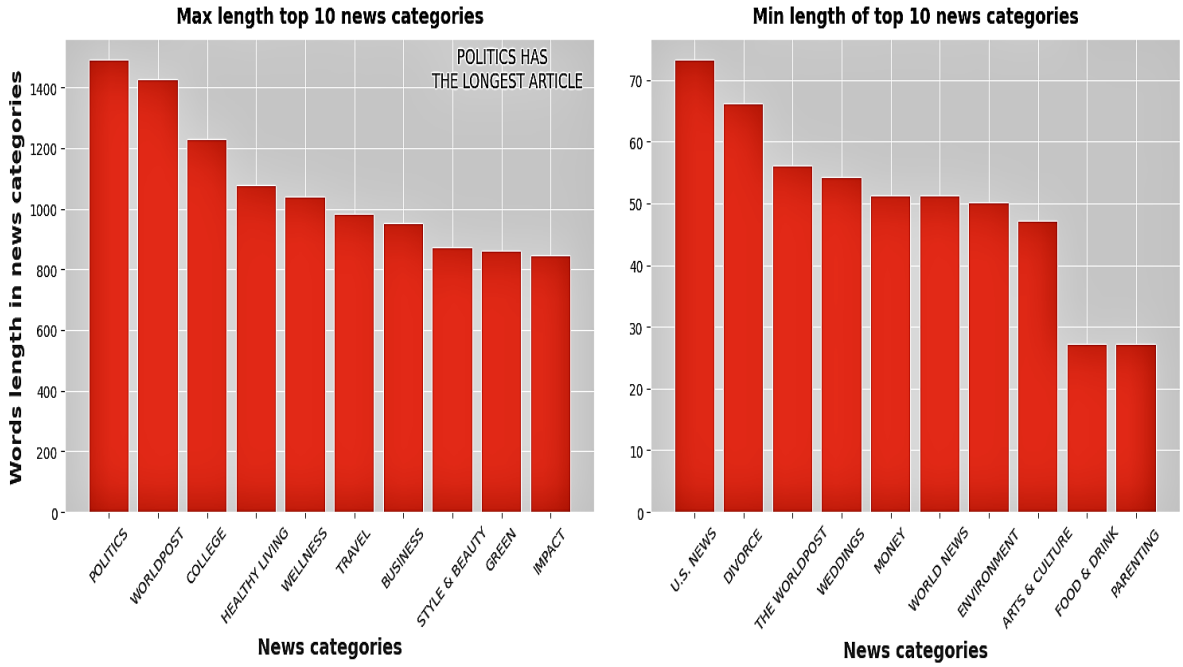


Figure-14: Comparison bar chart of Maximum & Minimum lengths news

Key findings:
From all the categories of news POLITICS has the greatest number of articles as well as length of characters in the article (headlines and short-description) After 'POLITICS' category, 'WORLDPOST', 'COLLEGE', 'HEALTHY LIVING','WELLNESS' categories are in top 5 most length of articles 'POLITICS', 'WELLNESS' and 'ENTERTAINTMENT' are top 3 categories having the greatest number of articles in our dataset.

7. Word Clouds of Categories and News Articles

7.1. Word cloud of categories of news articles in our dataset:

Figure-15: Words Clouds of Whole Dataset

7.2. Word Clouds New Data Frame of Category and Length Of Each News Articles in Those Categories:

Figure-16: Words Clouds for each category News Words

8. Text-data Preprocessing

Start with removing some data from politics data frame ndf. We create list of 16000 row labels of politics category. Then we drop these 16000 labels from the dataset. We clean the text data using regex and data cleaning function. We remove stop-words and word lemmatization.

Example of pre-processing using above function:
Text sentence before pre-processing:
Over 4 million Americans Roll Up Sleeves for Omicron-Targeted COVID Boosters Health experts said it is too early to predict whether demand would match up with the 171 million doses of the new boosters the U.S. ordered for the fall.

Text sentence after pre-processing:
million Americans roll sleeves omicron targeted covid boosters health experts say early predict whether demand match million dose new boosters us order fall

Apply data cleaning function to column 'length of news' then length of total characters before and after cleaning text data.
Old characters length of text data: **36169394**
New characters length of text data: **23045855**

Length of total words before and after cleaning text data.
Old word length of text data: **5942993**
New word length of text data: **3290751**

9. Tokenization And Vectorization

Some of the most common NLP terminologies
Document- Each and every training example used in text dataset known as a Document.
Corpus- Collections of documents called as a corpus of text data
Vocabulary (BoW)-Vocabulary or Bag-of-words is nothing but number of unique words are present in text corpus.
Stop words- Stop words are those used most commonly in any language, e.g., 'the', 'a', etc. they do not form any meaning to the context of the text
N-grams- N-grams is text representation in form of N words sequences to extract meaning and context out of each sentence or paragraphs.
Tokenization- It's an early step in NLP process to split text sentences into smaller words or tokens.
Vectorization- Machine do not understand text or words, so text data or tokens must be converted to corresponding word index or word vectors in order process text and build models. process of converting tokenized words into numerical vectors called as a vectorization.

One-hot encoding and word-indexing example on chunk of data.
Shape of stored results array: (5, 15, 91)

Token index of unique words:
{'million': 1, 'americans': 2, 'roll': 3, 'sleeves': 4, 'omicrontargeted': 5, 'covid': 6, 'boostershealth': 7, 'experts': 8, 'say': 9, 'early': 10, 'predict': 11, 'whether': 12, 'demand': 13, 'match': 14, 'dose': 15, 'new': 16, 'boosters': 17, 'us': 18, 'order': 19, 'fall': 20, 'american': 21, 'airlines': 22, 'flyer': 23, 'charge': 24, 'ban': 25, 'life': 26, 'punch': 27, 'flight': 28, 'attendant': 29, 'videohe': 30, 'subdue': 31, 'passengers': 32, 'crew': 33, 'flee': 34, 'back': 35, 'aircraft': 36, 'confrontation': 37, 'accord': 38, 'attorneys': 39, 'office': 40, 'los': 41, 'angeles': 42, 'funniest': 43, 'tweet': 44, 'cat': 45, 'dog': 46, 'week': 47, 'sept': 48, 'dont': 49, 'understand': 50, 'eat': 51, 'parent': 52, 'accidentally': 53, 'put': 54, 'grownup': 55, 'toothpaste': 56, 'toddlers': 57, 'toothbrush': 58, 'scream': 59, 'clean': 60, 'teeth': 61, 'carolina': 62, 'reaper': 63, 'dip': 64, 'tabasco': 65, 'sauce': 66, 'woman': 67, 'call': 68, 'cop': 69, 'black': 70, 'birdwatcher': 71, 'lose': 72, 'lawsuit': 73, 'exemployeramy': 74, 'cooper': 75, 'accuse': 76, 'investment': 77, 'firm': 78, 'franklin': 79, 'templeton': 80, 'unfairly': 81, 'fire': 82, 'brand': 83, 'racist': 84, 'video': 85, 'central': 86, 'park': 87, 'encounter': 88, 'go': 89, 'viral': 90}

One-hot encoding and indexing of train and test data.
Shape of Input Data: **(193527,)**
Shape of Target Variable: **(193527,)**
Length of Word Index: **180960**

10.What are the Wordembeddings?

A word embedding is a learned representation for text where words that have the same meaning and save similar representation

Reference: - Machinelearningmastery

This approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging NLP problems Word embeddings are alternative to one-hot encoding along with dimensionality reduction One-hot word vectors - Sparse, High-dimensional and Hard-coded

Word embeddings - Dense, Lower-Dimensional and Learned from the data

Keras library has embeddings layer which does word representation of given text corpus tf.keras.layers.Embedding(input_dim, output_dim, embeddings_initializer='uniform', embeddings_regularizer=None, activity_regularizer=None, embeddings_constraint=None, mask_zero=False, input_length=None, kwargs)**

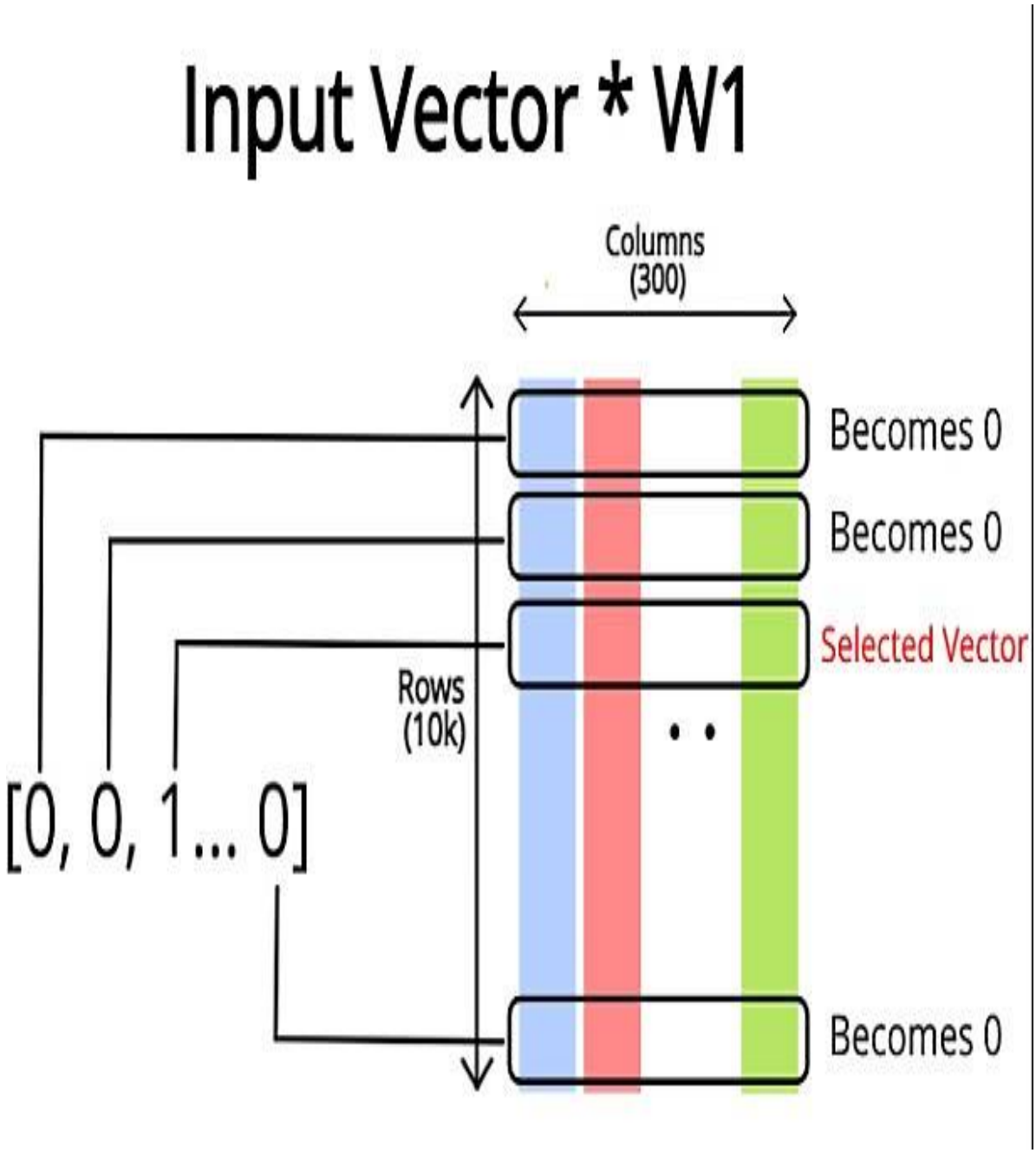
Key Arguments:

1) input_dim - Size of vocabulary - length of word index

2) output_dim - Output dimension of word representation

3) input-length - max input sequence length of document

Visual depiction of word embeddings: -



10.1. Model training using embedding layer and RNN (Baseline)

Baseline model 1 with SimpleRNN and embedding layer. Print the model summary *model.summary()*

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 130, 70)           12667200

 bidirectional (Bidirectiona  (None, 130, 128)         17280
 l)

 bidirectional_1 (Bidirectio  (None, 130, 128)         24704
 nal)

 simple_rnn_2 (SimpleRNN)    (None, 32)                5152

 dropout (Dropout)           (None, 32)                0

 dense (Dense)               (None, 42)                1386

=================================================================
Total params: 12,715,722
Trainable params: 12,715,722
Non-trainable params: 0
_________________________________________________________________
```

Figure-17: Sequential summary

10.2. What are the Recurrent Neural Networks??

A major difference between densely connected neural network and recurrent neural network, is that fully connected networks have no memory in units of each layer. while recurrent neural networks do store state of previous timestep or sequence while assigning weights to current input.

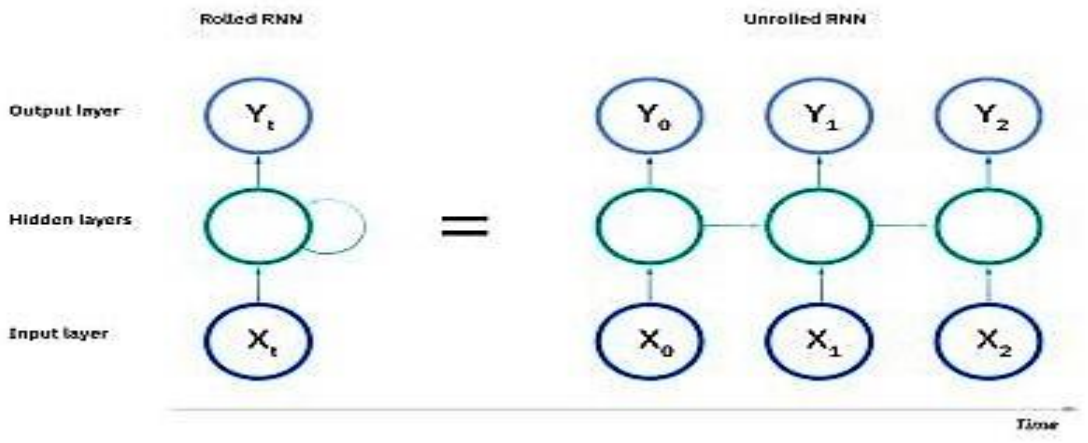
In RNNs, we process inputs word by word or eye saccade but eye saccade - while keeping memories of what came before in each cell. this gives fluid representation of sequences and gives neural network a ability to capture context of sequence rather than absolute representation of words.

"Recurrent neural network processes sequences by iterating through the sequence elements and maintaining a state containing information relative to what it has seen so far. In effect, an RNN is a type of neural network that has an internal loop."

-6.2 Understanding recurrent neural network, Deep learning using python by cholla

See the below depiction of how RNNs learns the context of sequences.

A visual dipiction of RNN cell with a loop and unrolled RNN cell.



```
Epoch 14/15
2949/2949 [==============================] - 709s 240ms/step - loss: 1.9089 - accuracy: 0.5067 - val_loss: 2.1350 - val_accuracy: 0.4607
Epoch 15/15
2949/2949 [==============================] - 727s 246ms/step - loss: 1.8991 - accuracy: 0.5110 - val_loss: 2.1390 - val_accuracy: 0.4550
test loss and accuracy: 2.1493613719940186 0.4507569968700409
```

Notes:

Experiment 1: Parameters: max_words=100000, output_dim=50, maxlen=50, epoch=10. Model is highly overfitting as training accuracy comes around 84% while test accuracy is barely 40%.

Experiment 2: Params: max_words=150000, out_dim=50, maxlen=80, epoch=10, added droput layers in RNN Model do not overfit but underfit and biased. val_acc comes again almost 39.3% while train accuracy is only 42%. model needs more parameters with more epochs

Experiment 3: Params: max_words=total_words, out_dim=70, maxlen=100, epoch=15, addding bidirection layer over both dropout RNN an improvement in train accuracy to 51.1% and val_accuracy to 45.5% max at epoch 15. after epoch 15 model started overfitting. (Next step) model can be more complex with regularization, model is forgetting information after 3rd RNN layer due to lower number of units, this needs to be solved.

10.3. Model 2, training using Conv1D, Bi-directial RNN, LSTMs and GRU layer

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 130, 1000)         180960000

 bidirectional_5 (Bidirectio  (None, 130, 128)         545280
 nal)

 bidirectional_6 (Bidirectio  (None, 130, 128)         98816
 nal)

 bidirectional_7 (Bidirectio  (None, 130, 128)         24704
 nal)

 conv1d_1 (Conv1D)           (None, 128, 72)           27720

 max_pooling1d_1 (MaxPooling  (None, 64, 72)           0
 1D)

 simple_rnn_6 (SimpleRNN)    (None, 64, 64)            8768

 gru_1 (GRU)                 (None, 64)                24960

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 42)                2730

=================================================================
Total params: 181,692,978
Trainable params: 181,692,978
Non-trainable params: 0
_________________________________________________________________
```

Figure-18: Sequential 2 summary

```
Epoch 14/15
925/925 [==============================] - 669s 723ms/step - loss:
0.6933 - accuracy: 0.8375 - val_loss: 2.4149 - val_accuracy: 0.4828
Epoch 15/15
925/925 [==============================] - 661s 715ms/step - loss:
0.6422 - accuracy: 0.8512 - val_loss: 2.3917 - val_accuracy: 0.5048
test loss and accuracy: 2.377286911010742 0.5080198049545288
```

Notes:

Experiment 4: params: max_words=total_words, out_dim=70, maxlen=100, epoch=15, in this case out train and test accuracy improved but model is again overfitting as train accuracy is at 74% while test accuray is at 49%. to avoid overfitting we need ot increase maxlen and output dimensions of vector and also adding CNN1D layer after 3rd RNN with gloablmax pooling with regularization should help, also GRU can be considered

Experiment 5: params: max_words=total_words, out_dim=100, maxlen=130, epoch=15, train accuracy=71%, test accuracy=49%, not any significant difference than previous model, (next up) add GRU, cnn1d with globalmaxpooling at the end, with shuffling of train data and run the notebook again

Experiment 6: params: max_words=total_words, out_dim=100, maxlen=130, epoch=15, train accuracy=85%, test accuracy=50%, we used LSTMs and GRU to process long sequences and retain previous inputs at perticuler input. while train accuracy improved significantly, test accuracy is still at 50%, a marginal improvement. Text data is always hard problem owing distribution on inputs. NLP models generally requires huge data to gain maximum accuracy on testing data. pre-trained models like BERT, Roberta can be used to overcome these challenges.

10.4. Learning curve of model 1 & model 2

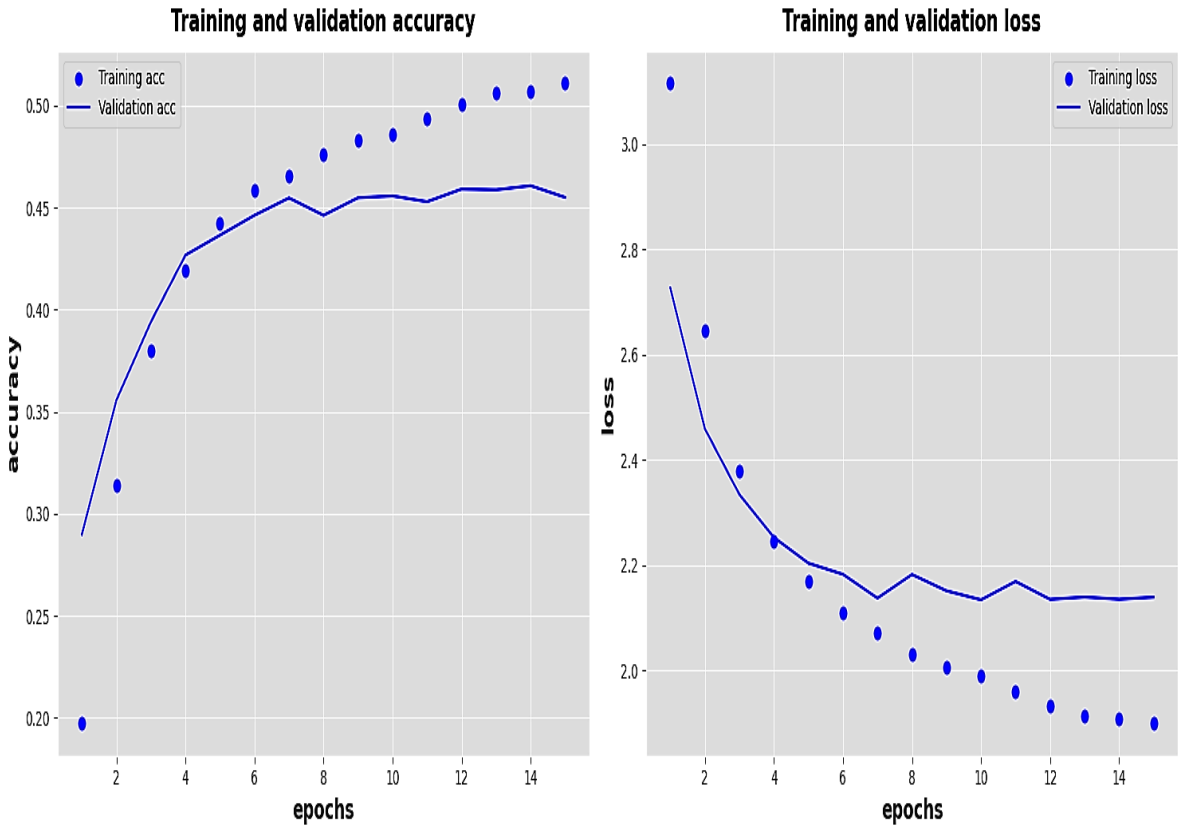


Figure-19: History of Model 01

Figure-20: History of Model 02

Chapter 05: Conclusion

In this notebook, we explored some of text data visualization techniques to derive insights out of text data and make use of them into model training.

We built first model using simpleRNN and embedding layer of keras where we found maximum of 49% accuracy on test data and also noticed foregtting of model due to large sequences of inputs.

In our second model, we trained model using LSTMs and GRU for retaining information of longer sequences. we could 'optimize' model as it improved accuracy of training data significantly but it could not 'generalize well' enough on unseen data.

In conclusion, this project involved the analysis of a news article dataset. The dataset was loaded and explored using the pandas library, providing insights into the shape of the data and the number of unique categories.

The top categories and their corresponding article counts were visualized using a bar plot, highlighting the distribution of articles across different categories. The analysis revealed the most prevalent categories in the dataset.

A baseline model using embedding layers and SimpleRNN was implemented for text classification. The model architecture was designed with bidirectional recurrent layers to capture the sequential nature of the text data. The model was compiled with appropriate loss and metrics functions.

The model was trained and evaluated using the provided input sequences and target labels. The training process included early stopping and model checkpoint callbacks to monitor and save the best performing model. The model's performance was evaluated on the test set using accuracy as the metric.

Throughout the project, the necessary preprocessing steps, including padding or truncating sequences, were applied to ensure compatibility between the input data and the model's requirements.

In conclusion, this project successfully analyzed the news article dataset, visualized the top categories, and implemented a baseline model for text classification. Further improvements and optimizations can be explored to enhance the model's performance, such as trying different architectures, tuning hyperparameters, and considering advanced techniques like pre-trained embeddings.

Overall, this project provides a solid foundation for further analysis and development in the field of text classification and demonstrates the potential for extracting valuable insights from news article data.

Chapter 06: References

1. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
2. Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
3. Zhang, Y., Marshall, I., & Wallace, B. C. (2016, November). Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing* (Vol. 2016, p. 795). NIH Public Access.
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
5. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1480-1489).
6. Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18* (pp. 194-206). Springer International Publishing.
7. https://www.kaggle.com/code/avikumart/nlp-news-articles-classif-wordembeddings-rnn#4.-Model-training-using-embedding-layer-and-RNN-(Baseline)
8. https://www.kaggle.com/code/vikashrajluhaniwal/recommending-news-articles-based-on-read-articles
9. https://www.kaggle.com/datasets/rmisra/news-category-dataset
10. https://app.diagrams.net