VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Technical Answers for Real World Problems

Submitted by:
Aparna Patra - 19BEE0069
Ekta Ray - 19BEE0060
Ayush Senapati -19BEE0171
Yash Sengupta - 19BEE0113

Project Guide: Prof.Venkatesh S.

<School of Electrical Engineering>

Fall Semester 2021-22

DECLARATION

We hereby declare that the report entitled **"Topic Modelling for Research Articles using Sequential Neural Networks. "** submitted by us, for the MGT1022 Technical Answer For Real World Problems (EPJ) to Vellore Institute of Technology is a record of bonafide work that has been carried out by us under the supervision of Prof. Venkatesh S.

We further declare that the work demonstrated in this report has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

APARNA PATRA

EKTA RAY

AYUSH SENAPATI

YASH SENGUPTA

PLACE:VELLORE INSTITUTE OF TECHNOLOGY

CONTENT

1. Abstract:

The aim of our project is to determine the category of any research paper by analyzing the abstract of the journal. Our approach is to train a model that uses text classification and deep learning to derive the category by analyzing just the abstract of the paper. The model will be trained from a clean dataset, removing missing values and using lemmatization. And it will use deep learning algorithms to finally classify the input text. Sequential Neural Networks will be used to carry out this task.

Our project aims to determine under what category a Research paper lies just by processing the abstract of the research paper. Among the tags include Mathematics, Statistics, Computers, Physics, applications of partial differential equations, Applications, AI, Astrophysics of Galaxies, computation and language, computer vision and pattern recognition, Cosmology, and Non- Galactic Astrophysics, Data Structures and Algorithms, Differential Geometry, Earth and planetary astrophysics. Fluid dynamics, Information Theory, Instrumentation and method for Astrophysics, Machine Learning and Material Science, Methodology, Number Theory, and much more. A research article can possibly have multiple tags. The research article abstracts are sourced from the following 4 topics:

- Computer Science
- Mathematics
- Physics
- Statistics.

Our model would read an abstract of our research paper as input and would return the category out of the four categories that particular research paper belongs to.

A website would be created that would accept an abstract from a user up to a word limit and generate the best suiting subcategory it belongs to along with the confidence score or the accuracy with which it was determined. One of the major possible applications of this project would be in the form of it being used for a ‘Digital Library’, wherein papers can be classified into specific and designated fields, thereby helping any researcher or user to access what they want to with ease and increase the degree of organization of the web resource.

This project takes a novel spin on the concept of text classification for research works as only a few works have worked in this field, with an application using sequential neural networks being a first.

2. Introduction

With the widespread rise in the levels of globalization as a direct result of the growth of the internet, accessibility to high quality content has improved considerably as well. Today, the only major barrier preventing the curiosity of any person being fulfilled is their curiosity alone as the resources required are available readily.

As a result of this unprecedented amount of information at the disposal of the beholder it is pivotal for the development of tools that assist the user. With the help of this project we aim to fulfill this much needed requirement as a user would be able to determine the category of any article or published work and determine whether or not the same is in line with their interest.

Additionally, this project would help researchers and aspirants in identifying the best journals or conferences they could proceed in as well. More often than not the decision making pertaining to the same is difficult for researchers, which is exactly wherein this project can come to the rescue. The recent rise in use of digital libraries further accentuate the significance of the potential of applications offered by this project.

This document aims to summarize and logically compose the progress made in this project and clearly explain aspects pertaining to the same using various techniques. This document would thereby serve as a documentation of the proposed architecture diagram, the dataset and the justification behind its use and the implementation carried out thus far, which includes the data cleaning.

3. Goals and Objectives

- Our project aims to determine the category of any research paper by analyzing the abstract of the journal.
- Training a model uses text classification and deep learning to derive the category by analyzing the abstract of the paper
- The main categories used to propose the domain of the research paper are :- Computer Science, Statistics, Physics and Mathematics and many more.
- A website will be created where users will be uploading the abstract upto a certain word limit and the model will generate the best suitable subcategory along with the accuracy with which it is categorized.

4. Literature Survey:

- ***ARTICLE-LEVEL CLASSIFICATION OF SCIENTIFIC PUBLICATIONS: A COMPARISON OF DEEP LEARNING, DIRECT CITATION AND BIBLIOGRAPHIC COUPLING-RIVEST, M., VIGNOLA-GAGNE, E., & ARCHAMBAULT, E. (2021).***

  This paper refers to the classification of scientific publications with the help of CNNs and other Deep Learning methods. The output of the same study is directly compared with existing methods including those of direct citation and bibliographic coupling. The key motivation behind this was firstly due to the need of classifying articles into sections and their wide variety of advantages and secondly as a result of the curiosity of exploring the effect a powerful technique like Deep Learning can have on such a field.The key benchmarks or the organization via which comparison and contrast is conducted for this is with the help of two key techniques: Herfindahl Index or citation concentration and Manual Article classification.It was seen that when DL was used it replicated more science metric classification than both BC and DC. A key reason for the same was the higher averaged macro F1 score and much higher in comparison macro precision. It was seen that when DL was used it replicated more science metric classification than both BC and DC. A key reason for the same was the higher averaged macro F1 score and much higher in comparison macro precision.

- ***A NOVEL REASONING MECHANISM FOR MULTI-LABEL TEXT CLASSIFICATION- WANG, R., RIDLEY, R., QU, W., & DAI, X. (2021).***

  The title of this paper directly reveals its true aim with its precise title. In a rather dense field of research when it comes to labeling of texts, this paper writes a new page in their book as they find and test a novel reasoning mechanism for multi label text classifications. By identifying this specific problem, the work was well cut out, and the implementation has been carried out with the aim of solving the same issue. The work carries out the development of a ‘ML Reasoner’ , a version of their own Binary Classifier which stands as the Multi-Label reasoner. Herein, every instance of the process can make use of the prior predictions as a label or as an additional input.. Further, as a result of multiple studies or

experiments carried out as a part of this work, it was found that the methodology used here was much better when compared to many other 'state-of-the-art' approaches when employed on the AAPD dataset. An application was also made by the researchers, for the reasoning on NN based models, which showed that with this, a boost in performance was regularly seen on subsequent iterations.

- ***SENTIMENT CLASSIFICATION AND ASPECT-BASED SENTIMENT ANALYSIS ON YELP REVIEWS USING DEEP LEARNING AND WORD EMBEDDINGS-ALAMOUDI, E. S., & ALGHAMDI, N. S. (2021).***

  The aim of this paper, as often with such studies, was realized in the gold mine that can lie in the reviews of users, Yelp in this case, and how they can help the business in multiple ways, only if analyzed properly. Which is where, NLP and its techniques, as always come into the picture. A key statement is made how the process of Opinion Mining has heavily supported aspects related to decision making. As a part of this study, the reviews of Restaurants found in the reviews were divided into two distinct classifications, as per their negative or positive sentiments, and as per their ternary classifications, which included an additional field of 'neutral'. For studying the same, the use of three predictive models has been done. This includes the applications of ML or Machine Learning, DL or Decision Learning and Transfer Learning. Out of the various models then studied, it was found that the ALBERT model produced the highest accuracy at an extremely high 98.30%. Further, the main technique employed in the paper was instrumental with an acceptable accuracy of 83.04%.

- **SEQUENTIAL THREE-WAY DECISION AND GRANULATION FOR COSTSENSITIVE FACE RECOGNITION-ZHANG, L., LI, H., ZHOU, X., & HUANG, B. (2020)**

  A Sequential Three-Way Decision (S3WD) model was created to adjust the misclassification cost and the time cost in autoencoder based characterizations and decisions. To execute the tradeoff methodology, it was important to extricate a multi-granular list of capabilities. Thus, for fulfilling the same, in the organization, the related discriminative data in the separated highlights increments with preparing epochs, which builds a multi-granular component structure. An autoencoder-based multi-granular highlight portrayal definition was employed. At last, the analyses exhibit the adequacy of the proposed approaches.An autoencoder network, created of stacked RBMs, was embraced to remove multi-granular highlights for dynamic 3WD models. The removed elements had the same aspect, which was not the same as the customary meaning of sequential elements. Through trial testing, the creators found a well-performing POS tagger for building codes that had one bidirectional LSTM teachable layer, used BERT_Cased_Base pre-prepared model and 50 epochs. This model arrived at a 91.89% accuracy without mistake driven groundbreaking principles and a 95.11% accuracy with blunder driven groundbreaking standards, which beat the 89.82% accuracy accomplished by the best in class POS taggers.

- **PART-OF-SPEECH TAGGING OF BUILDING CODES EMPOWERED BY DEEP LEARNING AND TRANSFORMATIONAL RULES-XUE, X., & ZHANG, J. (2021):**

  The creators proposed another POS tagger custom fitted to building codes. This uses profound learning NN models and error driven groundbreaking standards. The neural organization model contains a pre-prepared model and at least one teachable neural layer. The pre-prepared model was calibrated on Part-ofSpeech Tagged Building Codes (PTBC), a POS labeled building codes dataset. The calibrating of the pre-prepared model permits the proposed POS tagger to arrive at high accuracy with a limited quantity of accessible preparing data. Blunder driven groundbreaking principles were utilized to support execution further by fixing mistakes made by the neural organization model in the labeled building code. This model arrived at a 91.89% accuracy without mistake driven groundbreaking principles and a 95.11% accuracy with blunder driven groundbreaking standards, which beat the 89.82% accuracy accomplished by the best in class POS taggers.

SUMMARY TABLE:

Title of the Paper	Authors and Year of Publication	Overview of Paper
Article-level classification of scientific publications: A comparison of deep learning, direct citation and bibliographic coupling.	**Rivest, M., Vignola-Gagne, E., & Archambault, E. (2021).**	**Autoencoder network is a proficient portrayal learning strategy. As a rule, a better include set acquired from autoencoder prompts a lower blunder rate and lower absolute misclassification cost. Notwithstanding, the organization is typically prepared for quite a while to acquire a better include set, prompting a high time cost and complete expense. To resolve this issue, a Sequential Three-Way Decision (S3WD) model was created to adjust the misclassification cost and the time cost in autoencoder based characterizations and decisions.**
A novel reasoning mechanism for multi-label text classification. Information Processing & Management	**Wang, R., Ridley, R., Qu, W., & Dai, X. (2021)**	**This uses profound learning NN models and error driven groundbreaking standards. The neural organization model contains a pre-prepared model and at least one teachable neural layer. The pre-prepared model was calibrated on Part-of-Speech Tagged Building Codes (PTBC), a POS labeled building codes dataset.**

Topic Modeling Meets Deep Neural Networks: A Survey.	**Alamoudi, E. S., & Alghamdi, N. S. (2021)**	**A variety of NTMs based on different frameworks have been developed and the conclusions that were drawn out are, better evaluation across a unified system of evaluation metrics, making the comparisons across different NTMs. Compared to BPTMs, NTMs offer better flexibility for representing topic distributions for documents and word distributions for topics. Pre-trained language models provide more advanced, and higher-level representations of semantic knowledge, which can be leveraged in NTMs to boost performance.**
Sequential three-way decision based on multi granular autoencoder features	**Zhang, L., Li, H., Zhou, X., & Huang, B. (2020).**	**It tests a novel reasoning mechanism for multi label text classifications. The relevance of the same is very well justified as the work directly indicated how a strong technique such as labeling, which on its own can be used in across applications can often get too dependent on the order of the labels.**
Part-of-speech tagging of building codes empowered by deep learning and transformational rules.	**Xue, X., & Zhang, J. (2021).**	**This paper refers to the classification of sentiment from restaurant reviews using ML and DL, the reviews were divided into three classifications i.e positive, negative and neutral sentiments.**

5. METHODOLOGY:

- Apply various data visualisation tools to perfectly understand which classification algorithm/neural network would perfectly fit our problem
- Create a bag of all words in our dataset
- See how many times a particular word is present in our sentence
- Create a sequential neural network and train it on the dataset provided
- Use the trained model to classify sentences(abstracts) into a particular category of research - computer science, mathematics, physics, statistics with Bag of Words and Related approach.

5.1 Architecture Diagram

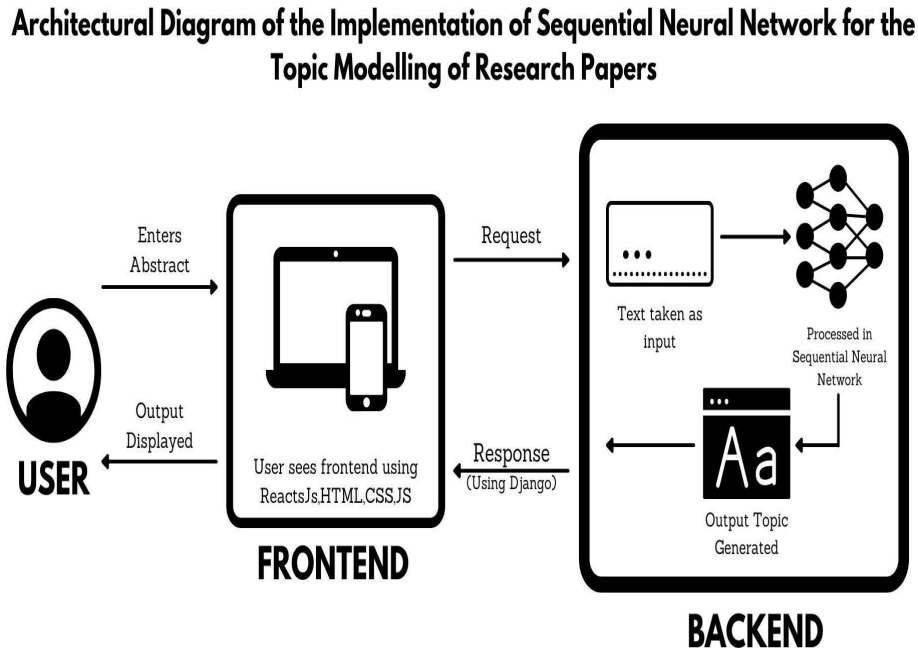


Fig 1 :Architecture Diagram

5.2 Flow Diagram

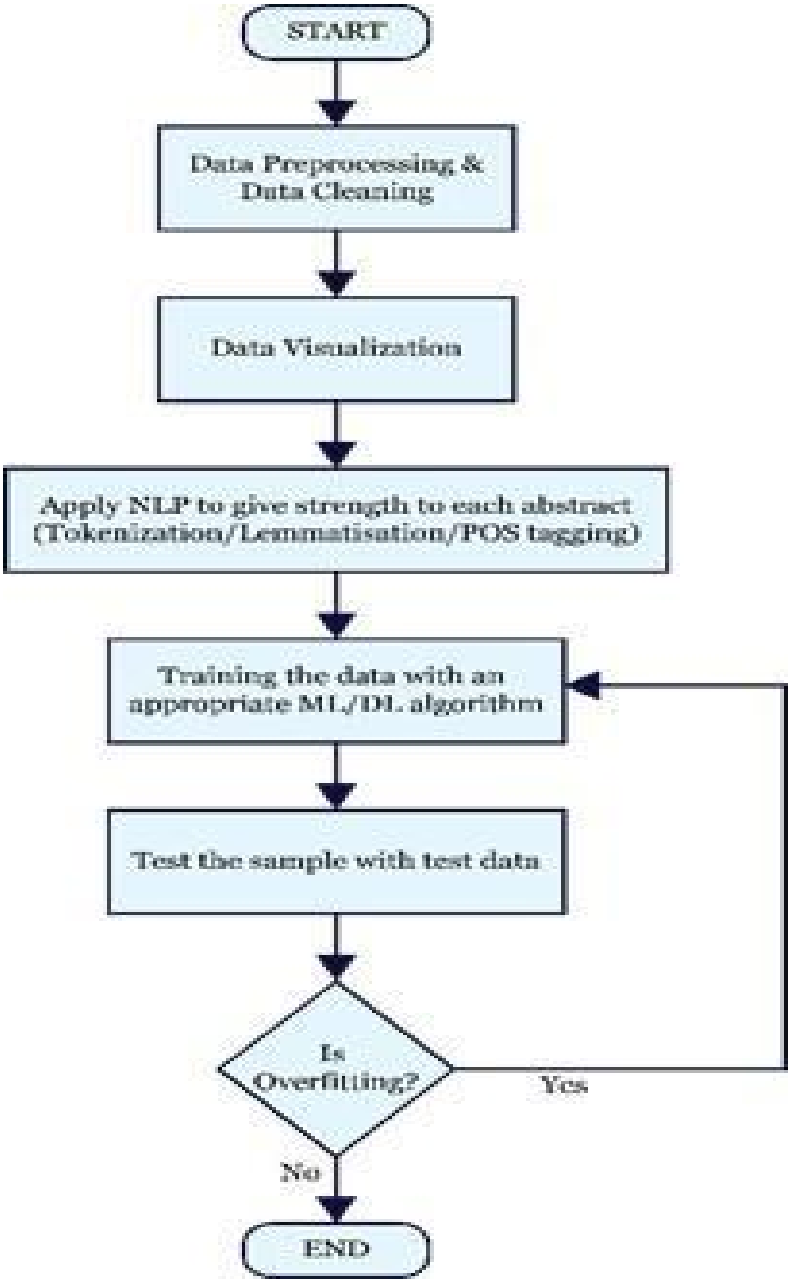


Fig 2: Flow Diagram

5.3 PseudoCode

```
BEGIN:
    READ Import Necessary Packages //As listed below
    READ: Import Train.csv and Test.csv //using pandas

    //Cleaning Dataset
    DETERMINE Db ← Drop Columns 6 to 31 in Db
    DETERMINE Db ← Drop ID Column in Db
    SET blanks=[]
    FOR each column in Db
    IF Abstract Column is Blank ADD blanks ← blanks[i]
        END IF
    END FOR
    DISPLAY blanks
    IF blanks is empty
        CONTINUE
    ELSE
        DETERMINE Db ← Drop Columns with index i
    END IF
    SET Db[‘label] ← 0
    FOR each row in
    Db
        IF Row[‘Mathematics’]==1 and Row[‘Other Columns’]==0
            ADD Db[‘label’] ← M
        ELSE IF Row[‘Computer Science’’]==1 and Row[‘Other Columns’]==0
            ADD Db[‘label’] ← C
        ELSE IF Row[‘Physics’’]==1 and Row[‘Other Columns’]==0
            ADD Db[‘label’] ← P
        ELSE
            ADD Db[‘label’] ← S
        END IF
    END FOR

    //Tokenization
    DETERMINE tokens ← nltk.word_tokenizer(row[‘ABSTRACT’])
```

```
FOR w for w in tokens
        IF w.isalpha()
                ADD token_words ← token_words[w]
        END IF
END FOR
DETERMINE db[‘abstract’] ← db.apply[identify_tokens]

//Stemming
READ PorterStemmer from nltk.stem()
COMPUTE stemming ← PorterStemmer
COMPUTE my_list ← row[‘ABSTRACT’]
FOR stemming.stem(word) for word in my_list
        ADD stemmed_list ← stemmed_list[word]
END FOR
DETERMINE db[‘stemmed_words’] ← db.apply[stem_list]

//Removing Stop Words
READ stopwords from nltk.corpus
COMPUTE stops ← stopwords.words[“english”]
COMPUTE my_list ← row[‘stemmed_words’]
FOR w for word in my_list
        IF not w in stops
                ADD meaningful_words ← meaningful_wordst[w]
        END IF
END FOR
DETERMINE db[‘stem_meaningful’’] ← db.apply[remove_stops]

//Brief Pseudocode for Proposed Implementation ahead

//Using Created Tokens to create a new vocabulary of all
words READ tokenizer and pad_sequences from keras
COMPUTE vocab_size using Tokenizer
//Perform One-Hot Encoding for each of obtained tokens using the vocabulary list
READ Sequential from Tensorflow
DETERMINE Sequential model with parameters listed below
```

//Use the keras library provided by tensorflow to build the sequential model and optimize the parameters of Embedding, Flatten and Dense

//Use the encoded training data such that the sequential model can train itself

//Implement the word embeddings approach and tune the dense layer weights

DETERMINE Model Summary and Fit the Model

//Optimize all the parameters to optimize the model and its accuracy

DETERMINE F-measure of the Model

//Test the accuracy of the model using F-measure and ensure low presence of false positives and false negatives

END

6. OBSERVATION AND ANALYSIS

Dataset Used and Requirement Analysis:

We have found the dataset on Kaggle. It is in the form of an excel sheet with one column containing the research article abstracts and the other column containing the tags. Firstly, we will clean the data and pre-process the data (removing the missing data, etc). Then we will apply many NLP methods in the Abstracts column including tokenization, lemmatization, etc, to give each abstract strength. Then we will use Deep Learning and Machine Learning algorithms to classify the abstract based on the tags and the categories.

I. Functional requirement:

Our model takes in the abstract of a research paper as an input and lets us know in which category the research paper lies.

II. Software required:

1. Google Collab
2. Jupyter notebook
3. VS Code

III. Packages required:

1. Nltk

2. Flask
3. Matplotlib
4. Pandas
5. Numpy
6. Seaborn
7. Scikit learn
8. Tensorflow
9. Keras
10. Spacey

6.1 Sample of the Datasets

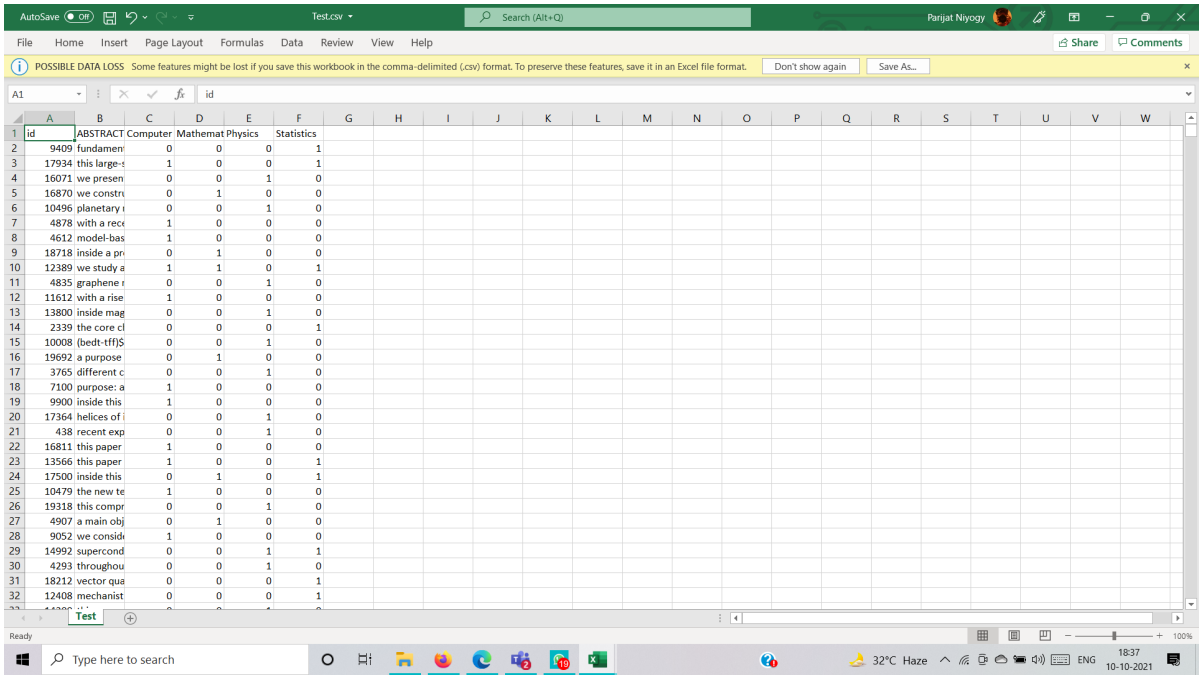
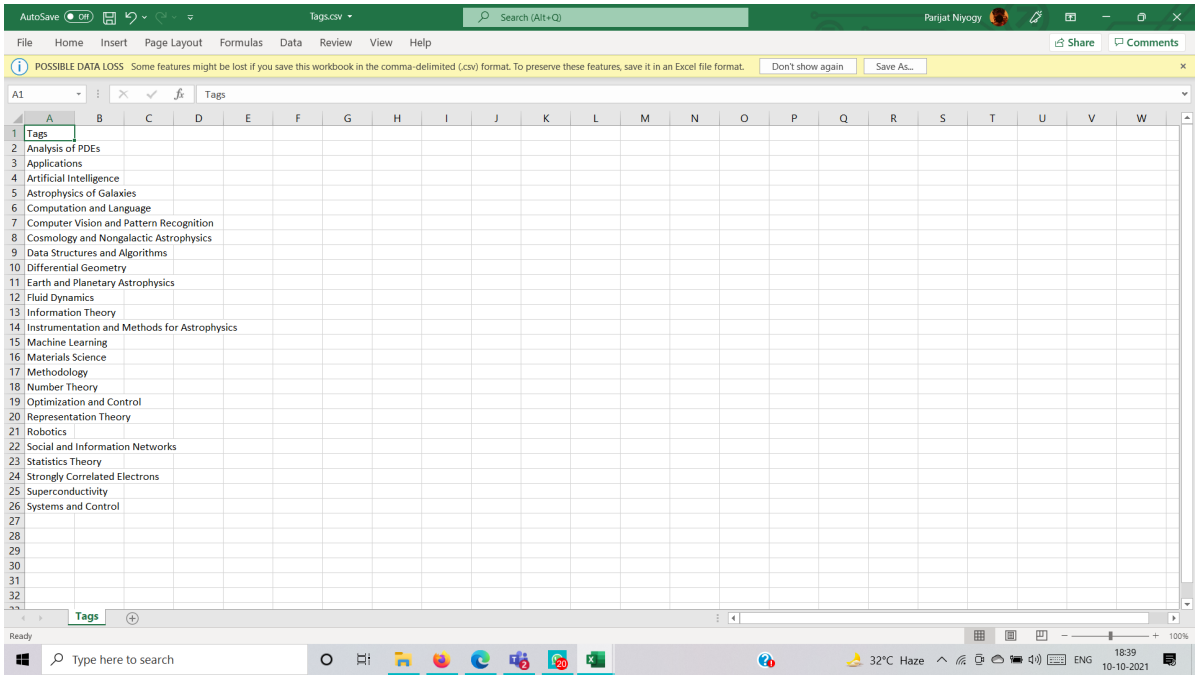
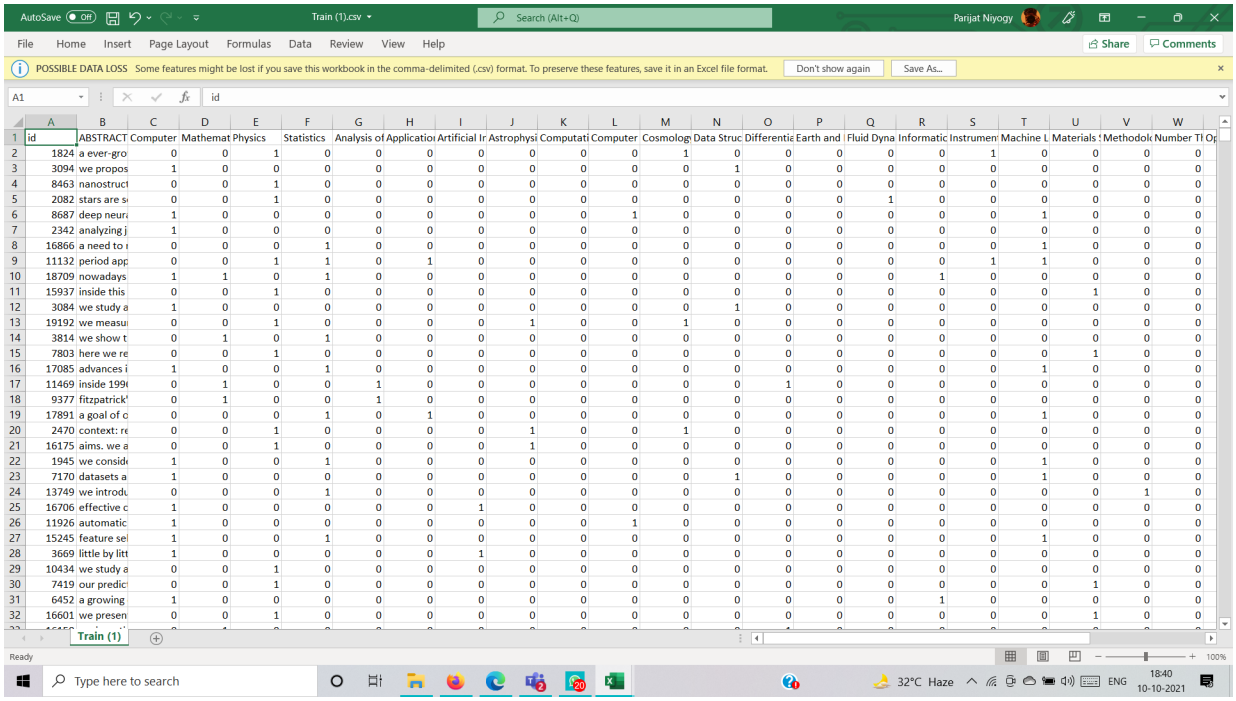


Fig 3 : Test.csv : Sample data used for testing the data with the trained model



Tags
Analysis of PDEs
Applications
Artificial Intelligence
Astrophysics of Galaxies
Computation and Language
Computer Vision and Pattern Recognition
Cosmology and Nongalactic Astrophysics
Data Structures and Algorithms
Differential Geometry
Earth and Planetary Astrophysics
Fluid Dynamics
Information Theory
Instrumentation and Methods for Astrophysics
Machine Learning
Materials Science
Methodology
Number Theory
Optimization and Control
Representation Theory
Robotics
Social and Information Networks
Statistics Theory
Strongly Correlated Electrons
Superconductivity
Systems and Control

Fig 4 : Tags.csv: For displaying the planned and available sample tags



id	ABSTRACT	Computer	Mathemat	Physics	Statistics	Analysis of	Applicatio	Artificial Ir	Astrophysi	Computati	Computer	Cosmolog	Data Struc	Differentia	Earth and	Fluid Dyna	Informatic	Instrumen	Machine L	Materials	Methodol	Number Th
1824	a ever-gro	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
3094	we propos	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8463	nanostruct	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2082	stars are s	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
8687	deep neura	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
2342	analyzing j	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16866	a need to r	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11132	period app	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
18709	nowadays	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
15937	inside this	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3084	we study a	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
19192	we measu	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
3814	we show t	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7803	here we re	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17085	advances i	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11469	inside 199	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9377	fitzpatrick'	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17891	a goal of c	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2470	context: re	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
16175	aims. we a	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1945	we conside	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7170	datasets a	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
13749	we introdu	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
16706	effective c	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11926	automatic	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15245	feature sel	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3669	little by litt	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10434	we study a	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7419	our predic	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6452	a growing	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
16601	we presen	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

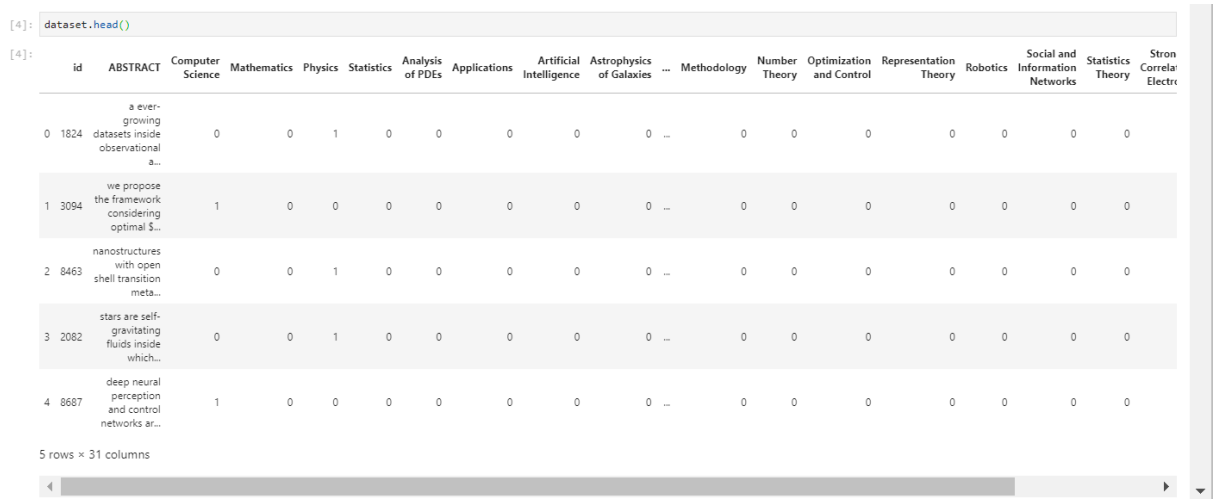
Fig 5 : Train.csv: Dataset used for training the model. The dataset underwent extensive cleaning methods before being used

6.2 Output

Cleaning the Dataset

The training Dataset has been cleaned using the techniques discussed in the pseudocode listed in detail above

Displaying the head, or the first few rows of the dataset:



```
[4]: dataset.head()
```

[4]:

	id	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	Analysis of PDEs	Applications	Artificial Intelligence	Astrophysics of Galaxies	...	Methodology	Number Theory	Optimization and Control	Representation Theory	Robotics	Social and Information Networks	Statistics Theory	Stron Correla Electr
0	1824	a ever-growing datasets inside observational a...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
1	3094	we propose the framework considering optimal $...	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
2	8463	nanostructures with open shell transition meta...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
3	2082	stars are self-gravitating fluids inside which...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
4	8687	deep neural perception and control networks ar...	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	

5 rows × 31 columns

Dropping the Columns such that 4 columns can be used for analysis:

```
[18]: db = dataset.drop(['id'], axis = 1)
      db=db.drop(dataset.iloc[:, 6:31],
                           axis = 1)
      db
```

[18]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics
0	a ever-growing datasets inside observational a...	0	0	1	0
1	we propose the framework considering optimal $...	1	0	0	0
2	nanostructures with open shell transition meta...	0	0	1	0
3	stars are self-gravitating fluids inside which...	0	0	1	0
4	deep neural perception and control networks ar...	1	0	0	0
...	...	...	...	...	...
13999	a methodology of automatic detection of a even...	1	0	0	0
14000	we consider a case inside which the robot has ...	1	0	0	0
14001	despite being usually considered two competing...	0	0	1	0
14002	we present the framework and its implementatio...	1	0	0	0
14003	here we report small-angle neutron scattering ...	0	0	1	0

14004 rows × 5 columns

Checking for null values present in the ABSTRACT column with the help of a blanks[] array

```
[20]: blanks=[]
      for i, ab, c,m,p,s in db.itertuples():
          if(ab.isspace()):
              blanks.append(i)
      # for index, row in dataset.iterrows():
      #     if(dataset.isnull(row['myCol'])):
      #         blanks.append(row['my'])
```

```
[21]: blanks
```

[21]: []

Adding a column of labels and filling it with ‘0’

```
[22]: db['label']='0'
```

```
[23]: db.head()
```

[23]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	label
0	a ever-growing datasets inside observational a...	0	0	1	0	0
1	we propose the framework considering optimal $...	1	0	0	0	0
2	nanostructures with open shell transition meta...	0	0	1	0	0
3	stars are self-gravitating fluids inside which...	0	0	1	0	0
4	deep neural perception and control networks ar...	1	0	0	0	0

Tokenization Implementation

```
In [37]: for index, row in dataset.iterrows():
#     print(row['Mathematics'], row['Statistics'])
      if(row['Mathematics'] == 1 and row['Computer Science'] == 0 and row['Physics']==0 and row['Statistics'] == 0):
        dataset['label'] = 'M'
      elif(row['Computer Science'] == 1 and row['Mathematics'] == 0 and row['Physics']==0 and row['Statistics'] == 0):
        dataset['label'] = 'C'
      elif(row['Physics'] == 1 and row['Computer Science'] == 0 and row['Mathematics']==0 and row['Statistics'] == 0):
        dataset['label'] = 'P'
      else:
        dataset['label'] = 'S'
```

```
In [82]: ##Applying Tokenization to all the rows in our dataset
def identify_tokens(row):
    abstract = row['ABSTRACT']
    tokens = nltk.word_tokenize(abstract)
    # taken only words (not punctuation)
    token_words = [w for w in tokens if w.isalpha()]
    return token_words
```

```
In [83]: dataset['ABSTRACT'] = dataset.apply(identify_tokens, axis=1)
```

```
In [84]: dataset.head(20)
```

Out[84]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P
12	[we, show, that, an, embedding, inside, euclid...	0	1	0	1	M
13	[here, we, report, a, measurement, of, a, inte...	0	0	1	0	P
14	[advances, inside, a, field, of, inverse, rein...	1	0	0	1	C
15	[inside, proved, that, every, riemannian, mani...	0	1	0	0	M

Implementing Stemming

In [86]:
```
## Stemming our date in the abstract field
from nltk.stem import PorterStemmer
stemming = PorterStemmer()

def stem_list(row):
    my_list = row['ABSTRACT']
    stemmed_list = [stemming.stem(word) for word in my_list]
    return (stemmed_list)

dataset['stemmed_words'] = dataset.apply(stem_list, axis=1)
```

In [87]:
```
dataset.head(20)
```

Out[87]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels	stemmed_words
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P	[a, dataset, insid, observ, astronomi, have, c...
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C	[we, propos, the, framework, consid, optim, t,...
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P	[nanostructur, with, open, shell, transit, met...
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P	[star, are, fluid, insid, which, pressur, buoy...
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C	[deep, neural, percept, and, control, network,...
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C	[analyz, job, hop, behavior, wa, import, consi...
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S	[a, need, to, reason, about, uncertainti, insi...
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P	[period, approxim, wa, one, of, a, central, to...
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M	[nowaday, data, compressor, are, appli, to, ma...
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P	[insid, thi, work, the, potenti, of, nb, consi...
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C	[we, studi, a, problem, of, extract, the, sele...
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P	[we, measur, a, stellar, mass, function, smf, ...
12	[we, show, that, an, embedding, inside, euclid...	0	1	0	1	M	[we, show, that, an, embed, insid, euclidean, ...

Removing the Stop Words

In [88]:
```
# Removing stopwords
from nltk.corpus import stopwords
stops = set(stopwords.words("english"))

def remove_stops(row):
    my_list = row['stemmed_words']
    meaningful_words = [w for w in my_list if not w in stops]
    return (meaningful_words)

dataset['stem_meaningful'] = dataset.apply(remove_stops, axis=1)
```

In [89]:
```
dataset.head(20)
```

Out[89]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels	stemmed_words	stem_meaningful
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P	[a, dataset, insid, observ, astronomi, have, c...	[dataset, insid, observ, astronomi, challeng, ...
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C	[we, propos, the, framework, consid, optim, t,...	[propos, framework, consid, optim, exclud, pre...
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P	[nanostructur, with, open, shell, transit, met...	[nanostructur, open, shell, transit, metal, mo...
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P	[star, are, fluid, insid, which, pressur, buoy...	[star, fluid, insid, pressur, buoyanc, rotat, ...
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C	[deep, neural, percept, and, control, network,...	[deep, neural, percept, control, network, like...
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C	[analyz, job, hop, behavior, wa, import, consi...	[analyz, job, hop, behavior, wa, import, consi...
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S	[a, need, to, reason, about, uncertainti, insi...	[need, reason, uncertainti, insid, larg, compl...
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P	[period, approxim, wa, one, of, a, central, to...	[period, approxim, wa, one, central, topic, in...
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M	[nowaday, data, compressor, are, appli, to, ma...	[nowaday, data, compressor, appli, mani, probl...
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P	[insid, thi, work, the, potenti, of, nb, consi...	[insid, thi, work, potenti, nb, consid, radiat...
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C	[we, studi, a, problem, of, extract, the, sele...	[studi, problem, extract, select, connector, c...
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P	[we, measur, a, stellar, mass, function, smf, ...	[measur, stellar, mass, function, smf, galaxi,...

Sequential Neural Network

Word embeddings approach: Using NN

In [16]:
```
train['text'] = ' '
test['text'] = ' '

#this is our corpus basically
train['text'] += train['ABSTRACT']
test['text'] += test['ABSTRACT']

trn, val = train_test_split(train, test_size=0.2, random_state=2)
```

In [20]:
```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

#100000 is the max. no. of words to keep in the tokenized list
tok = Tokenizer(num_words = 1000000)
tok.fit_on_texts(train['text'].str.lower().tolist() + test['text'].str.lower().tolist())

vocab_size = len(tok.word_index) + 1
vocab_size
```

Out[20]: 51665

In [18]:
```
X_trn = tok.texts_to_sequences(trn['text'])
X_val = tok.texts_to_sequences(val['text'])
X_test = tok.texts_to_sequences(test['text'])

```

Embedding the model and using back propagation to learn the model

In [19]:
```
maxlen = 200 #maximum length of all sequences(i.e, to what length is each sentence padded upto)
X_trn = pad_sequences(X_trn, maxlen=maxlen)
X_val = pad_sequences(X_val, maxlen=maxlen)
X_test = pad_sequences(X_test, maxlen=maxlen)

X_test
```

Out[19]:
```
array([[    0,     0,     0, ...,   280,   965,    53],
       [    0,     0,     0, ...,  1278,   423,  4957],
       [    1, 10832,    75, ...,     5,  9884,  4154],
       ...,
       [   36,  1514,    10, ...,    99,   264,  2804],
       [    2,  7933,    22, ...,    62,   123,   125],
       [    0,     0,     0, ...,   412,  6056,   164]])
```

In [24]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Flatten, Dense, Dropout, SpatialDropout1D, LSTM


embedding_dim = 50 # taken 50 'features'
vocab_size = len(tok.word_index) + 1

model = Sequential()
model.add(Embedding(input_dim=vocab_size,
                    output_dim=embedding_dim,
                    input_length=maxlen))

model.add(Flatten())
model.add(Dense(200, activation='relu', name = 'Fully_Connected'))
model.add(Dense(25, activation='sigmoid', name = 'Output'))
model.compile(optimizer=tf.keras.optimizers.Adam(lr = 1e-3),
              loss='binary_crossentropy',
              metrics=['accuracy'],
              )

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 200, 50)           2583250
_________________________________________________________________
flatten (Flatten)            (None, 10000)             0
_________________________________________________________________
Fully_Connected (Dense)      (None, 200)               2000200
_________________________________________________________________
Output (Dense)               (None, 25)                5025
=================================================================
Total params: 4,588,475
Trainable params: 4,588,475
Non-trainable params: 0
_________________________________________________________________
```

Fitting the model

[26]:
```
model.fit(X_trn, trn[TARGET_COLS], validation_data=(X_val, val[TARGET_COLS]), verbose=True, epochs=20, batch_size=256,
          callbacks = [tf.keras.callbacks.ReduceLROnPlateau()])
```

```
Epoch 1/20
44/44 [==============================] - 7s 122ms/step - loss: 0.3871 - accuracy: 0.0883 - val_loss: 0.1925 - val_accuracy: 0.1
685
Epoch 2/20
44/44 [==============================] - 6s 129ms/step - loss: 0.1850 - accuracy: 0.1861 - val_loss: 0.1701 - val_accuracy: 0.2
363
Epoch 3/20
44/44 [==============================] - 4s 101ms/step - loss: 0.1540 - accuracy: 0.2897 - val_loss: 0.1495 - val_accuracy: 0.3
013
Epoch 4/20
44/44 [==============================] - 4s 96ms/step - loss: 0.1248 - accuracy: 0.4494 - val_loss: 0.1348 - val_accuracy: 0.35
27
Epoch 5/20
44/44 [==============================] - 4s 94ms/step - loss: 0.0958 - accuracy: 0.6092 - val_loss: 0.1228 - val_accuracy: 0.43
66
Epoch 6/20
```

Calculating the precision,recall and F1 values

In [27]:
```
import numpy as np
def get_best_thresholds(true, preds):
  thresholds = [i/100 for i in range(100)]
  best_thresholds = []
  for idx in range(25):
    f1_scores = [f1_score(true[:, idx], (preds[:, idx] > thresh) * 1) for thresh in thresholds]
    best_thresh = thresholds[np.argmax(f1_scores)]
    best_thresholds.append(best_thresh)
  return best_thresholds

val_preds = model.predict(X_val)
best_thresholds = get_best_thresholds(val[TARGET_COLS].values, val_preds)
for i, thresh in enumerate(best_thresholds):
  val_preds[:, i] = (val_preds[:, i] > thresh) * 1
f1_score(val[TARGET_COLS], val_preds, average='micro')


```

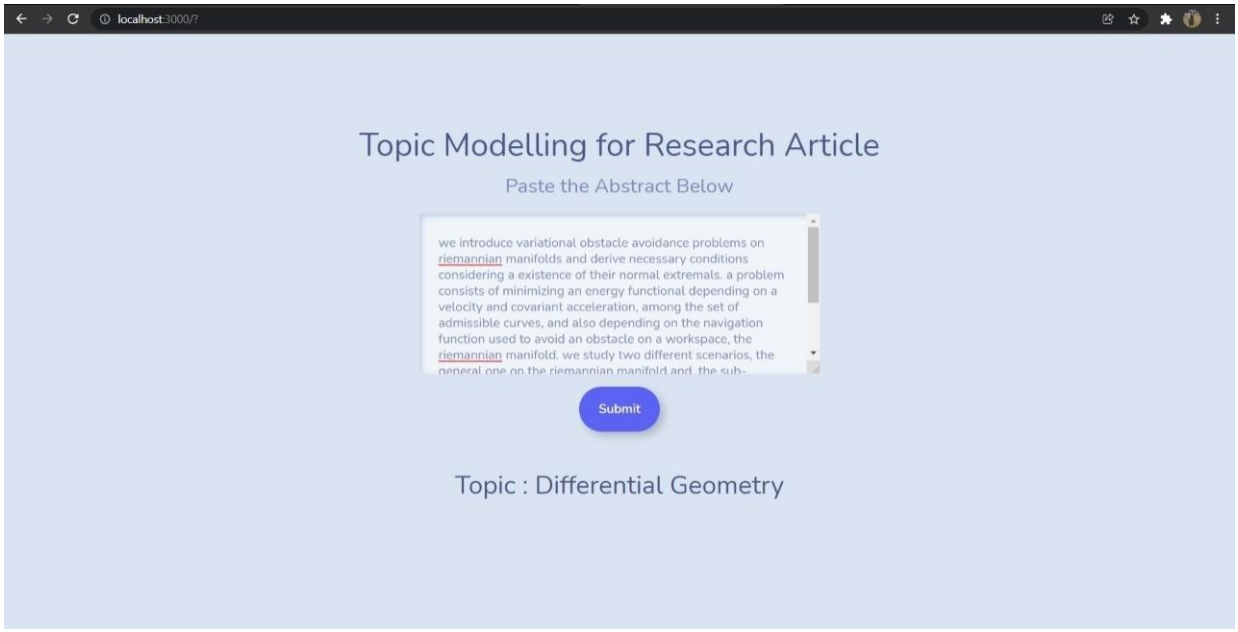
Out[27]:
```
0.5850256912160509
```

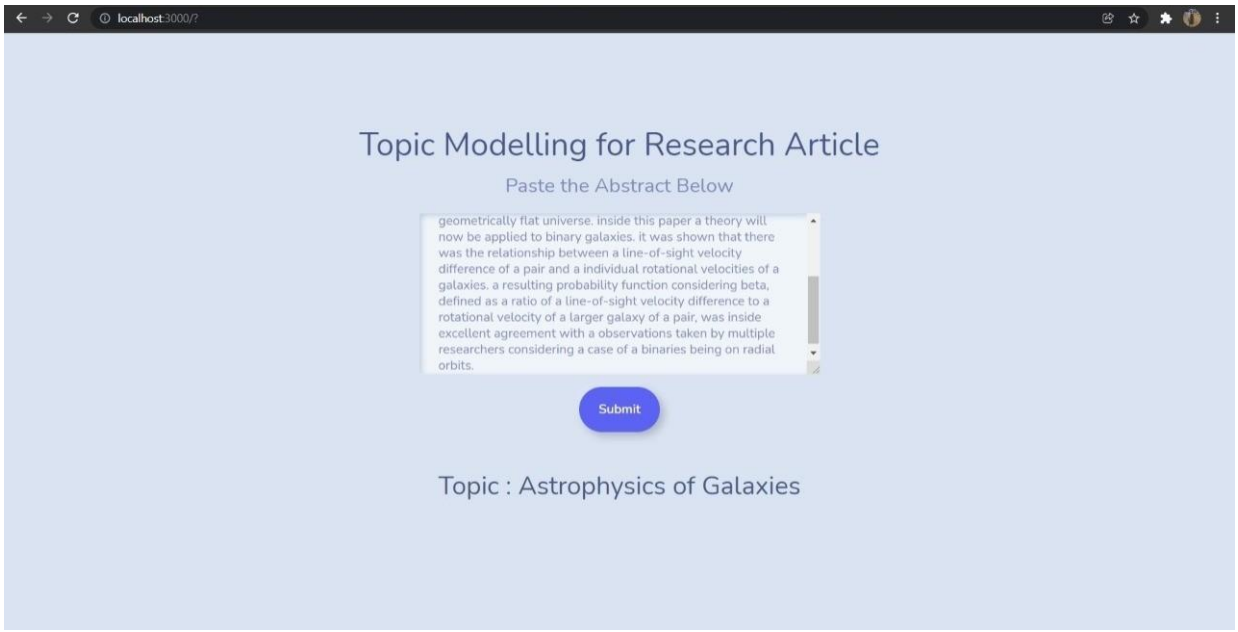
f1 score = 2*((precision*recall)/(precision+recall))
recall(True Positive Rate): When it's actually yes, how often does it predict yes?
precision:When it predicts yes, how often is it correct?

So our's is a good f1 score of around 0.6, so we have low false positives and low false negatives in our predictions

Instead of using accuracy(Overall, how often is the classifier correct?) ,
f1 score can help us judge the real-life applicability of our model.

6.3 Sample Output screen




localhost:3000/?
Topic Modelling for Research Article
Paste the Abstract Below
geometrically flat universe. inside this paper a theory will now be applied to binary galaxies. it was shown that there was the relationship between a line-of-sight velocity difference of a pair and a individual rotational velocities of a galaxies. a resulting probability function considering beta, defined as a ratio of a line-of-sight velocity difference to a rotational velocity of a larger galaxy of a pair, was inside excellent agreement with a observations taken by multiple researchers considering a case of a binaries being on radial orbits.
Submit
Topic : Astrophysics of Galaxies


localhost:3000/?
Topic Modelling for Research Article
Paste the Abstract Below
learning nonlinear dynamics from diffusion data was the challenging problem since a individuals observed may be different at different time points, generally following an aggregate behaviour. existing work cannot handle a tasks well since they model such dynamics either directly on observations or enforce a availability of complete longitudinal individual-level trajectories. however, inside most of a practical applications, these requirements are unrealistic: a evolving dynamics may be too complex to be
Submit
Topic : Machine Learning

7. Conclusion

1. The crucial features of our project are meant to solve problems of digital annotation in research libraries. In addition to these we aim to address all the possible application use cases that have been discussed over the course of this document.
2. First we had completed the vital task of data cleaning which was done using multifarious strategies that have been explained in detail. This includes the use of stemming and the removal of stop words.
3. For the implementation of our project we have developed and implemented the sequential model using Sequential Neural Network. We have done the same using tensorflow. One of the key reasons that we have used Neural networks over any other Machine Learning model is due to the tremendous help provided in accuracy in prediction by the latter.
4. At the same time have also explored other efficient strategies such as the BERT Model and have also successfully implemented the same. We have thoroughly discussed the theoretical aspects of the same and analyzed the role played by one-hot encoding and the mathematical working of the Word Embeddings approach. A detailed comparison of the same has been provided.
5. Further, we have deployed the sequential model using Flask and have also used frameworks such as ReactJS to prepare a frontend for the implementation of the project.
6. With these data and model foundation, a number of future works can be done for further research and experiment. As the limitation of the computation power, this project is based on a relatively small sample by the time we start writing. However, the result is quite convincing even with the small size. Applying to a larger dataset will more likely achieve better results.
7. Further, an application on topic modeling to manage, search and explore the abstracts of the research paper offline can be implemented. This application will not be limited to only research paper abstracts, but may be used beyond for a variety of corpus available. Also we could use this approach to classify news articles to find relevant news updates.

.

8. References and Citation:

[1] Rivest, M., Vignola-Gagne, E., & Archambault, E. (2021). Article-level classification of scientific publications: A comparison of deep learning, direct citation and bibliographic coupling. *PloS one*, *16*(5).

[2] Wang, R., Ridley, R., Qu, W., & Dai, X. (2021). A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, *58*(2), 102441.

[3] Alamoudi, E. S., & Alghamdi, N. S. (2021). Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. *Journal of Decision Systems*, 1-23.

[4] Li, H., Zhang, L., Huang, B., & Zhou, X. (2016). Sequential three-way decision and granulation for cost-sensitive face recognition. *Knowledge-Based Systems*, *91*, 241-251.

[5] Xue, X., & Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*, 101235.

[6] Chang, H. S., Agrawal, A., McCallum, A. (2021). Extending multi-sense word embedding to phrases and sentences for unsupervised semantic applications. *arXiv preprint arXiv:2103.15330*.

[7] Boyer, C., Dolamic, L., & Falquet, G. (2015). Language independent tokenization vs. stemming in automated detection of health websites' HONcode conformity: An Evaluation. *Procedia Computer Science*, *64*, 224-231.

[8] Watanabe, W. M., Felizardo, K. R., Candido Jr, A., de Souza,É. F., de Campos Neto, J. E., & Vijaykumar, N. L. (2020). Reducing efforts of software engineering systematic literature reviews updates using text classification. *Information and Software Technology*, *128*, 106395.

[9] Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2020). An evaluation of document clustering and topic modeling in two online social networks: Twitter and Reddit. *Information Processing & Management*, *57*(2), 102034.

[10] Martinez, C., Ramasso, E., Perrin, G., & Rombaut, M. (2020). Adaptive early classification of temporal sequences using deep reinforcement learning. *Knowledge-Based Systems*, *190*, 105290.

[11] Sokolowska, M., Mazurek, M., Majer, M., & Podpora, M. (2019). Classification of user attitudes in Twitter-beginners guide to selected Machine Learning libraries. *IFAC-PapersOnLine*, *52*(27), 394-399.

[12] Chowanda, A., Sutoyo, R., & Tanachutiwat, S. (2021). Exploring text-based emotions recognition machine learning techniques on social media conversation. *Procedia Computer Science*, *179*, 821-828.

[13] Dalal, M. K., & Zaveri, M. A. (2014). Opinion mining from online user reviews using fuzzy linguistic hedges. *Applied computational intelligence and soft computing*, *2014*.

[14] Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., & Buntine, W. (2021). Topic Modeling Meets Deep Neural Networks: A Survey. *arXiv preprint arXiv:2103.00498*.

[15] Oriola, O. Exploring N-gram, Word Embedding and Topic Models for Content-based Fake News Detection in FakeNewsNet Evaluation. *International Journal of Computer Applications*, *975*, 8887.

[16] Roman, M., Shahid, A., Uddin, M. I., Hua, Q., & Maqsood, S. (2021). Exploiting Contextual Word Embedding of Authorship and Title of Articles for Discovering Citation Intent Classification. *Complexity*, *2021*.

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (pp. 5998-6008).