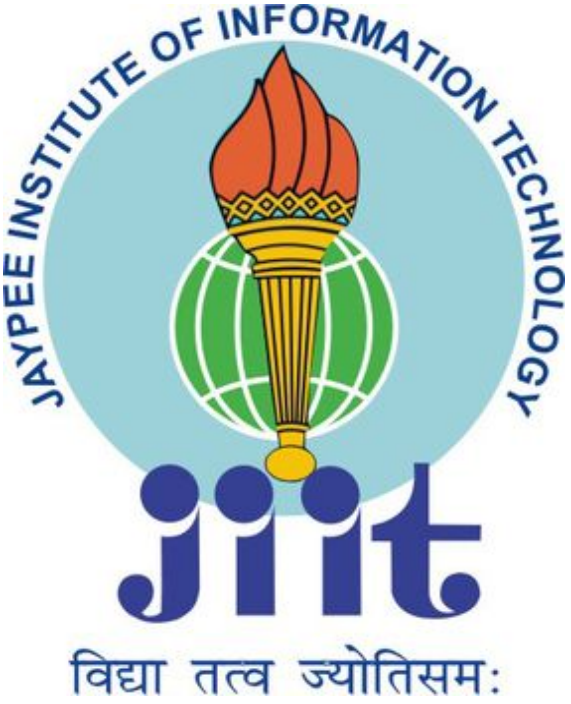
SHORT TEXT CLASSIFICATION USING MEMORY NETWORKS

Enrollment No(s) - 16103221, 16103046, 16103015
Name of Student(s) - Aman Parmar, Palak Arora, Anjali Sharma
Name of supervisor - Mr. Prashant Kaushik



May-2020

Submitted in partial fulfillment of the Degree of
Bachelor of Technology
in
Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

TABLE OF CONTENTS

(II)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Signature:

Date: Name:

Enrollment No:

(III)

CERTIFICATE

This is to certify that the work titled "**Short Text Classification using Memory Networks**" submitted by "**Aman Parmar, Palak Arora, Anjali Sharma**" in partial fulfillment for the award of degree of **B.Tech.** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: ……..

Name of Supervisor: ……..

Designation: ……..

Date: ……..

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We are highly indebted to **Mr. Prashant Kaushik** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to industry persons for giving us such attention and time.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

Name of Student	Enrollment Number	Signature
Aman Parmar	16103221	
Palak Arora	16103046	
Anjali Sharma	16103015	

Date:

(V)

SUMMARY

With the continued explosion of web-based business and Internet connection, new content type, shorter text, is often used in many places. Many research centers around the world are mining for short messages. It is a test to distinguish short content designed for its main characters, in small format, large size, fast, casual. It is difficult for custom techniques to handle short content; titles are great because words that are compressed in short content cannot speak to create space and connection between words and texts.Alternative content collection, has captured the last 5-6 years of time, since the unexpected explosion of web-based life use is universal. In the meantime, in the proper functioning of the various online phases and in making customer engagement and client experience continue to be fun and entertaining, the formal layout of the short content is crucial. In addition due to the lack of data and the set of short content produced, standard and standard AI and deep learning models are not efficient. Here comes our answer to the statement of matter. Fans who get an unintentional step by step prefer to use less words as reasonably expected while getting all the content, so the age of relevant content, which contains a large amount of data, is not normal.Short content orientation is not the same as standard content classification, in that it contains very little data to enable the model used to function properly.

Name & Signature of Students

Aman Parmar (16103221) ……

Palak Arora (16103046) ……

Anjali Sharma (16103015) ……

Date -

Signature of Supervisor

……....

Name - Mr. Prashant Kaushik

Date -

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS & ACRONYMS

Chapter No.	Symbol /Acronym	Definition
Chapter 1	PDA	Predictive Data Analysis
Chapter 1	SVM	Support Vector Machines
Chapter 1	KNN	K-Nearest Neighbours
Chapter 2	TMN	Topic Memory Network
Chapter 2	NTM	Neural Topic Model
Chapter 2	LSTM	Long Short Term Memory
Chapter 2	BoW	Bag of Words
Chapter 6	GLoVe	Global Vectors

Chapter 1

Introduction

1.1 General Introduction

With the recent explosion of e-commerce and online communication, new text type, short text,is widely used in many places. Most research focuses on the mines of short texts. It is a challenge to separate the short text made for its natural characters, as well as sparseness, large size, fast, rare. It is difficult for traditional methods to deal with short text; the distinction is largely due to the very limited names in the short text cannot represent feature space and relationships between words and texts. More research and updates for text segmentation are shown in recent times. However, only a few researchers focused on the short text to be separated. This report deals with short characters text and complexity of short text separation. After that we appreciate existing works popular in short text reading fashion and models, including short text input using semantic critique, supervised text fragmentation, coherence short text separation, and real-time programming. The analysis of short text editing is analyzed in our report. Finally we summarize the existing distinctions technology and hope for the development of short texts to be separated.

With the blast of web based business and on the web correspondence, short messages become accessible in numerous application zones, for example, Instant Messages, online Chat Logs, Bulletin Board System Titles, Web Logs Remarks, Internet News Comments, SMS, twitter and so forth. Along these lines, effectively handling them becomes progressively significant in many Web and IR applications. Notwithstanding, it is another difficulty that characterizes these sorts of content and Web information. In contrast to ordinary reports, these content and Web portions are normally noisier, less theme centered, and a lot shorter, that is, they comprise from twelve words to a couple of sentences. In view of the short length, they try not to give enough word co-event or shared setting for a decent similitude measure. Thus, ordinary AI strategies, which depend on the word recurrence, enough word co-events or shared settings to gauge the closeness of records, normally neglect to accomplish exactness because of the information scantiness.

New ordering techniques on short content are shown, for example, sematic examination, semi-directed short content grouping, gathering models for short content, and real time arrangement. Be that as it may, contrasted and a great deal of audits and reviews on content characterization, just not many of overviews seem to talk about the ongoing looks into short content order. This report dissects the challenges related with ordering short content and foundational sums up the current related strategies to short content characterization utilizing systematic measures.

1.2 Problem Statement

Since the boost of technological era, the use of social media, messaging applications, e commerce web portals has boosted exponentially in a very short time. Since with the rise of the technological age, more and more companies are focused on improving their products by making them smarter with each coming day, sentiment analysis of the text written by the user has gained immense popularity. Various famous application softwares like Siri by Apple, Alexa by Amazon, Cortana by Microsoft and Google Assistant by Google itself have been working on the same principle of analysing the sentiment of the user input, be it speech or text, and classifying the same to improve the sorted results in a much better way.

Humans getting lazy day by day opt to use as less of words as possible while conversing over text, so the generation of proper text, containing good amount of information, is hard to find.
Classification Of short text is not the same as classification of general length text, as it contains much less information to enable the incorporated model to work efficiently.

1.3 Significance / Novelty of the problem

The classification of short text in the correct way, has gained significance in the last 5-6 years of time, since the sudden boom of the use of social media globally. This time, for the proper working of various online platforms and to make the user interaction and user experience more fun and easy going, proper classification of short text is necessary. Also due to the lack of information and context in the generated short text, the traditional and general machine learning and deep learning models can’t work efficiently. At this place comes our solution to the problem statement.

1.4 Empirical Study

Short content is generally utilized in numerous fields Versatile content informing, texting, BBS article, news Theme, online talk record, blog remarks, news remarks, And so forth and its primary component is the length of the report Short, close to 200 characters. Like a PDA The message we utilize each day is no more 70 letters, BBS title and under 30 news features. Texting programming (IM) underpins long message. For texting and ensure it's protected, IM Programming additionally restricts its length, for example, Windows Live Microsoft Messenger permits an any longer message 400 characters. Truth be told, in regular communications, I The text is only twelve words. By and large, short content highlights are as per the following :

Spareness: There are just a couple in the short content A few words have certain characteristics that they don't give Lacking word blends or very much shared setting The match rate. It is hard to catch enactment Attributes of language.

Moment: Short content is additionally sent Adjusted continuously. Also, the size is extremely huge Fantastic.

Detriment: Short content portrayal Short, with a ton of missed words, new words Audio cues and unbalanced appropriation: application The space, (for example, organize security) should be tended to A limited quantity of little content information. All things considered, we can Concentrate just on the littlest part in the center (getting things) Large information. In this way, helpful conditions are restricted, Furthermore, the appropriation of short content is lopsided.

Enormous Data and Labeling for Robots: This Is Hard Penmanship all cases for a huge scope. Just restricted conditions can give constrained data. So how might you exploit these labeled circumstances Another case of undeveloped turning into a major issue Short content partition. Numerous customary techniques, (for example, SVM, BAYES, and KNN dependent on name coordinating) When all is said in done, overlook the short content property. The Customary techniques may not be related with short content Must be isolated. The majority of them (like BAYES) may fall flat Get high precision when named subtleties That is insufficient. What's more,

different strategies for order In light of the vector space model (SVM) they should utilize Semantic data to improve the presentation of Trespassers.

1.5 Brief Description of Solution Approach

In this project, we have used different machine learning and deep learning algorithms, ranging from the traditional ones to advanced ones, like topic memory networks and long short memory networks so as to extract as much information from the short text instances as possible, and thus classify them appropriately.

Numerous text instances have been collected through various sources, which cover multiple classes and categories like, crime, entertainment, politics, music etc.

Approach used in this project for classification of short text is:

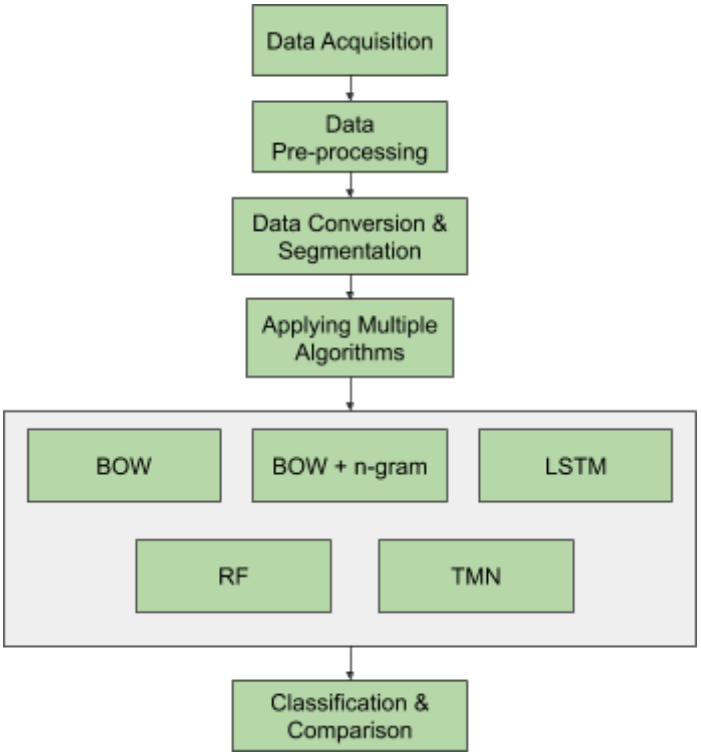


Figure 1 : Basic working model

A. Data Acquisition

The first stage of any machine learning based software solution is acquiring proper data in the proper format. We collected data from, consisting of text instance and associated class label, fro multiple sources like kaggle, github and google database search.

B. Data Pre-Processing

IData preprocessing consists the steps involved in making the acquired data suitable for the desired models and application. We transformed the data by combining some of the classes and by deleting some as well.

C. Data Conversion and Segmentation

We converted the data into multiple formats, with multiple classes and features to make it suitable for the applicable model. Many classes within the dataset were combined together and many were removed which were found not suitable for the classification.

D. Application of multiple algorithms

For the sake of understanding the working of different algorithms and to appreciate the complexity of the problem statement we used multiple algorithms over the dataset. The applied algorithms are:

- Bag Of Words
- Bag of Words + n-grams
- Random Forest
- Long Short Term Memory or LSTM
- Topic Memory Network or TMN

E. Classification & comparison of results

Finally, classifiers are used for the training and testing of the datasets. Application of different machine learning and deep learning models helped us compare the results in a much better way, and thus proved why memory networks work the best for this sort of problem statement.

F. Proposed Approach

Proposed system starts with the acquisition of proper data samples. The data samples are pre-processed to remove noise and any other unwanted parameters and features. It is then segregated and combined so as to remove unnecessary classes and to combine similar classes. Later classification is done using the advanced memory networks based on which we are going to compare the effectiveness of various algorithms.

1.6 Comparison of existing approaches to the problem framed

Table 1 : Comparison of different approaches

Technique	Advantage	Disadvantage
Bag Of Words (BoW)	Simpler classifier in terms of training and building process Applicable in case of a small dataset of proper length text instances	Due to the formation of word vectors, can lead to the curse of dimensionality. Expensive in terms of space.
Bag of Words + n-grams	Helps in extracting more information from the text instances by allowing multi word phrases. Simpler classifier in terms of training and building process.	Disadvantages similar to that of a vanilla Bag of Words model. Due to the formation of word vectors, can lead to the curse of dimensionality. Expensive in terms of space.
Random Forest Classifier	More complex classifier, thus captures more feature correlations. The prescient presentation can rival the best managed learning calculations.	Random Forest is not able to take into account the correlation of words in the text phrase. Do not work best for text data, especially short text instances.
Support Vector Machine (SVM)	Simple geometric interpretation and a sparse solution. Can be robust, even when the training sample has some bias. Works well with even unstructured and semi structured data like text.	Slow training. Difficult to understand the structure of algorithms. Curse of dimensionality can be a real mess in this algorithm due to the kernel design.

Chapter 2

Literature Survey

2.1 Summary of papers studied

Text classification has been a topic of great interest for the past many years, infact decades. Many researchers, being in groups or organisations or individually, have worked with immense interest in this field of problem, just to improve the solutions and make more and more effective and efficient approaches.Following is the related literature review of proposed work:

[1] As of now, content arrangement investigation is pointing in a few fascinating bearings. One of them is the endeavor at discovering better portrayals for content; while the sack of words model is as yet the magnificent content portrayal model, scientists have not disavowed to the conviction that a content must be something more that a unimportant assortment of tokens, and that the mission for models more modern than the pack of words model is as yet worth seeking after.

[2] Thought about different neural strategies on another legitimate XMTC dataset, EUR LEX 57K, additionally exploring not many shot and zero-shot learning. We appeared that BIGRU-ATT is a solid benchmark for this XMTC dataset, outflanking CNN-LWAN, which was exceptionally intended for XMTC, yet that supplanting the vanilla CNN of CNN-LWAN by a BIGRU encoder (BIGRU-LWAN) leads to the best generally results, aside from zero-shot names. For the last mentioned, the zero-shot rendition of CNNLWAN of Rios and Kavuluru produces excellent outcomes, contrasted with different strategies, and its presentation improves further when its CNN is supplanted by a BIGRU (Z-BIGRU-LWAN). Shockingly HAN and other various leveled strategies we considered (MAX-HSS, LW-HAN) are more fragile contrasted with the other neural strategies we explored different avenues regarding, which don't think about the structure (areas) of the records.

[3] In microblogging management, for example, Twitter, end-users can be overcome by confidential information. One answer to this issue is the order of short messages. Since shorter messages do not

provide enough word events, standard layout techniques, for example, "Bag-Of-Words" are problematic. To address this issue, we propose to use a small space highlighting arrangement that is extracted from the profile and content of the creator. The proposed method adequately classifies content content to a defined level of general classes, for example news, events, ideas, deals, and private messages. With such a framework, clients can purchase an entry or view specific types of tweets based on their interests. Experimental results show that the BOW method works fairly well but 8F works best with this nonlinear classification scheme. By using a small discriminatory optimization arrangement, their method provides a benchmark for displaying new tweets online with high confidence. In any case, inaccurate information can ruin this proposed road show; successive procedures for the transfer of stubbornness are important in such cases.

[4] Long Short Memory (LSTM) is a multipurpose neural application engine that exceeds expectations for removing sequential memory and repeats many steps later. LSTM's unique configuration calculation provides important structures for a specific and temporary region, which is not in the other configurations, with the constraint of forcing its configuration into a small program configuration. Here we present Short-Term Developmental Calculation (LSTM-g), which provides an environment similar to LSTM while corresponding without any change to the most common type of secondary planning models. With LSTM-g, all units have a vague set of practical guidelines for use and learning, depending on the setting of their neighbor's system; this has an initial LSTM configuration count, where each type of units has its own implementation and configuration indicators. In the case where it is included in the LSTM frameworks that have peephole relationships, the LSTM-g uses the additional endpoint of the retrieved error that would enable the preferred execution over the first calculation. Empowered by the extensive LSTM-g ordering capability, we show that transient optimization schemes built for transparency can create more popular effects than single-range systems. We teach that LSTM-g is likely to improve presentation and extend the understanding of spatial and temporal patterns of proximity in preparation for calculation of recurrent neural systems.

[5] The word sack model is one of the most popular techniques for object organization. The main idea is to enlarge each point divided into one visual word, and then talk to each image about the history of visual words. For this reason, group counting (e.g., K-means), is often used to generate visual cues. Various investigations have shown to strengthen the strength of the effects of

word-combining, the conceptual evaluation of items in the word sack model is unclear, possibly due to the problem presented through the grouping process. In this paper, we present a comparative scheme that summarizes the sack of words that are expressed. In this setting, virtual words are generated by a factual process instead of using clustering calculations, while a more intuitive presentation and group consolidation strategy. The hypothetical assessment is based on the true consensus of the proposed system. In addition, in the light of the program we created two interdependent computations, while at the same time achieving less complexity in object planning as compared to the display of group-based word groups.

[6] After Louis' powerful conclusion, the use of machine learning processes for text classification became notorious. One requirement for the use of Mnost machine learning calculations is that manufacturing information can be talked about as the most highlighted vectors. The Straight-Forward Method for Speaking for Content as Highlight Vectors is the arrangement of the word system: the collection of the report classification is done by an element vector that has a Boolean property. This means that in the event that a word is in a particular manufacturing record, its corresponding statement is set to 1, if it is not set to 0. Along these lines, each record is spoken by the arrangement of words within it. In this paper, we study the effect of capturing the alignment of the word-inheritance approach using n-gram as a highlight. The results show word clusters of length 2 or 3 as a rule, improving the characterization function as a rule, but most groups do not help: they are similarly valued as dependent medium redundancy, while repetition-based pruning brings a diminishment on reflective datasets.

2.2 Integrated summary of literature studied

After elaborately analysing all the research papers on text classification algorithms and techniques or short text classification or memory networks, we clearly understood our target problem statement of classifying short text instances. Research papers nt only cleared the path of working on the problem statement but also helped us understand the catches and the issues with analysing short text data.

After considering multiple research papers on varied techniques and algorithms, we finalised some of them to work on in the final implementation of the solution for the problem. The algorithms finalised are;

- Bag of Words
- N-grams
- Random Forest
- Memory networks
    - Long Short Memory Networks
    - Topic Memory Networks

One more algorithm, Bidirectional Encoder Representations for Transformers or BERT, was also considered for implementation, and has been considered as the foremost topic to work on in the future.

With regard to the research papers we came to know the traditional algorithms used for classification of text instances, their working procedure and also why they cannot be used as the first choice for classifying short text instances. Bag of Words and N-grams work by creating vectors, but in context of short texts, the information associated with them is pretty less and thus these algorithms cannot work properly. Similar is the problem of using random forest as a solution, as it works by mapping multiple correlations between the data, low presence of information hinder the working of the algorithm.

Memory networks are advanced machine learning algorithms that incorporate some complex deep learning ideas and work very closely to how a human brain works in similar conditions. TMN for instance, works by finding latent topics in the text instances, topics which are correlated to the text instance but are not present directly in it, it then maps the latent topics with the suitable class and thus increases the associated information with each instance of short text. This helps in improving the working of the algorithm and also removes the common problem of less information associated with the instance.

Chapter 3

Requirement Analysis and Solution approach

3.1 Overall Description of the Project

In today's scenario, where the world is majorly driven by technology of various sorts, human speech has become kinda obsolete, as the majority of the communication is taking place across various platforms of technology like whatsapp, Emails, twitter, and many many more.
In this situation, understanding human emotions through the text associated with him is a priority. Understanding emotions, classification of messages, proper grouping and other such tasks which can help us improve the technology much more by making it more sophisticated. As the whole of the world is working towards making technology more sophisticated and agile so as to improve the user's interaction experience, proper understanding of text for proper classification is kind of a must. Since the boost of the messaging generation, people are finding new and innovative methods to express their feelings in as few words as possible, by using emoticons, slangs and some special phrases. A general smart machine is not able to classify small text instances as the associated information is very less.

This is where our solution to this problem comes in. We incorporated advanced machine learning and deep learning algorithms, which work in a more similar pattern to a human brain, like long short memory networks, topic memory networks and bidirectional encoder Representations from Transformers to solve the problem statement in a much better way.

Not only did we use the algorithms to classify the small text, but we also used the results extracted to compare with the results of some of the more traditionally used machine learning and deep learning algorithms, like bag of words model, n-gram model and random forest, associated with the tack of text classification.

3.2 Requirement Analysis

A. Python 3.7.1 (Latest version)

Python is a broadly useful deciphered, intelligent, object-situated, and elevated level programming language. It was made by Guido van Rossum from 1985-1990. Like Perl, Python source code is additionally accessible under the GNU General Public License (GPL). This instructional exercise gives enough comprehension of Python programming language.

It's most recent form discharge is 3.7.1, which is likewise utilized in the task created. Python being intuitive and because of its gigantic library support ends up being the best language to work over AI and information examination and profound learning too.

B. Jupyter

The IDE utilized for performing illness forecast is Jupyter Notebook. It is a python supervisor that joins Anaconda. The Jupyter Notebook can be executed on a nearby work area requiring no web get to or can be introduced on a remote server and got to through the Internet. For utilizing an irregular backwoods classifier, Scikit-learn is imported, which is a free programming AI library for Python. It highlights different characterization, relapse and grouping calculations including SVM, Random Forests, Gradient Boosting, K-means and DBSCAN, and is intended to interoperate with the Python numerical and logical libraries NumPy and SciPy.

C. Google Colab

Colaboratory is a free Jupyter notebook condition that requires no arrangement and runs altogether in the cloud. With Colaboratory you can compose and execute code, spare and offer your investigations, and access ground-breaking registering assets, just for nothing from your program. Like jupyter notebook, google colaboratory or colab as called is additionally an intuitive meeting that takes a shot at .ipynb record designs.

Colab causes us to execute tremendous and substantial AI and profound learning calculations that can't be run over the nearby machine effectively because of the nonappearance of a legitimate graphical preparing unit or a GPU as called. Colab gives a coordinated GPU bolster utilizing the K80 GPU working at a speed of 1.8TFlops directions per unit time, alongside a 12 GB of RAM. Alongside the GPU colab additionally gives a TPU backing to running much greater datasets like the one dealt with in huge information analysis.TPU or tensor preparing unit as brought works over disseminated registering by giving a speed of 180TFlops per unit time and an aggregate of 180GB of RAM.

3.3 Solution Approach

Since this project is developed by keeping the general public in mind and all the possible inconvenience that can be faced by them, we, unlike the other research papers, studied along the way of developing the project, we incorporated a new and wholesome approach towards the development of the project. The approach incorporated by us in this project is less researched and thus is hard to understand and the practical application of the same is complex and tricky.

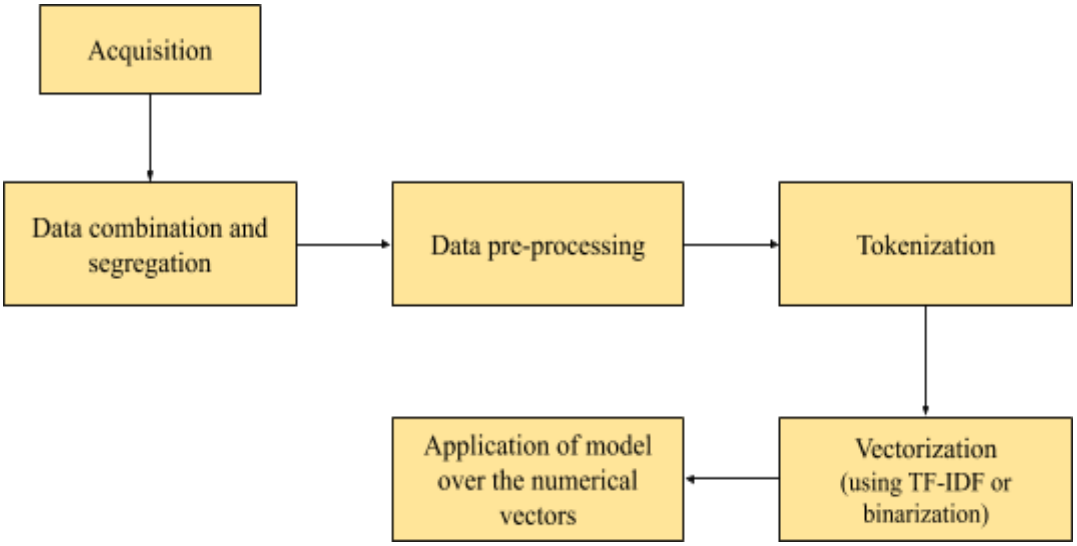


Figure 2 : Traditional working of text classification models

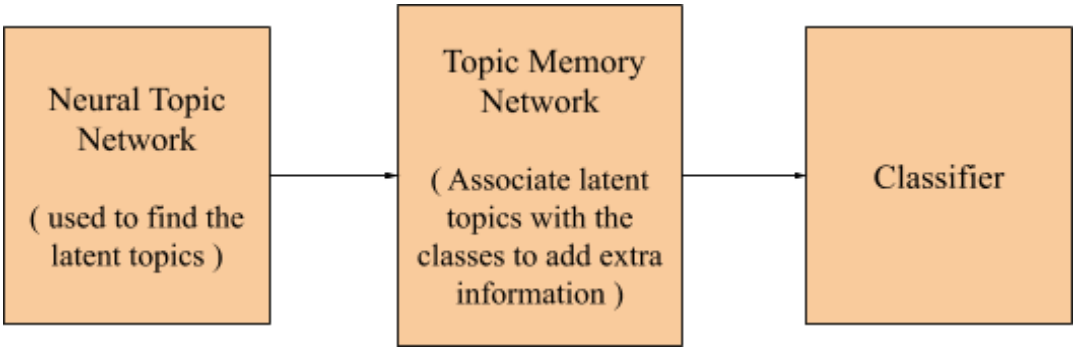


Figure 3 : Topic Memory Network

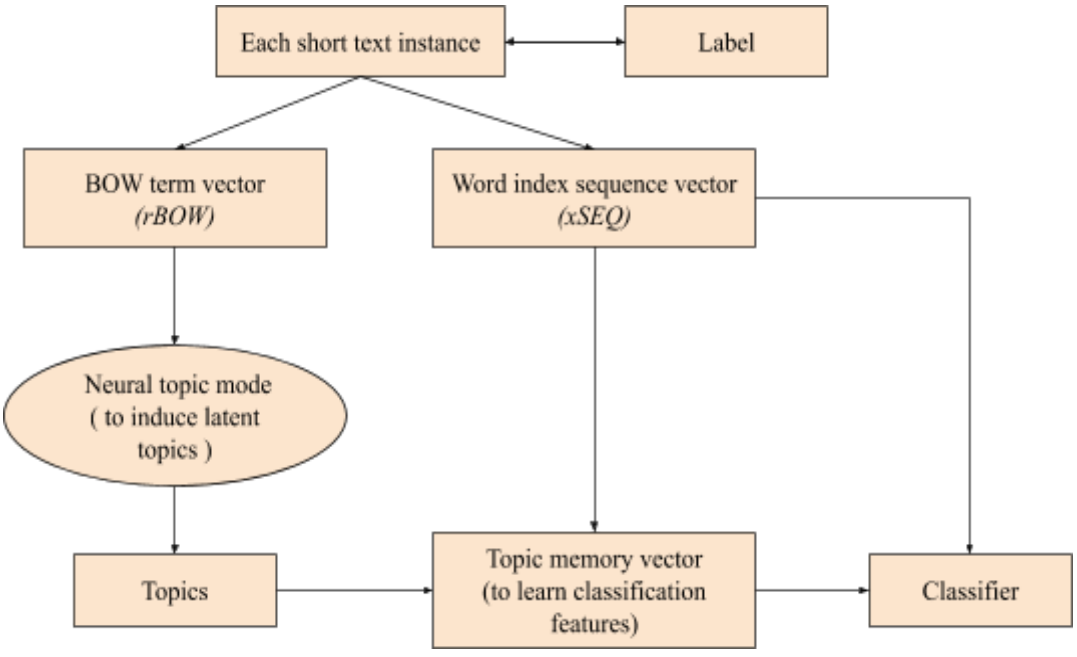


Figure 4 : Working of TMN

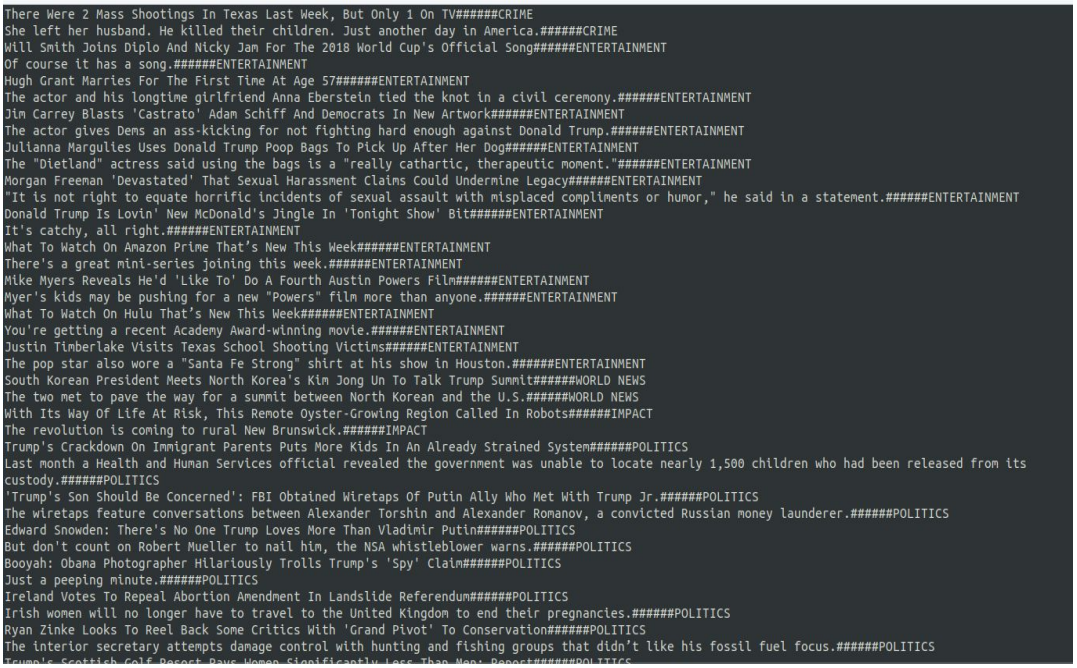
Dataset Description:

- **Datasets' sources:**
  - The dataset has been collected from various sources :
    - Kaggle
    - GitHub
    - Google Database Search
    - Twitter

  - **Dataset Size:**
    - Original dataset 1 : 41+ MB of text data
      - Format : <text_instance>######<class_label>
      - 86000+ text instances
    - Original dataset 2 : 84MB of json data
      - Format : {short_text , description ,timestamp, class_label}
      - (200K+) * 2 = 400K+ text instances

  - The major datasets used in the application of the algorithms consist of about 28 major classes. While preprocessing the data sets to remove any noise and irregular data fields and instances, we removed some of the classes having less number of instances, and combined some similar classes into one more general class so as to increase the number of instances associated with it. Some of the classes included in the dataset were:
    - Crime
    - Entertainment
    - Politics
    - Music
    - Education, and so on..

Screenshots to understand the dataset format better



```
There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV######CRIME
She left her husband. He killed their children. Just another day in America.######CRIME
Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Official Song######ENTERTAINMENT
Of course it has a song.######ENTERTAINMENT
Hugh Grant Marries For The First Time At Age 57######ENTERTAINMENT
The actor and his longtime girlfriend Anna Eberstein tied the knot in a civil ceremony.######ENTERTAINMENT
Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats In New Artwork######ENTERTAINMENT
The actor gives Dems an ass-kicking for not fighting hard enough against Donald Trump.######ENTERTAINMENT
Julianna Margulies Uses Donald Trump Poop Bags To Pick Up After Her Dog######ENTERTAINMENT
The "Dietland" actress said using the bags is a "really cathartic, therapeutic moment."######ENTERTAINMENT
Morgan Freeman 'Devastated' That Sexual Harassment Claims Could Undermine Legacy######ENTERTAINMENT
"It is not right to equate horrific incidents of sexual assault with misplaced compliments or humor," he said in a statement.######ENTERTAINMENT
Donald Trump Is Lovin' New McDonald's Jingle In 'Tonight Show' Bit######ENTERTAINMENT
It's catchy, all right.######ENTERTAINMENT
What To Watch On Amazon Prime That’s New This Week######ENTERTAINMENT
There's a great mini-series joining this week.######ENTERTAINMENT
Mike Myers Reveals He'd 'Like To' Do A Fourth Austin Powers Film######ENTERTAINMENT
Myer's kids may be pushing for a new "Powers" film more than anyone.######ENTERTAINMENT
What To Watch On Hulu That’s New This Week######ENTERTAINMENT
You're getting a recent Academy Award-winning movie.######ENTERTAINMENT
Justin Timberlake Visits Texas School Shooting Victims######ENTERTAINMENT
The pop star also wore a "Santa Fe Strong" shirt at his show in Houston.######ENTERTAINMENT
South Korean President Meets North Korea's Kim Jong Un To Talk Trump Summit######WORLD NEWS
The two met to pave the way for a summit between North Korean and the U.S.######WORLD NEWS
With Its Way Of Life At Risk, This Remote Oyster-Growing Region Called In Robots######IMPACT
The revolution is coming to rural New Brunswick.######IMPACT
Trump's Crackdown On Immigrant Parents Puts More Kids In An Already Strained System######POLITICS
Last month a Health and Human Services official revealed the government was unable to locate nearly 1,500 children who had been released from its
custody.######POLITICS
'Trump's Son Should Be Concerned': FBI Obtained Wiretaps Of Putin Ally Who Met With Trump Jr.######POLITICS
The wiretaps feature conversations between Alexander Torshin and Alexander Romanov, a convicted Russian money launderer.######POLITICS
Edward Snowden: There's No One Trump Loves More Than Vladimir Putin######POLITICS
But don't count on Robert Mueller to nail him, the NSA whistleblower warns.######POLITICS
Booyah: Obama Photographer Hilariously Trolls Trump's 'Spy' Claim######POLITICS
Just a peeping minute.######POLITICS
Ireland Votes To Repeal Abortion Amendment In Landslide Referendum######POLITICS
Irish women will no longer have to travel to the United Kingdom to end their pregnancies.######POLITICS
Ryan Zinke Looks To Reel Back Some Critics With 'Grand Pivot' To Conservation######POLITICS
The interior secretary attempts damage control with hunting and fishing groups that didn’t like his fossil fuel focus.######POLITICS
Trump's Scottish Golf Resort Pays Women Significantly Less Than Men: Report######POLITICS
```

Figure 5: Dataset 1 (format: <instance>######<class_label>)

	A	B	
5	5	The actor and his longtime girlfriend Anna Eberstein tied the knot in a civil ceremony.	ENTERTAINMENT
6	6	Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats In New Artwork	ENTERTAINMENT
7	7	The actor gives Dems an ass-kicking for not fighting hard enough against Donald Trump.	ENTERTAINMENT
8	8	Julianna Margulies Uses Donald Trump Poop Bags To Pick Up After Her Dog	ENTERTAINMENT
9	9	The "Dietland" actress said using the bags is a "really cathartic, therapeutic moment."	ENTERTAINMENT
10	10	Morgan Freeman 'Devastated' That Sexual Harassment Claims Could Undermine Legacy	ENTERTAINMENT
11	11	"It is not right to equate horrific incidents of sexual assault with misplaced compliments or humor," he said in a statement.	ENTERTAINMENT
12	12	Donald Trump Is Lovin' New McDonald's Jingle In 'Tonight Show' Bit	ENTERTAINMENT
13	13	It's catchy, all right.	ENTERTAINMENT
14	14	What To Watch On Amazon Prime That’s New This Week	ENTERTAINMENT
15	15	There's a great mini-series joining this week.	ENTERTAINMENT
16	16	Mike Myers Reveals He'd 'Like To' Do A Fourth Austin Powers Film	ENTERTAINMENT
17	17	Myer's kids may be pushing for a new "Powers" film more than anyone.	ENTERTAINMENT
18	18	What To Watch On Hulu That’s New This Week	ENTERTAINMENT
19	19	You're getting a recent Academy Award-winning movie.	ENTERTAINMENT
20	20	Justin Timberlake Visits Texas School Shooting Victims	ENTERTAINMENT
21	21	The pop star also wore a "Santa Fe Strong" shirt at his show in Houston.	ENTERTAINMENT
22	26	Trump's Crackdown On Immigrant Parents Puts More Kids In An Already Strained System	POLITICS
23	27	Last month a Health and Human Services official revealed the government was unable to locate nearly 1,500 children who had been released fr	POLITICS
24	28	'Trump's Son Should Be Concerned': FBI Obtained Wiretaps Of Putin Ally Who Met With Trump Jr.	POLITICS
25	29	The wiretaps feature conversations between Alexander Torshin and Alexander Romanov, a convicted Russian money launderer.	POLITICS
26	30	Edward Snowden: There's No One Trump Loves More Than Vladimir Putin	POLITICS
27	31	But don't count on Robert Mueller to nail him, the NSA whistleblower warns.	POLITICS
28	32	Booyah: Obama Photographer Hilariously Trolls Trump's 'Spy' Claim	POLITICS
29	33	Just a peeping minute.	POLITICS
30	34	Ireland Votes To Repeal Abortion Amendment In Landslide Referendum	POLITICS
31	35	Irish women will no longer have to travel to the United Kingdom to end their pregnancies.	POLITICS
32	36	Ryan Zinke Looks To Reel Back Some Critics With 'Grand Pivot' To Conservation	POLITICS
33	37	The interior secretary attempts damage control with hunting and fishing groups that didn’t like his fossil fuel focus.	POLITICS
34	38	Trump's Scottish Golf Resort Pays Women Significantly Less Than Men: Report	POLITICS
35	39	And there are four times as many male as female executives.	POLITICS
36	42	Twitter #PutStarWarsInOtherFilms And It Was Universally Entertaining	ENTERTAINMENT
37	43	There's no such thing as too much "Star Wars."	ENTERTAINMENT
38	54	Warriors Coach Steve Kerr Calls NFL Ban On Protests 'Fake Patriotism'	BLACK VOICES
39	55	Forbidding players to take a knee during the national anthem is “idiotic,” the coach said.	BLACK VOICES
40	56	In Historic Victory, Barbados Elects First Female Prime Minister	BLACK VOICES
41	57	Mia Amor Mottley even earned the backing of the country's most recognizable national: Rihanna.	BLACK VOICES
42	58	Police Killed At Least 378 Black Americans From The Moment Colin Kaepernick Protested	BLACK VOICES

dataset_17classes_more7000

Figure 6 : Dataset 1 - converted to csv file

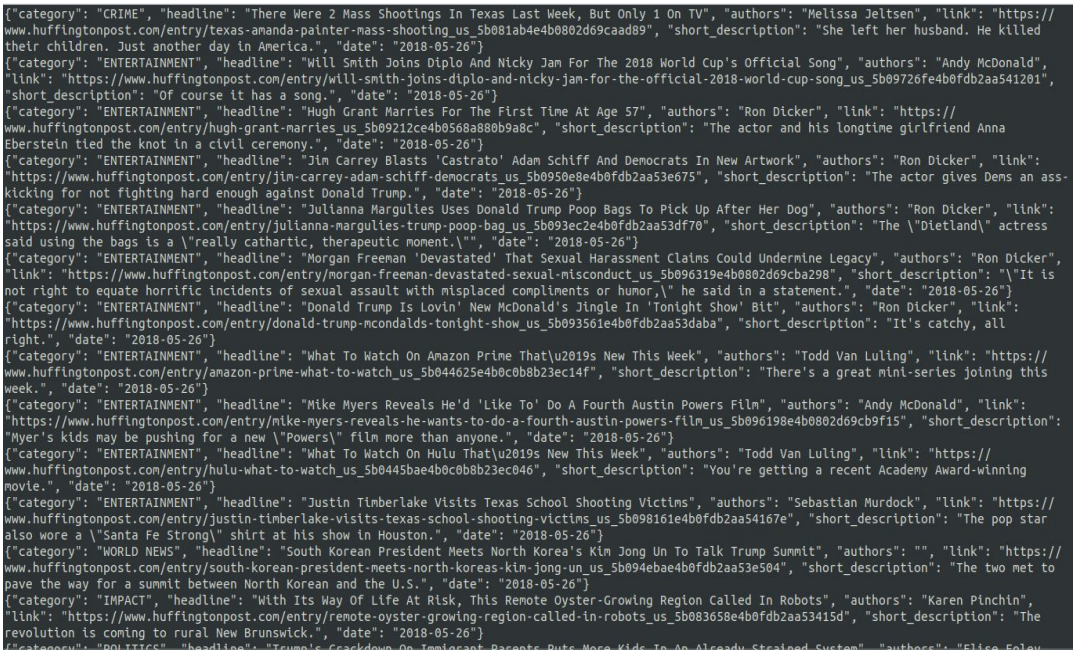


Figure 7 : Dataset 2 - json file

Our project for now focuses on the implementation of the basic and traditional machine learning algorithms used for text classification along with the detailed analysis and implementation of one of the most intuitive and advance text classification algorithms these days, Memory Networks.

The algorithms tested during the course of implementing the solution for this problem statement are:

- Bag of Words model
- Bag Of Words + n-grams model
- Random Forest
- Long Short Memory Networks
- Topic Memory Networks
- Bidirectional Encode Representations from Transformers

Before the application of the machine learning models over the dataset, proper preprocessing is required. The preprocessing involves steps like:

```
[ ]  dataset1=pd.read_csv("/content/drive/My Drive/MAJOR 2- WORK IN PROGRESS/dataset_25classes_less7000.csv")

[ ]  # quick analysis of dataset1

     #shape of the dataset
     print(dataset1.shape)
     print("==================================================")

     #finding the columns in the dataset
     print(dataset1.columns)
     print("==================================================")

     #all the unique classes
     po=dataset1.category.unique()
     for i in po:
         print(i)
     print("==================================================")

     #number of different classes in the dataset
     print(len(po))
     print("==================================================")

     #value count for each of the class in the dataset
     print(dataset1.category.value_counts())
     print("==================================================")

     #dropping the na isntances from the dataset, adn counting the value counts again
     dataset1=dataset1.dropna()
     print(dataset1.shape)
     print("==================================================")
     print(dataset1.category.value_counts())
     print("==================================================")
```

1. Figure 8 : Quick Analysis of the dataset

```
[ ]  #combining some of the similar classes into 1
     print("combining some of the classes into 1....")
     dataset1['category'].replace({"COLLEGE":"COLLEGE & EDUCATION","EDUCATION":"COLLEGE & EDUCATION",
                                   "GREEN":"ENVIRONMENT","TECH":"TECH & SCIENCE","SCIENCE":"TECH & SCIENCE",
                                   "CULTURE & ARTS":"ARTS & CULTURE","WEIRD NEWS":"NEWS","GOOD NEWS":"NEWS","WORLD NEWS":"NEWS","ARTS":"ARTS & CULTURE"}, inplace=True)

     removed_classes=["LATINO VOICES","FIFTY"]
     dataset1=dataset1[~dataset1['category'].isin(removed_classes)]
     print("processing DONE.")


     #value count of the new dataset formed
     print(dataset1.category.value_counts())
     print("==================================================")

     #printing the head of the dataset
     print(dataset1.head(5))
     print("==================================================")
```

2. Figure 9 : Combining similar classes and removing extra classes

```
[ ]  from nltk.corpus import stopwords
     nltk.download('stopwords')
     from nltk.tokenize import word_tokenize

     text = "Nick likes to play football, however he is not too fond of tennis."

     for  text in documents['text_instance']:
         text_tokens = word_tokenize(text)
         tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]
         print(tokens_without_sw
```

3. Figure 10 : Removing Stopwords

```
from nltk.tokenize import sent_tokenize, word_tokenize
def stemSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(porter.stem(word))
        stem_sentence.append(" ")
    return "".join(stem_sentence)

x=stemSentence(sentence)
print(x)
```

4. Figure 11 : Stemming and Lemmatization

```
# factorising the caterogy column of the dataset .i.e, converting classes into numbers using hard code

print("creating a dictionary for mapping each class to a unique number")
classes=dataset1.category.unique()
count =1
dict_classes= {}
for i in classes:
    dict_classes[i]=count
    count+=1

print("final value of count ", count)
print("=====================================================")
```

5. Figure 12 : Vectorization

The models used :

1. **Bag Of Words Model:**

   The most basic yet efficient text classification algorithm in use these days. Since bag of words work on the principle of information contained in the text instances, since the dataset is formed of short text instances, bag of words model worked in a pitiful manner, by providing an accuracy of just about 56%.

2. **Bag of Words + n-grams :**

   N-grams help in adding another dimension of information to the text instances by providing information regarding phrases and combination of words of the text instances. But adding this information along with the bow model didn't worked as well. Bow + n-grams provided an accuracy of about 61-62% only.

3. **Random Forest :**

   Random Forest was supposed and expected to work in a much better way, as it forms multiple and complex correlations amongst the data points. But since the dataset consisted of text with length of about 7-10 words, these correlations proved to be futile. The accuracy achieved by the random forest classifier was about the same that was obtained by applying the BOW+n-grams model , i.e., about 60%.

4. **Long Short Memory Networks (LSTM) :**

   Memory networks were supposed to work better in case of short text classification, as they work in a way which is more similar to the human brain, and the same was evident after analysing multiple research papers on the same topics. Long Short Memory Networks (LSTM) worked in a decent manner by providing an accuracy of about 67%.

5. **Topic Memory Networks (TMN) :**

   These kind of machine learning networks are one of the most complex networks used for text classification. These models not only find the latent topics, which are hidden to other classifiers, but only help in successfully mapping these latent topics to their respective classes. Also, apart from all this, TMN also takes into account the phrase information and

word sequence patterns. As expected, TMn worked in the most exceptional manner by providing an accuracy of about 83%, over the short text instances data.

Chapter 4

Modelling and Implementation Details

4.1 Design Diagrams

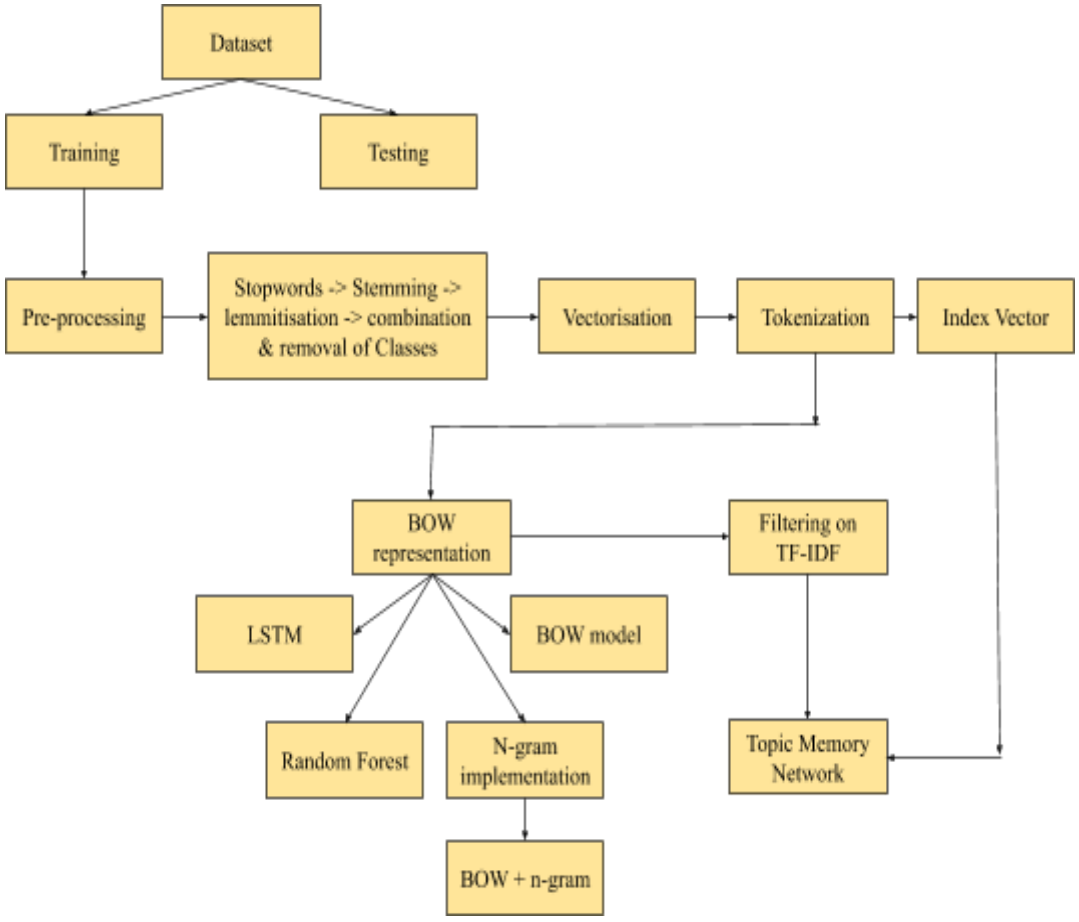


Figure 13 : Project Flowchart

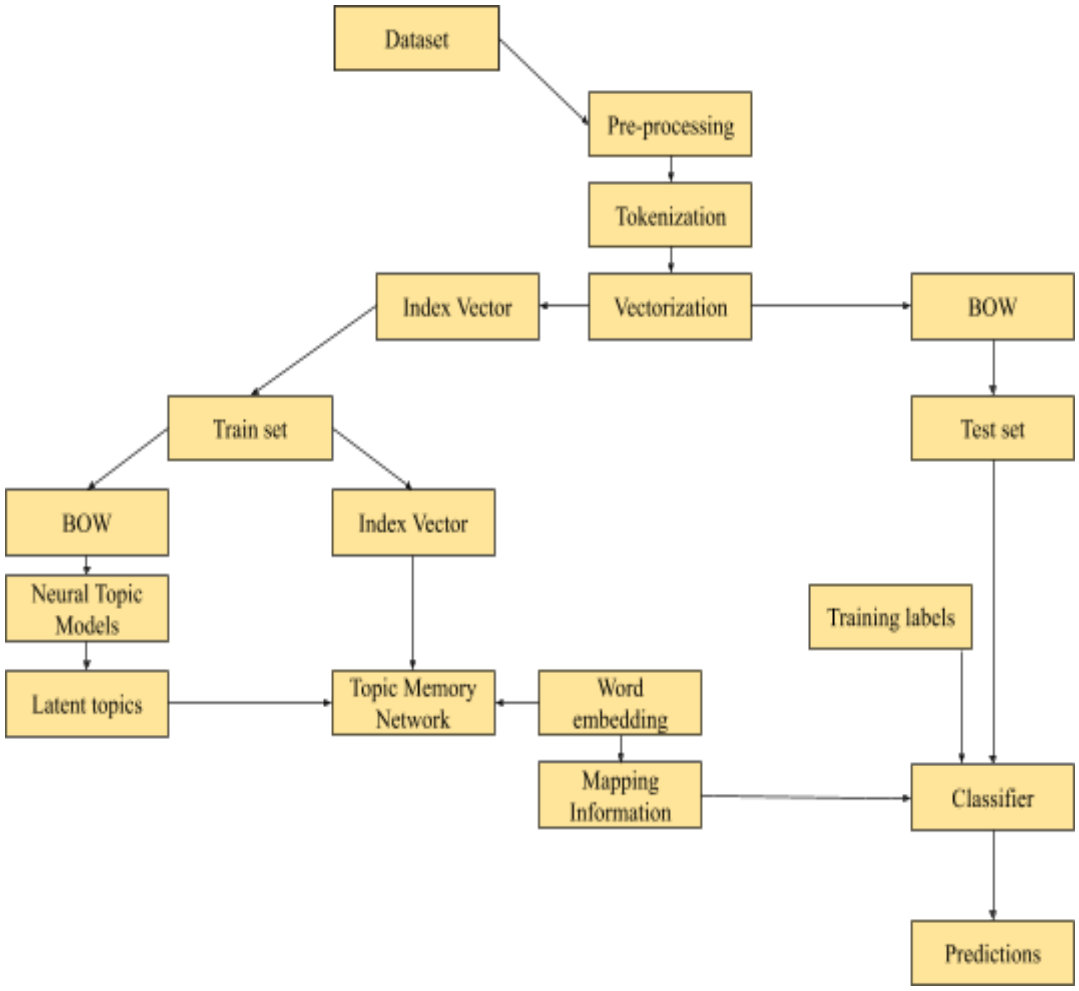


Figure 14 : TMN explained

4.2 Implementation details and issues

Drive Mounting

To access the dataset from cloud, colab in this case, drive is mounted using the Google Drive API, so as to access large datasets without any hassle.

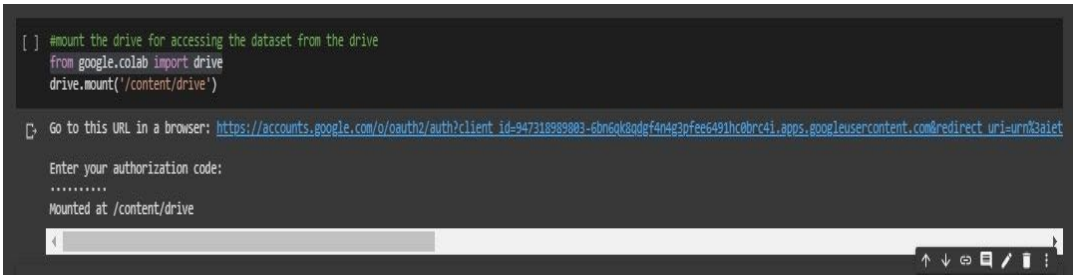


Figure 15 : Drive Mounting

Import the Libraries

```
import gensim   #used for topic modelling, document indexing and similarity retreival
from scipy import sparse        #used to create sparse matrices
from gensim.parsing.preprocessing import STOPWORDS
import logging #This module defines functions and classes which implement a flexible event logging system for applications and libra

#deep learning libraries


from keras import backend as K
from keras import regularizers
from keras.layers import Input, Dense, Lambda, Activation, Dropout, Flatten, Bidirectional, Conv2D, MaxPool2D, Reshape, BatchNormali
from keras.models import Model, load_model
from keras.preprocessing.sequence import pad_sequences
from keras.utils import plot_model, Progbar, normalize
from keras.layers.recurrent import LSTM
from keras.layers.merge import add, concatenate
import utils
import keras
import numpy as np
from datetime import datetime
import os
import sys
import json
import pickle
import gensim
from sklearn.metrics import f1_score, accuracy_score
```


Figure 16 : Import libraries

Dataset

This dataset consists of leaf images of different plants with various diseases. It consists of 38 well distinguished classes, giving a total of more than 54000 images.

The images are divided into training and testing sets (80-20) and 10% images are used for validation.

```
[ ] dataset = pd.read_csv("/content/drive/My Drive/train_features.csv")

[ ] type(dataset)

    pandas.core.frame.DataFrame
```

Figure 17 : Read dataset

Preprocessing

Performing basic preprocessing steps is very important before we get to the model building part. The basic steps have already been defined in the previous section. Steps after that:

- **Splitting in train and test set, and vectorising**

Splitting the dataset in 80-20 ratio, i.e., 80% for train set and the remaining 20% for test set. Also vectorisation of the text samples is done using TF-IDF vectoriser ,i.e., Term-Frequency --- Inverse Document Frequency vectoriser.

importing and splitting the dataset into train and test

```
[ ] from sklearn.model_selection import train_test_split

    #%% handlig the dataset with classes with less than 7000 instances each
    s
    dataset=dataset.dropna()
    X=dataset['text_instance']
    y=dataset['category']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
```

vectorising the data samples and fitting a model

```
[ ] print("Vectorizing train...")

    vectorizer = TfidfVectorizer( max_features = 40000, ngram_range = ( 1, 3 ),
        sublinear_tf = True )
    train_x = vectorizer.fit_transform( X_train )

    print("Vectorizing test...")

    test_x = vectorizer.transform( X_test)

    print("Training...")
```

Figure 18 : Splitting in train and test set, and vectorising

- **Fitting the model**

After creating the dataset, using the TF-IDF vectoriser, the model is fitter over the vectors formed.

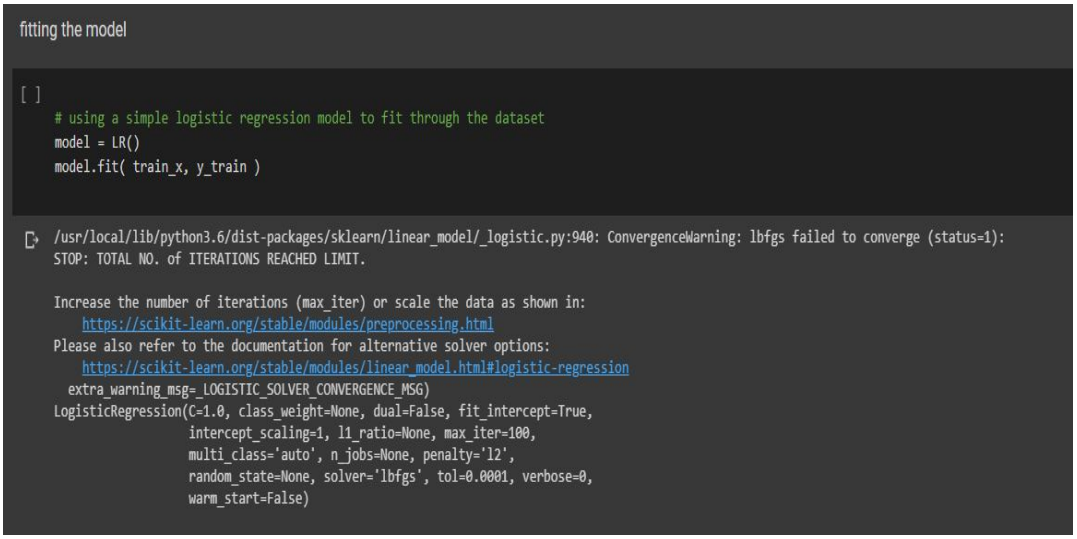


Figure 19 : Fitting the Model

- **Predicting**

  The results acquired using this simple model were below satisfactory with an accuracy of about 56%.

Figure 20 : Prediction through the model

Steps for Topic Memory Network - main section of the application

- **Convert the text instances into UNICODE**

  Unicode is an information technology standard for consistent encoding. Instead of using the basic ASCII, we used UNICODE conversion as it is more universal and covers each and every possible character, be it numbers, letters, symbols or emoticons.

Figure 21 : Convert the text instances into UNICODE

- **Splitting and tokenize the data instances**

  Splitting the data instances into separate columns of text data and the associated class label. Also using the gensim library, we tokenized the words in each of the text by first applying stemming.

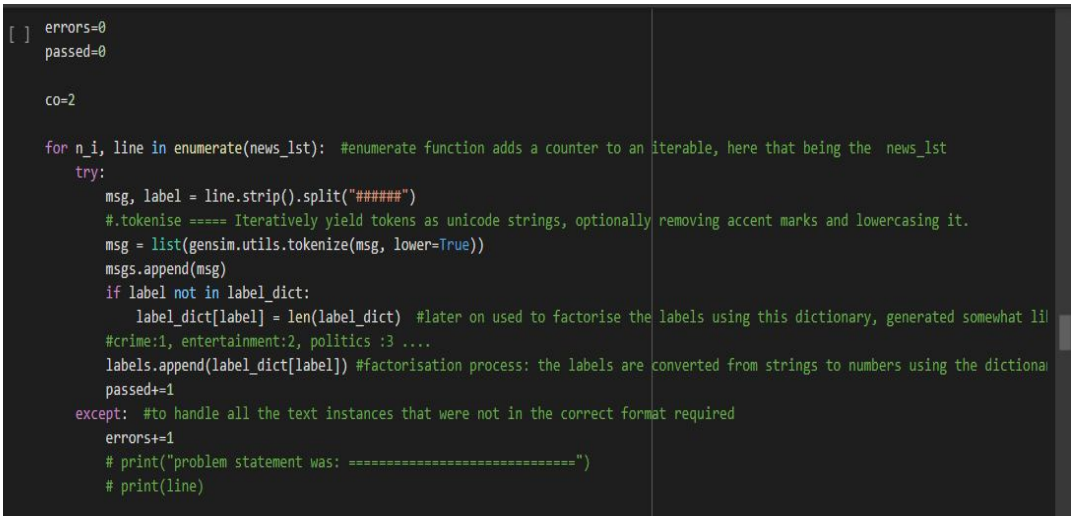


Figure 22 : Splitting and tokenize the data instances

- **Forming a dictionary of tokens**

  Forming a dictionary of unique tokens from all the data instances, later used to create the

index vector, bow representation and to find the latent topics. All this conversion is done using the gensim.corpora.Dictionary command.

Figure 23 : Forming a dictionary of tokens

- **Filtering the dictionary**

  Since stop words were not removed in the initial stage, we filter the dictionary formed by removing the stopwords, removing one character long tokens like “I”, “a”, etc., removing the tokens with document frequency less than 3.

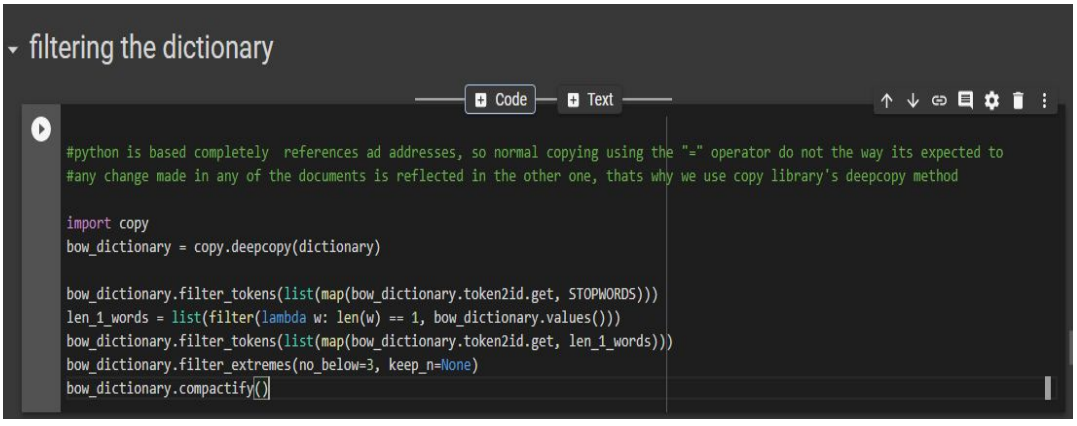


Figure 24 : Filtering the dictionary

- **Creating a BOW representation**

  Converting the tokenized document instances into bow representations using the filtered dictionary formed in the previous step.

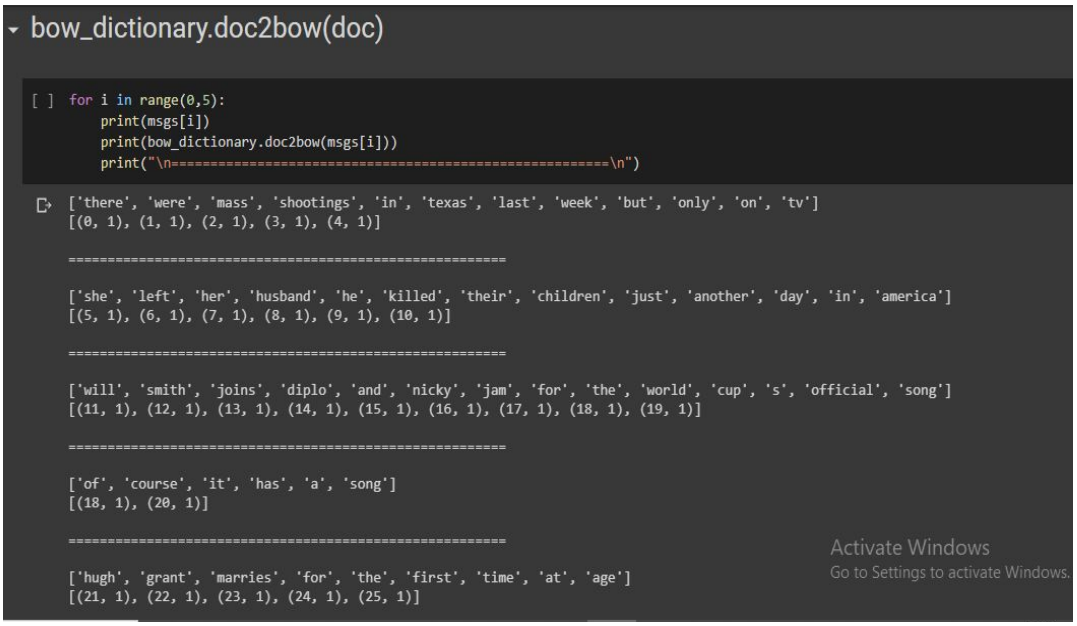


Figure 25 : Creating a BOW representation

- **Creating index vectors and bow sparse matrix**

  Creating a sparse matrix of BOW representations using each of the instances. Also creating an index vector and vector label array.

Figure 26 : Creating index vectors and bow sparse matrix

- **Saving file for future uses**

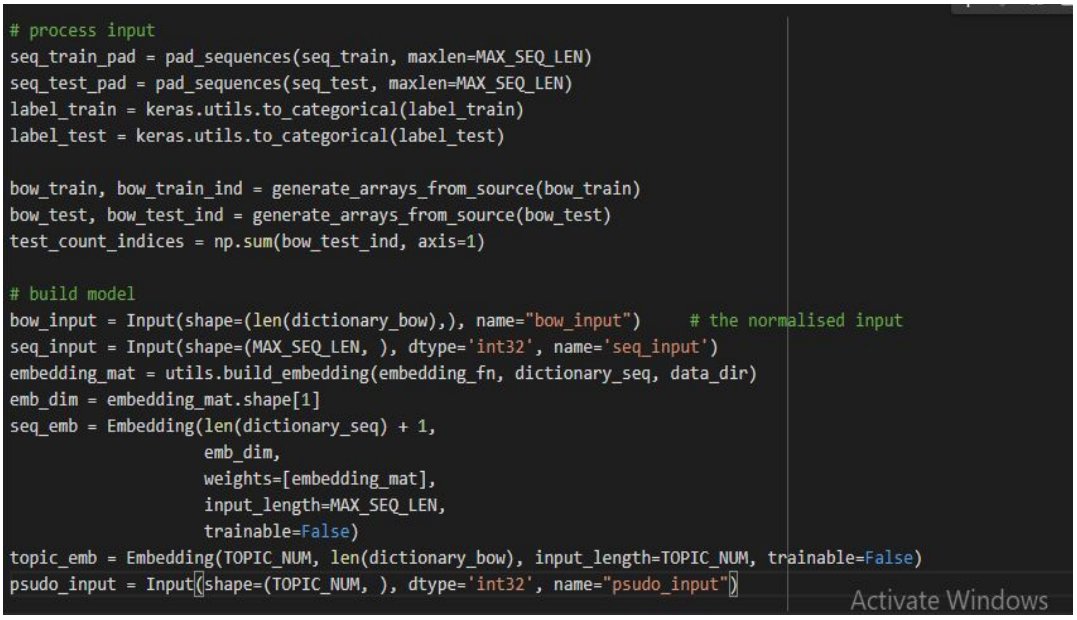
  Dumping each of the intermediate states formed in binary files using pickel for future use.

```
logging.info("save data...")
pickle.dump(seq_title, open(os.path.join(data_dir, "dataMsg"), "wb"))
pickle.dump(seq_title_train, open(os.path.join(data_dir, "dataMsgTrain"), "wb"))
pickle.dump(seq_title_test, open(os.path.join(data_dir, "dataMsgTest"), "wb"))
pickle.dump(bow_title, open(os.path.join(data_dir, "dataMsgBow"), "wb"))
pickle.dump(bow_title_train, open(os.path.join(data_dir, "dataMsgBowTrain"), "wb"))
pickle.dump(bow_title_test, open(os.path.join(data_dir, "dataMsgBowTest"), "wb"))
pickle.dump(label_title, open(os.path.join(data_dir, "dataMsgLabel"), "wb"))
pickle.dump(label_title_train, open(os.path.join(data_dir, "dataMsgLabelTrain"), "wb"))
pickle.dump(label_title_test, open(os.path.join(data_dir, "dataMsgLabelTest"), "wb"))
dictionary.save(os.path.join(data_dir, "dataDictSeq"))
bow_dictionary.save(os.path.join(data_dir, "dataDictBow"))
json.dump(label_dict, open(os.path.join(data_dir, "labelDict.json"), "w"), indent=4)
logging.info("done!")
```

Figure 27 : Saving file for future uses

- **Processing the input for Neural Topic Model**

  Processing the input for the NTm, which are the bow representation, index vector representation and the vectorised label array.



```
# process input
seq_train_pad = pad_sequences(seq_train, maxlen=MAX_SEQ_LEN)
seq_test_pad = pad_sequences(seq_test, maxlen=MAX_SEQ_LEN)
label_train = keras.utils.to_categorical(label_train)
label_test = keras.utils.to_categorical(label_test)

bow_train, bow_train_ind = generate_arrays_from_source(bow_train)
bow_test, bow_test_ind = generate_arrays_from_source(bow_test)
test_count_indices = np.sum(bow_test_ind, axis=1)

# build model
bow_input = Input(shape=(len(dictionary_bow),), name="bow_input")     # the normalised input
seq_input = Input(shape=(MAX_SEQ_LEN, ), dtype='int32', name='seq_input')
embedding_mat = utils.build_embedding(embedding_fn, dictionary_seq, data_dir)
emb_dim = embedding_mat.shape[1]
seq_emb = Embedding(len(dictionary_seq) + 1,
                    emb_dim,
                    weights=[embedding_mat],
                    input_length=MAX_SEQ_LEN,
                    trainable=False)
topic_emb = Embedding(TOPIC_NUM, len(dictionary_bow), input_length=TOPIC_NUM, trainable=False)
psudo_input = Input(shape=(TOPIC_NUM, ), dtype='int32', name="psudo_input")
```


Figure 28 : Processing the input for Neural Topic Model

- **Building the NTM**

```
# build encoder
e1 = Dense(HIDDEN_NUM[0], activation='relu')
e2 = Dense(HIDDEN_NUM[1], activation='relu')
e3 = Dense(TOPIC_NUM)
e4 = Dense(TOPIC_NUM)
h = e1(bow_input)
h = e2(h)
if SHORTCUT:
    es = Dense(HIDDEN_NUM[1], use_bias=False)
    h = add([h, es(bow_input)])

z_mean = e3(h)
z_log_var = e4(h)
# sample
hidden = Lambda(sampling, output_shape=(TOPIC_NUM,))([z_mean, z_log_var])
# build generator
g1 = Dense(TOPIC_NUM, activation="tanh")
g2 = Dense(TOPIC_NUM, activation="tanh")
g3 = Dense(TOPIC_NUM, activation="tanh")
g4 = Dense(TOPIC_NUM)

def generate(h):
    tmp = g1(h)
    tmp = g2(tmp)
    tmp = g3(tmp)
    tmp = g4(tmp)
    if SHORTCUT:
        r = add([Activation("tanh")(tmp), h])
    else:
        r = tmp
    if TRANSFORM is not None:
        r = Activation(TRANSFORM)(r)
```


Figure 29 : Building the NTM

1. **Building Classifier**

   Building the classifier that will work using the latent topics mapped with the labels and the text instances to learn about the working of short text analysis.

```
# build classifier
filter_sizes = [1, 2, 3]
num_filters = 512
c1 = Dense(TOPIC_EMB_DIM, activation='relu')
t1 = Dense(TOPIC_EMB_DIM, activation='relu')
f1 = Dense(TOPIC_EMB_DIM, activation="relu")
f2 = Dense(TOPIC_EMB_DIM, activation="relu")
f3 = Dense(TOPIC_EMB_DIM, activation="relu")
f4 = Dense(TOPIC_EMB_DIM, activation="relu")
f5 = Dense(TOPIC_EMB_DIM, activation="relu")
o1 = Dense(TOPIC_EMB_DIM, activation='relu')
o2 = Dense(TOPIC_EMB_DIM, activation='relu')
o3 = Dense(TOPIC_EMB_DIM, activation='relu')
o4 = Dense(TOPIC_EMB_DIM, activation='relu')
o5 = Dense(TOPIC_EMB_DIM, activation='relu')

conv_0 = Conv2D(num_filters, kernel_size=(filter_sizes[0], TOPIC_EMB_DIM), padding="valid",
                kernel_initializer='normal', activation='relu')
conv_1 = Conv2D(num_filters, kernel_size=(filter_sizes[1], TOPIC_EMB_DIM), padding="valid",
                kernel_initializer='normal', activation='relu')
conv_2 = Conv2D(num_filters, kernel_size=(filter_sizes[2], TOPIC_EMB_DIM), padding="valid",
                kernel_initializer='normal', activation='relu')
s1 = Bidirectional(LSTM(80))
s2 = Dense(CATEGORY, activation='softmax')
cls_vars = [c1, t1, f1, o1, s1, s2]
x = seq_emb(seq_input)
x = c1(x)   # reducing dim
x = Dropout(0.05)(x)
wt_emb = topic_emb(psudo_input)
wt_emb = t1(wt_emb)     # reducing dim
# first match layer
```


Figure 30 : Building Classifier

2. **Building combined model**

   Building a model combining the NTM and the classifier, that will lead to the completion of the TMN model.

```
ntm_model = Model(bow_input, [represent_mu, p_x_given_h])
ntm_model.compile(loss=[kl_loss, nnl_loss], loss_weights=[kl_strength, 1.0], optimizer="adagrad")

combine_model = Model([bow_input, seq_input, psudo_input], cls_out)
combine_model.compile(optimizer="adadelta", loss=K.categorical_crossentropy, metrics=["accuracy"])

vis_model = Model([bow_input, seq_input, psudo_input], [represent_mu, wt_emb, match, cls_out])

print_weight_shape(combine_model)
```

Figure 31 : Building combined model

4.3 Risk Analysis and Mitigation

There were some prominent and hard to surpass issues and risks that arose with this solution to the problem statement of increasing leaf diseases. Some of the issues faced are:

- Lack of proper dataset:
  - To create a proper and efficient model for the problem statement, a large dataset of short yet meaningful texts is required so as to properly analyze the patterns and to extract features in a more efficient manner.
  - Presence of public datasets is hard to find, and organizations with proper datasets either asks for a cost to pay or do not reply to the requests
  - The public dataset acquired does not contain as much text instances as required or expected. The dataset used in this project contained only about 86000 text instances.
  - The data instances present in the dataset is imbalanced in number and is way too little to properly create some efficient models.
- Lack of computation power:
  - Creating some properly working and viable high dimensional models require high processing and computation power.
  - Algorithms like LSTM for predictions and computation of embedding files in a 100-200 dimensional plane to find latent topics require a good amount of RAM as well as proper support of high GPUs.

- Lack of all these resources made us try cloud computing for machine learning in the form of
  - Google colab
  - Microsoft azure
  - Google cloud
  - Amazon web services
- But all these technologies contained problems and issues and restrictions of their own
  - Google colab:
    - Machine learning and deep learning algorithms like CNNs sometimes take more than days to complete execution and google colab provides a stable runtime session of 12 hours at one stretch.
    - Microsoft Azure is free to use for machine learning but requires an account that can be used to make worldwide transactions.
    - Google cloud and amazon web services also lead to the same problems.

Chapter 5

Testing(Focus on Quality of Robustness and Testing)

5.1 Testing Plan

The testing datasets after segregation and proper and appropriate preprocessing and analysis were passed through the respective models and the accuracy at each epoch for each of the models were observed.

The testing has been conducted using the metrics provided by scikit-learn, and accuracy has been calculated accordingly.

The final performance of the models have been mentioned below in the table.

Table 2 : Accuracy Table

Step	Model	Accuracy
1.	Bag of Words Model	~ 56%
2.	Bag of Words + n-grams Model	~ 61% - 62%
3.	Random Forest Model	~ 60%
4.	Long Short Term Memory Model	~ 67%
5.	Topic Memory Networks	~ 83%

5.2 List all test cases in prescribed format

```
[ ] y_pred[:5]

array(['MEDIA', 'DIVORCE', 'RELIGION', 'WEIRD NEWS', 'MONEY'],
      dtype=object)

[ ] y_test[:5]

5046          MEDIA
90699       DIVORCE
30914     WORLDPOST
43396    WEIRD NEWS
17987         STYLE
Name: category, dtype: object

[ ] X_test[:5]

5046     Woman's Elaborate Scheme To Discredit Washingt...
90699    Why Dating And Men Are Better When You're A Si...
30914    Rome is Berning! The Bromance Between Bernie a...
43396              Covered Wagon Turns Over, Snarls Traffic
17987                              No need to break the bank.
Name: text_instance, dtype: object
```

Figure 32 : Test Cases and Results

```
[ ] print(pd.crosstab(test['category'], y_pred, rownames=['Actual class'], colnames=['Predicted class']))

Predicted class  ARTS  ARTS & CULTURE  COLLEGE  ...  WOMEN  WORLD NEWS  WORLDPOST
Actual class                                    ...
ARTS               10               0       48  ...     30          38         27
ARTS & CULTURE     15               3       36  ...     17          73         34
COLLEGE             8               1       30  ...      1           3          5
CRIME              11               3       46  ...      7           1         11
CULTURE & ARTS      3               1       27  ...    115          11         16
DIVORCE            17               2       41  ...      3           4         15
EDUCATION           7               2       42  ...      0           1          6
ENVIRONMENT         9               2       57  ...     10           4          9
FIFTY               9               3       55  ...      3           4         15
GOOD NEWS           9               2       53  ...      3           9         29
GREEN              14               1      103  ...      9           6         11
IMPACT             26               5      490  ...      5          13         29
LATINO VOICES      15             116       34  ...      2           5         22
MEDIA              34               8       45  ...      3           5         31
MONEY              28               2       53  ...      8           2         14
RELIGION            6               4       76  ...      3          10         15
SCIENCE            17               0       44  ...      4           9         19
STYLE              23               5       31  ...      4          10        352
TASTE              27               4       41  ...      4           6         40
TECH              380               3       60  ...      2           7         33
WEIRD NEWS         26               2       29  ...      9          12         68
WOMEN              20               4       90  ...      9          15         54
WORLD NEWS          8               6       44  ...      3           7         10
WORLDPOST          18               5       72  ...      7           5         12

[24 rows x 24 columns]
```

Figure 33 : Confusion Matrix for the Test data

5.3 Error and Exception Handling

Since the problem statement being fairly new and complex, the implementation of a proper solution for the same was not easy at all. Multiple errors were generated during the journey and many were hard to seek.

- Use of new libraries, like gensim, and functionalities created a whole lot of issue, as gensim being a scientific library for text analysis, was hard to implement
- Handling of the text input in multiple ways for various different models, created different issues of preprocessing and conversion of the data in the proper format.
- Errors regarding noises in the dataset and abnormalities and anomalies in the dataset were handled in a professional manner.

5.4 Limitations of the Solution

- As of now, due to the lack of proper computation power, and high GPUs, proper training of the model is difficult to expect
- As topic memory networks are the latest and most advanced version of memory networks suitable for this problem statement, not very much of a study or research has been done on it. Thus the current implementation is hard to be incorporated in practical applications for the marketplace and require more complex research for optimisation.
- Since most of the small text data generated is personal to people, it's hard to find proper quantities of viable data to develop advanced models and expect good results.
- Due to development of sparse matrix containing the bow representation of the tokenized text instances, overfitting can happen really easily, and can be a really hard nut to crack at the time of optimisation.
- Finding the latent topics by using pre-trained word embeddings provided by Harvard university can be really hard to use, as the embeddings are marked in a super high dimensionality space of about 100 - 200 dimensions.

Chapter 6

Findings , Conclusion and Future Work

6.1 Findings

After an in-depth analysis of the datasets, using the various algorithms scaling at different levels of difficulty and complexity, we found our some pretty decent and varied results:

- The current implementation of text classification using the traditional bag of words representation provided an accuracy of about 56%.
- After increasing the associated information for the text instances using unigrams and bi-grams, the accuracy of the model improved but not significantly. The new model provided an accuracy of about 61% - 62%.
- Using Random Forest classifier over the vectorised data samples, so as to form some complex correlations didn't work the way it was expected to. The model accuracy did not improve, due to overfitting and thus remained about 61%.
- Use of memory networks improved the overall accuracy, but Long Short Term Memory Networks did not work exceptionally well. In Fact the improvement in the accuracy over random forest and bow model was about 6% - 7%, thus, providing an overall accuracy of about 67%.
- The main highlight of the research and implementation, the topic memory networks worked exceptionally well by providing an accuracy of about 87%.

Thus, short text data can be really tricky for the machine to understand, and thus normal and traditional machine learning models will not work well over it. But implementing the latest memory networks can handle the problem for us.

6.2 Conclusion

The target of the project was to help the current machines become more smart , interactive and user friendly by enabling them to understand our textual input, be it of any length long or small. Our implementation of the less researched algorithm like topic memory network, will help others to implement this algorithm in production of market specific products, as they all will be having a proper documentation supporting the practical implementation of the same.

Traditional algorithms like bag of words model, random forests and naive bayes algorithm have proved to be of great support, but have a strict restriction of the availability of proper length text instances, thus they will not work properly over the short and simple text inputs created by our generation these days. As a solution, using a more smart and complex algorithm, that can work in ways which are much more similar to those of the human brain will significantly change the results in our favour. Memory networks are such algorithms, specially topic memory networks, which not only finds the hidden topics in the texts but also maps the latent topics to the inferred classes.

The solution offered through our approach plus the methodologies that we have learned from the research papers will help the researchers and developers incorporate some more user friendly actions in their products, improving the human interaction experience ever further.

6.3 Future Work

Short text classification being a new topic in the field of text analysis and classification , requires much more research and practical applications. Future work for this project are:

- Implementation of other traditional text classification algorithms like naive bayes and artificial neural nets, has to be done, so as to compare the results more thoroughly.
- Algorithms like GLOVE, which are used to create word embeddings in dense dimensional spaces, have to be implemented to improve the process of finding latent topics.
- Collection of more varied and densely populated dataset is required so as to improve the training of the model to an extent.
- Optimisation of memory networks by tweaking the types of layers, and the number of layers involved in the neural nets has to be done in the future.
- Incorporation of the current solution into one of the user interactive software applications.

References

[1] F Sebastiani, VG Moruzzi, Research in Automated text Classification: trends and perspectives, *Actas del 6º Congreso del Capítulo Español de*, 2003

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Extreme Multi-Label Legal Text Classification: A case study in EU Legislation, *Proceedings of the Natural Legal Language Processing Workshop* 2019, pages 78–87

[3] Sriram B., Fuhry D., Demir E., Ferhatosmanoglu H., & Demirbas M, Short text classification in twitter to improve information filtering, *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* - SIGIR 2010

[4] Monner, D., & Reggia, J. A.. Systematically grounding language through vision in a deep, recurrent neural network. *To appear in Proceedings of the conference on artificial general intelligence,* 2011

[5] Zhang, Y., Jin, R., & Zhou, Z.-H.. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics,2010, 1(1-4), 43–52.*

[6]Johannes Furnkranz, A Study Using n-gram Features for Text Categorization, , *Artificial Intelligence* , 1998

[7] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Re- trieval*, SIGIR '17, pages 115–124.

[8] Jeffrey Pennington, Richard Socher, and Christo- pher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Nat- ural Language Processing (EMNLP)*, pages 1532– 1543.

[9] Banerjee, S., Ramanthan, K., and Gupta, A. Clustering short text using Wikipedia. *In Proc. SIGIR (Amsterdam, The Netherlands)*, 2007, 787-788.

[10] Phan, X.-H., Nguyen, L.-M., and Horiguchi, S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *In Proc. WWW (Beijing, China)*, 2008, 91-100.

[11] Monner D., & Reggia J.A., A generalized LSTM-like training algorithm for second order recurrent neural networks. *Neural Networks, 25*, 2012, 70-83

[12] Gers, F. A., Pérez-Ortiz, J. A., Eck, D., & Schmidhuber, J.. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks, 16*, 2003, 241–250.

[13] Schmidhuber, J., Wierstra, D., Gagliolo, M., & Gomez, F., Training recurrent networks by Evolino. *Neural Computation, 19*, 2007, 757–779.

[14] Sebastiani, F., Machine learning in automated text categorization. ACM Computing Surveys, 34(1), pp. 1–47, 2002.

[15] Yang, Y. & Pedersen, J.O., A comparative study on feature selection in text categorization. Proceedings of ICML-97, *14th International Conference on Machine Learning, ed. D.H. Fisher, Morgan Kaufmann Publishers*, San Francisco, US: Nashville, US, pp. 412–420, 1997.

[16] Nowak E, Jurie F, Triggs B.Sampling strategies for bag-of-features image classification. In: *Proceedings of the 9th European conference on computer vision*, Graz, Austria,2006, pp 490–503
[17] Slonim, N. & Tishby, N., The power of word clusters for text classifi- cation. *Proceedings of ECIR-01, 23rd European Colloquium on Infor- mation Retrieval Research*, Darmstadt, DE, 2001.

Brief Bio-data (Resume) of the Students

1. Aman Parmar (16103221)

Last Updated on 7th May 2020

Aman Parmar

+91-8130832780 | aman.parmar17@gmail.com

EDUCATION

JAYPEE INSTITUTE OF IT
BTech in Computer Science
Expected Jun 2020 | Noida, India
CGPA: 7.3/10

KARAN PUBLIC SCHOOL
AISSCE
May 2016 | Meerut, India
Percentage: 93.8%

LINKS

Github:// **amanparmar17**
LinkedIn:// **Aman Parmar**
EyeEm:// **aman parmar**
Quora:// **Aman Parmar**

COURSEWORK

Advanced Machine Learning
Python Programming
Neural Networks and Deep Learning
Machine Learning with Python
Deep Learning Fundamentals
Unix Tools and Scripting
Data and Web Mining

SKILLS

PROGRAMMING AND TOOLS

Experienced
- Core Java
- Python
- Data Structures
- Machine Learning
- Deep Learning
- Data Analytics
- Microsoft Excel
- MongoDB

Intermediate
- Natural Language Processing
- MySQL
- Tableau
- Django
- Text Mining

Familiar
- HTML
- CSS
- JavaScript
- PHP

OPERATING SYSTEMS
- Windows
- Linux

EXPERIENCE

FORSK TECHNOLOGIES | Machine Learning and Data Science Trainee
May 2019 - July 2019 | Jaipur, India
- Developed and maintained positive customer relations and coordinated with team members to ensure requests and questions were handled appropriately.

BANK OF INDIA | Business Analyst Intern
Dec 2018 – Jan 2019 | Muzaffarpur, India
- Revamped plans to enhance company's capability of maintaining and recovering critical business functions.
- Organized and updated strategies for managing crisis events to modernize and enhance approach.

TRENDING SKILL | Full Stack Developer (Intern)
Jun 2018 – Aug 2018 | New Delhi, India
- Worked on existing web application to correct coding errors, upgrade interfaces and improve overall performance.
- Collaborated with cross-functional development team members to analyze potential system solutions based on evolving client requirements.

PERSONAL PROJECTS

CULTIVO - SMART CROP CONSULTANT | (Python | Django | Machine Learning)
Oct 2018 – Nov 2018
- A machine learning and web-based product, trained extensively over large dataset to provide many valuable insights about the crop
- Use of algorithms like Random Forest, SVM, Polynoial Regression, custom built batch processing and k-fold cross validation.

ADVANCE LEAF DETECTION AND DISEASE PREDICTION | (Python | Neural Nets)
Aug 2019 – Sept 2019
- Advanced image classification and web scraping project built to identify the leaf and detect any disease present.
- Advanced image classification and web scraping project built to identify the leaf and detect any disease present.

SMALL TEXT CLASSIFICATION | (Python | NLP | Neural Nets)
March 2020 – May 2020
- Use of various complex and custom machine learning and natural language processing algorithms like, Random Forest, Naive Bayes, BERT, BF, LSTM and Topic Memory Networks, to classify short length text instances.

INTERESTS

- Photography
- Blogging
- Reading
- Studying
- Social Work

2. Palak Arora (16103046)

PALAK ARORA

Jaypee Institute of Information Technology, Noida

palakarora1401@gmail.com +91-9785375117 github.com/palakarora14 linkedin.com/in/palak-arora/

ABOUT ME

- A creative and logical analyzer who can be trusted to come up with new ideas and solutions.I like sharing knowledge with my fellow classmates.I like programming, Android Development and a quick learner who is always eager to learn new things.Want to work in an environment that will let me develop new skills, while being efficient with what I already know.

EXPERIENCE

National Informatics Center

Engineering Intern

May 2019-July 2019

- Developed and designed an Android Application for Treatment prediction of certain diseases.
- Worked towards getting optimal solution by removing server dependency for testing on ANN model.
- **Technologies Used: Android Development,Java,Python,Deep Learning**

PROJECTS

Medical Analyser

Android application - ANN model - Web crawler

- An application which will take Symptoms as input from the user and will predict the disease according to it (with the help of Artificial Neural Network model embedded in it).Further showing the treatment , precautions , management measures etc. for the predicted disease (done with the help of web crawler).
- This app shall prove useful for the doctors/patients to find out there precise disease and its treatment in no time with the help of few clicks.

Palan : The Shopping App

eCommerce application - Image processing - Payment gateway

- It is an e-commerce app that works efficiently for a user to choose between various products available on the application , categorised into three categories namely Women,Men,Accessories . Add then to cart and then pay for it successfully using InstaMojo payment gateway .
- Also, if the user provides an item image it will recommend its availability on the application ,by using Convolution Neural Network model.

ContactBook

Contacts stored/retrieved like a dictionary

- This project represents the basic functionalities of a phonebook i.e inserting new contacts , find/recommendation of already existing contacts etc.
- All implemented through Trie data structure .

PUBLICATIONS

Arora, P., Singh, S., & Rathi, M.(in press). ANN incorporated in Android application for treatment prediction of diseases. International Journal Taylor & Francis .

EDUCATION

JIIT, Noida

Bachelor of Technology, Computer Science

July 2016 - Present

CGPA: 7.6 / 10

GRM Sr. Sec. School, Bareilly

Senior Year (Class XII)

May 2015

Percentage: 94.3%

GPM College, Bareilly

Class X

May 2013

Percentage: 83.33%

SKILLS

Technical Skills

•C++/C•Data Structure •Algorithms •Android •Deep Learning(Neural Networks) •Object Oriented Programming• MySql

Soft Skills

•Communication Skills •People Management • Analytical and Reasoning ability

COURSES / TRAINING

- Deep Learning, Coursera
- Android App Development, Internshala

POSITION OF RESPONSIBILITY

- Co-Ordinator : Game Development Hub
- Volunteer : Its Our Earth

AWARDS/HONOURS

- Won the Title of Miss Impressions 2016
- Secured 91% percentile in National Genius Search Olympiad (Mains) 2012.

3. **Anjali Sharma (16103015)**

ANJALI SHARMA

4th year Undergraduate (Computer Science) Jaypee Institute of Information Technology, Noida

anjali.ss.sharma8@gmail.com Delhi, India linkedin.com/in/anjali-sharma-9a028716a +91-9560671151

Career Objective:

To work in a challenging and competitive environment, that provides a vision to analyse situations and people, to prove my potential.

Education:

Bachelor of Technology (CSE)
Jaypee Institute of Information Technology, Noida
2016- Present CGPA 7.0/10.0

Senior Secondary, CBSE
Mother's Global School, Delhi
2016 Percentage 93.8

Secondary, CBSE
Mother's Global School, Delhi
2014 CGPA 8.8/10.0

Internships:

Software Intern- ORC (2019)
Oracle India Pvt. Ltd.
Backport Request Application (Oracle APEX).

Industrial Trainee (2018)
Delhi Metro Rail Corporation
Revenue Register Project (HTML, PHP, MySQL, XAMPP)

Social Media Manager (2017)
Travart Travel and Explorations

Skill Set:

- C and C++
- Database and Web
- Data Structures and Algorithms
- MySQL, PHP.

Key Projects:

SONG SUCCESS ANALYSER (R Language)
- Predictive analysis of commercial success of music tracks based on lyrics and other metrics using classification models.

PALAN SHOPPING APP (JAVA AND KOTLIN)
- An e-commerce app for shopping with image detection (CNN) and payment gateway through Firebase.

REVENUE REGISTER (HTML, PHP, MYSQL)
- Web application to calculate revenue generated per day per station of Delhi Metro. Backend connection using XAMPP server.

BACKPORT REQUEST APPLICATION (APEX).
- Oracle APEX application to track the backport requests, with BugDB integration using REST API.

Mini Projects:

- Medical Shop Management System (C Language)
- SEO and Word Recommendation (In C++ using Data Structures)
- K-means clustering using iris dataset (In C++ using Algorithms)
- Dinner Decider Android Application (Java)
- Restaurant Management System Android Application.

Achievements:

- Co-ordinator: JIIT Game Development Hub.
- Marketing Head: Tedx JIIT, Tech Fest 2019.
- Internshala Student Partner 9.0 member.
- Former member of student chapter of Optical Society of America.
- Volunteer: Sankalp NGO, Duayein NGO.

Plagiarism Check Summary :

16103046_SHORT . TEXT CLASSIFICATION USING MEMORY NETWORKS

ORIGINALITY REPORT

14%	10%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ojs.academypublisher.com Internet Source	2%
2	Submitted to Indian Institute of Technology Guwahati Student Paper	2%
3	documents.mx Internet Source	1%
4	www.ijert.org Internet Source	1%
5	eprints.whiterose.ac.uk Internet Source	1%
6	link.springer.com Internet Source	1%
7	www.slideshare.net Internet Source	1%
8	marswebsolutions.files.wordpress.com Internet Source	1%

9	de.slideshare.net Internet Source	1%
10	faure.iei.pi.cnr.it Internet Source	1%
11	Submitted to Texas A & M University, Kingville Student Paper	<1%
12	"Image Colour Prediction using Deep Learning", International Journal of Recent Technology and Engineering, 2020 Publication	<1%
13	pt.slideshare.net Internet Source	<1%
14	www.cs.bilkent.edu.tr Internet Source	<1%
15	embeddedtechnosolutions.com Internet Source	<1%
16	Submitted to Maulana Azad National Institute of Technology Bhopal Student Paper	<1%
17	Monner, D.. "A generalized LSTM-like training algorithm for second-order recurrent neural networks", Neural Networks, 201201 Publication	<1%
18	Submitted to Chester College of Higher Education Student Paper	<1%
19	Submitted to University of Edinburgh Student Paper	<1%
20	Submitted to De Montfort University Student Paper	<1%
21	Submitted to Syracuse University Student Paper	<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		