Topic Change Detection in Textual Documents Using Natural Language Processing

B. Tech. Project Stage-II Report

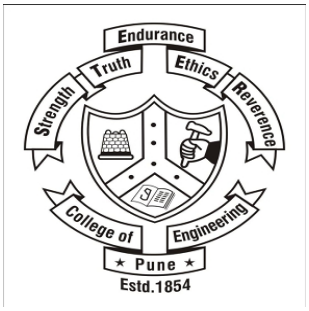
Submitted by

Omkar Ashok Bairagi	**111803039**
Pradeep Uttamrao Bhalerao	**111803043**

Under the guidance of

Prof. Vaibhav Khatavkar

College of Engineering, Pune



DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

April-May 2022

DEPARTMENT OF COMPUTER ENGINEERING

AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

Certified that this project, titled "Topic Change Detection in Textual Documents Using Natural Langauge Processing" has been successfully completed by

Omkar Bairagi 111803039

Pradeep Bhalerao 111803043

and is approved for the partial fulfillment of the requirements for the degree of "B.Tech. Computer Engineering".

SIGNATURE

Prof. Vaibhav Khatavkar

Project Guide

Department of Computer Engineering and Information Technology,

SIGNATURE

Dr. Vahida Attar

Head

Department of Computer Engineering and Information Technology,

College of Engineering Pune,
Shivajinagar, Pune - 5.

College of Engineering Pune,
Shivajinagar, Pune - 5.

Abstract

We present the difficult challenge of spotting topics and detecting significant changes in text paragraphs. Our objective is to recognise major topics related to a specific part of paragraph, rather of detecting and tracking events as in the TDT configuration, focus the detection of major changes inside an existing event or narrative. Change point detection approaches have been used in the past to detect major changes in sensor signals, but they do not leverage the textual content of textual streams. In order to construct a time series, we first perform linguistic preprocessing or gather simple statistics on the messages in the discussion. Topic modelling and transformation techniques are then applied on it. We present a collection of transcripts of video lectures and evaluate the approach of our method. Using a variety of topic modelling methods and time series, we show that it is feasible to detect significant changes in an online discussion with a high degree of accuracy. Monitoring complex systems that generate high-volume, high-velocity streaming data may be a difficult undertaking. While applicable to a wide range of domains of difficulty, it is especially useful for monitoring of high-value assets and important engineering resources.

Final Report

ORIGINALITY REPORT

12%	9%	7%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	nrc-publications.canada.ca Internet Source	2%
2	aniketbhadane.github.io Internet Source	2%
3	Viktar Atliha, Dmitrij Šešok. "Text Augmentation Using BERT for Image Captioning", Applied Sciences, 2020 Publication	1%
4	Bernadin Namoano, Christos Emmanouilidis, Cristobal Ruiz-Carcel, Andrew G Starr. "Change detection in streaming data analytics: A comparison of Bayesian online and martingale approaches", IFAC-PapersOnLine, 2020 Publication	1%
5	www.coursehero.com Internet Source	1%
6	downloads.hindawi.com Internet Source	1%

13/04/22

Figure 1: Scanned copy of plagarism percentage approved by project guide

Contents

List of Figures

Chapter 1

Introduction

Massive streams of textual data are being generated by social media, microblogs, and news sources. These streams are impacted via real-world events and changes. To recognize these changes, autonomous approaches based on the basis of a hybrid of natural language processing and statistical modelling must be used to monitor message streams. Topic extraction swiftly identifies the important words and concepts in an article or document to give us the substance of it. However, unlike categorization or entity extraction, topic extraction is not bound by a finite number of recognised entity types or categories. Instead, the topic endpoint determines "keys" and "concepts" for the provided input based on frequency and linguistic patterns in the text, ranking them in order of relevance. On a larger scale, the same technique may be used for a corpus of papers in order to comprehend the main ideas. Understanding the important terms and concepts in each document allows users to automatically classify, categorise, and arrange their data, making it more helpful to analysts and database administrators.

Chapter 2

Problem Statement

2.1 Existing Solution

The vast majority Topic Detection and Tracking has been the subject of a lot of research focuses on detecting emergent events. Laban and Hearst (2017), for example, gathered 4 million news items, produced news themes, integrated them into stories, and showed the tales along a timeline. Atkinson et al. (2017) used lexico-semantic patterns to construct a corpus of security-related events derived from news data, and then categorised events into security-related categories. Recent research has focused on recognising shifts in a tale. For example, Bruggermann et al. (2016) used the dynamic topic model (Blei and Lafferty, 2006) to identify subjects from news, and changes in the word distributions of the topic model were used to reflect changes in the storyline. Wang and Goutte (2017) additionally evaluated their findings on two Twitter datasets, using the temporal profile of hashtags in tweets to detect changes inside events.

2.2 Problem Statement

To develop the Topic analysis System in textual corpus Using Natural Language Processing to extract dominant topics in textual corpus and visualise

on timeline.

Chapter 3

Literature Review

1. Real-time Change Point Detection using On-line Topic Models (2018) [11]

This paper was part of the 27th International Conference on Computational Linguistics. The methodology used in this paper was :

(a) Data Collection and Preprocessing

(b) LDA modelling to detect topic shifts

(c) Output

This paper uses a mix of topic modelling and bayesian change point detection techniques. This research paper proposes a way for live identification of change in topic in data stream. Four techniques of depicting main topic change time series were investigated, as well as many text or paragraph modes. This LDA and OCP+ deliver excellent results, detecting at most 56 percent of changes of reference in live mode. Other alterations went unnoticed.

2. Detection of Bayesian Changeopint in Textual Data(July 2015) [3]

This paper was published in 2015 at the International Conference on Topic Detection in text data. The methodology used is :

(a) Data Preprocessing

(b) Text Mining

(c) BCPD model to detect topic shifts

(d) print the output

Text mining is the process of collecting meaningful information from textual data sources, and it has a wide range of applications in many disciplines. Statistical changepoint detection methods can provide a new tool for temporal text analysis, uncovering interesting trends in data across time. In this study, a general real-time changepoint detection system was developed to cope with streams of textual data for two unique tasks: recognising topic changes and detecting author changes.A synthetic corpus, as well as two real corpora: State of the Union addresses and Twitter tweets, are used to evaluate the system's performance.

3. Evaluating Change Detection in Online Conversation (2018) [10]

This research paper was published in 2018 at the International Conference for Language Resources and Evaluation (LREC 2018). The methodology used in this paper is as follows :

(a) Linguistic Data Preprocessing

(b) Change point detection algorithm

(c) Detecting Topic Shift

(d) Output

This paper contributes two change point detection versions, OCPD and OCPD+, which may be utilised in a completely online manner, allowing the identification of changes to occur as soon as they occur rather than awaiting the analysis of the time series. This is a critical function for real-time moni-

toring of social media feeds. The detection performed by OCPD+ has a high degree of precision, but it also misses multiple opportunities for real-time detection.

4. Pilot Study for Topic Detection and Traking (1998) [1]

This research paper was published in 1998 at the Workshop on Broadcast News Transcription and Understanding, sponsored by DARPA. The methodology used in this paper is as follows:

(a) Segmentation of Data Streams

(b) Identification of new event/topic occurred

(c) prints the small number of new event/topic

The overall TDT task was investigated in this research study, and important technological obstacles were highlighted. This document describes these tasks, as well as the performance metrics that will be used to assess progress. Segmentation is a manageable process when employing well-known technology. For most occurrence, pure retrospective detection may be achieved relatively accurately using clustering algorithms

5. For textual subject detection, a soft frequent pattern mining technique is used. (2014) [4]

This paper was published in 2014 at the 4th International Conference. The methodology used in this research paper is as follows

(a) Data Cleaning and Data Preprocessing

(b) Feature Pivot Method to detect topic change

(c) Similarity measure to detect direction of topic

When the data consists of a group of closely related fine-grained themes, the research report proposes that investigating the simultaneous co-occurrence a vast number of word patterns is a preferable alternative.Furthermore, we treat

the topic detection issue as a frequent pattern mining problem and present a novel'soft' Frequent Pattern Mining technique.We compare the proposed method a collection of techniques that incorporates a feature-pivot based on graphs strategy that only evaluates co-occurrence patterns and put it to the test using three Twitter datasets that have been annotated.

6. New Event Detection with Text Classification and Named Entities (2004) [2]

This paper was published in 2004 at the 27th Annual International ACM. The methodology used in this paper is as follows :

(a) Data Categorization

(b) Novel way to detect new topic(NED)

(c) Result

This study shows how text classification techniques, as well as named things, may be used in a unique way, may increase performance on New Event Detection (NED). In addition,the research investigates changesin a vector space-based NED system to the document representation Preferentially addressing named entities is only beneficial in a few situations. A multi-stage NED system is created by combining all of the aforementioned. That considerably outperforms basic single-stage NED systems.

7. Topic Detection and Tracking Techniques on Twitter : A Systematic [15]

This paper was published in 2021 by Hindawi aspart of publishing collaboration with JohnWiley Sons,Inc. It is a fully Open Access journal produced under the Hindawi and Wiley brands. This paper has following methodology : (a) Data Classification and Preprocessing (b) Post-detect approach as

deep learning short sentence categorization (c) Detect new topics Categorization based on the technique of the reliant algorithms suggested in this research is included in the publication. Finally, this paper explored a potential post-detection approach known as deep learning short sentence categorization,which can be effective after an event has been detected. Compared variousTEDT algorithms for various types of datasets and found the best method for detection and tracking of events.

8. A Comparison of Martingale Approaches and Bayesian Online to Change Detection in Streaming Data Analytics[13]

This paper was published in 2020 by Elsevier, The IFAC Publisher, IFAC-Papers online series hosted at the ScienceDirect web service. The methodology used in this paper is as follows :

(a) Data Cleaning and Preprocessing

(b) Bayesian Online Change Detection method

(c) Print Detected Topic

Two algorithmic strategies for identifying changes in streaming data are empirically evaluated in this paper. A modified martingale is employed, as well as the Bayesian Online Change Detection (BOCD) approach. The research presents a modified martingale method for detecting change in multidimensional streams, as well as an empirical assessment of both the modified and unmodified martingale and Bayesian online change detection methods.

9. Detecting Topic Shifts in Online Discussions Using Structural Context[12]

This research paper was published in 2019 at IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). The methodology used

in this paper is as follows :

(a) Data Classification and Preprocessing

(b) Topic shift detection model to detect topics

(c) Print Topics

This study provides a model of topic change detection that for vector representation uses word embedding to build the semantic link between comments and posts. It enriches the background knowledge for each remark by using the tree structure in which each conversation thread is organised automatically presents as context information. Word embedding, as when compared to word occurrence measurement, can enhance topic shift recognition accuracy.

10.Sentiment Analysis and Topic Detection Using the BNgram Method on Twitter Dataset[7]

This research paper was published in 2015 at the 4th International Conference on Reliability, Infocom Technologies and optimization (ICRITO). The methodology used in this paper is as follows :

(a) Data Cleaning and Preprocessing

(b) BNgram method to detect trending topics

(c) Print the topics

This study uses BNgram, a new topic identification approach, to big Twitter Datasets. We may also find people's reactions to various occurrences. Analysing big Twitter Datasets to assess the performance of multiple approaches,as well as the BNgram topic detection method for quickly discovering themes. The quality of discovered subjects was highly influenced by data pre-processing and sampling processes.

Chapter 4

Methodology

4.1 Purpose

Vast amount of information is produced each second on various social media as well as educational platforms, but humans have some limitations for how much data we can process in a given time bound. Hence there is a need for a system which can process this data and give a brief information about it. Hence the reason of study is to make a system which will detect the dominant topics from textual data and return the topic of it at specific period using Natural Language Processing.

4.2 Overall Description

4.2.1 Product Perspective

The main objective of this study is to develop a text analysis system for textual data corpus. This system can discover the dominating topic from huge data of conversation like news articles or video transcripts. The system aims to exploit data mining techniques and NLP techniques for text embedding, topic detection and change point.

Some Specific objectives of the system:

i. To push human limits to process Big Data.

ii. To overcome ambiguity, misspelling, grammatical mistakes and expressions from the data.

iii. To improve performance of system using appropriate algorithm

4.2.2 Product Functions

Data Collection: Dataset for this system can be collected using Web Scraping from various news sites or transcripts of learning platforms like coursera and pickling it into one single file.

Data Preprocessing and cleaning: Preprocessing steps like stop words are removed and punctuation, removal of handels and URL, stemming, tokenization can be accomplished through the use of language preprocessing techniques such as topic models.

Analysis: Our purpose is to perform topic analysis in two steps:

- **Document-level**: The topic model extracts the many themes from a whole text.
  e.g. topic of an mail or report.

- **Sentence-level**: The topic model is used to determine the subject of a single sentence. e.g. topics of report headlines.

Detection: We presume that data preceding the change point conforms to one distribution in change point detection, Data after the change point is derived from a second, distinct distribution. In our scenario, we want to figure out where the change happened as quickly as feasible after it happened.

4.3 Approach

First, rather than detecting and tracking occurrences as in the TDT configuration, we focus on discovering major changes inside an existing event or plot. Second, rather than extracting descriptive terms from the text, we concentrate on recognising the locations of major changes in the message stream. We also employ linguistically driven signals generated from text analysis pre-processing instead of external signals received from sensors or signals to identify changes.

4.3.1 Steps and Algorithm

i. Preprocessing :

a) Removing punctuations and URLs : All punctuation from the text is deleted in this stage. Python's string library has a predefined collection of punctuation marks, such as '!'

b) Removing Stop words : Stopwords are often used words that are taken out of the text since they bring no value to the analysis. These terms have little or no meaning.

c) Lower casing : One of the most typical preprocessing procedures is to convert the text to the same case, ideally lower case. However some documents may lose some information.

d) Tokenization : The text is divided into smaller units in this step.

e) Stemming/Lemmatization : It is referred as text standardisation The words are stemmed or reduced to their root/base form in this phase. e.g. 'eater', 'eating', 'ate' will be stemmed to 'eat'.

f)TF-IDF : Frequency of Term TF-IDF stands for Time Frequency Inverse Document Frequency. It is described as determining how relevant a word in

a series or data-set is to a text. The meaning grows based on the amount of times a word occurs in text, but this is offset by the data-word set's frequency.

ii. Topic Detection(Topic Modelling) :

a) LDA : LDA is an abbreviation for Latent Dirichlet Allocation, and it is based on the notion that each document is generated by a statistical process. That is, each text is composed of a number of themes, each of which is composed of a number of words. To produce this document, first select a subject from the document-topic distribution, and then select a word from the multinomial topic-word distributions from the selected topic.

Documents are represented by LDA as a collection of themes. In the same way, a topic is a collection of words. If a word has a high likelihood of appearing in a topic, all documents containing w will also be more strongly correlated with t. Similarly, If w is not very likely to be in t, papers including w will have a very low chance of being in t, because the rest of the words in d will belong to a different topic, and d will have a greater probability for those topics. As a result, even if w is added to t, it will not bring many similar papers to t.

b) BERT : Bidirectional Encoder Representations from Transformers (BERT) is an acronym for Bidirectional Encoder Representations from Transformers. BERT makes use of Transformer, which is an attention mechanism that learns contextual associations between words (or sub-words) in a text. Transformer, at its most basic version, made up of two distinct procedures:an encoder for reading text input and a decoder for generating a job prediction. Because the purpose wants construct a language model, just conversion from one format to another format procedure is required.

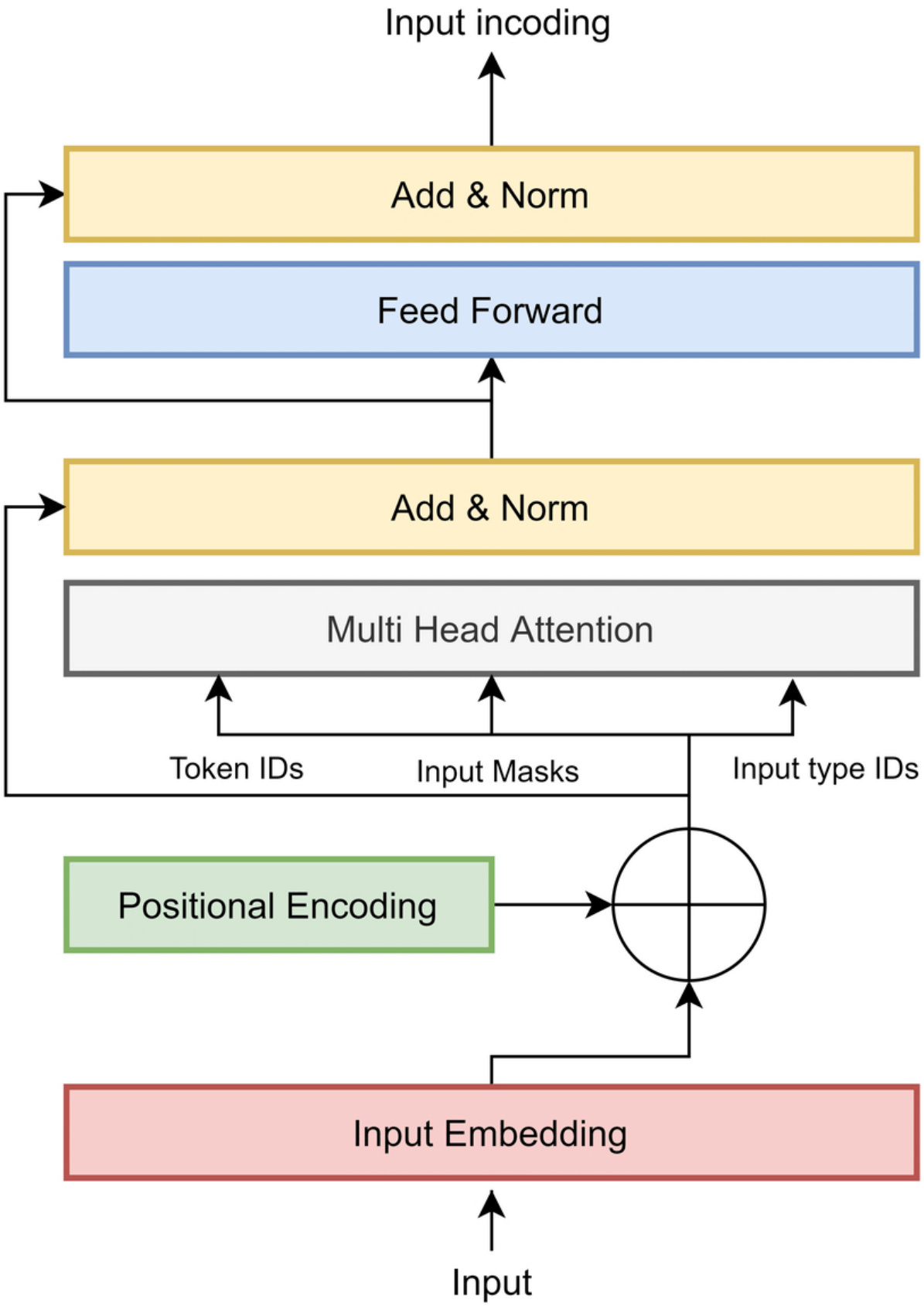


Figure 4.1: Transformer Encoder Architecture BERT

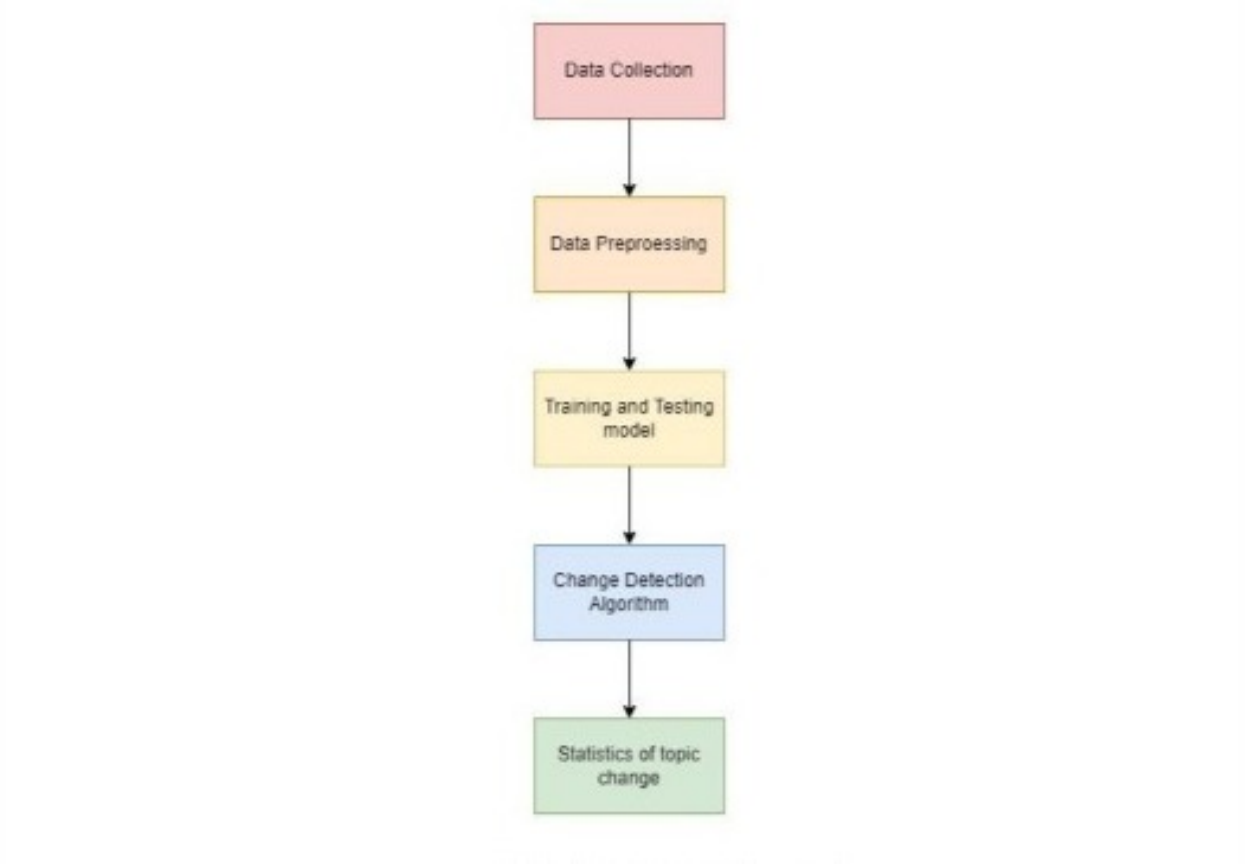


Figure 4.2: Methodology Workflow

4.3.2 Datasets

Data is the most important part of any automation process involving machine learning. The accuracy of the model heavily depends on data. The better the collection of the dataset, the better will be the accuracy. We can't train any model without data, therefore all of today's research and automation will be for naught.

Web scraping will be the main source for obtaining dataset. Datasets for study can also be obtained using directly pasting in the ".txt" file.

i) Web Scraping using python:

Python contains a large library of libraries, such as Numpy, Matplotlib, and Pandas, that provide methods and services for a variety of uses. As a result,

```
In [1]: # Web scraping, pickle imports
import requests
from bs4 import BeautifulSoup
import pickle

# Scrapes transcript data from scrapsfromtheloft.com
def url_to_transcript(url):
    '''Returns transcript data specifically from scrapsfromtheloft.com.'''
    page = requests.get(url).text
    soup = BeautifulSoup(page, "lxml")
    text = [p.text for p in soup.find_all('p')]
    print(url)
    return text

# URLs of transcripts in scope
urls = ["https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/gNXI3/vocabulary-feature-extraction",
        "https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/dDdRc/welcome-to-the-nlp-specialization",
        "https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/cITmZ/negative-and-positive-frequencies"]

In [2]: transcripts = [url_to_transcript(u) for u in urls]

https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/gNXI3/vocabulary-feature-extraction
https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/dDdRc/welcome-to-the-nlp-specialization
https://www.coursera.org/learn/classification-vector-spaces-in-nlp/lecture/cITmZ/negative-and-positive-frequencies
```

Figure 4.3: Web Scraping

it's suitable for site scraping and additional data manipulation.

ii) Directly Storing into .txt files

4.3.3 Preprocessing:

Because the machine learning model does not operate with categorical or null values, null values must be removed from the input and categorical values must be encoded beforehand. Numerical values must also be normalised. Finally, the dataset must be divided into two parts: a train dataset and a test dataset.

4.3.4 Training and Testing Model:

After the data preprocessing this dataset is provided to the models used for Exploratory Data Analysis (EDA) to check if our data makes sense.

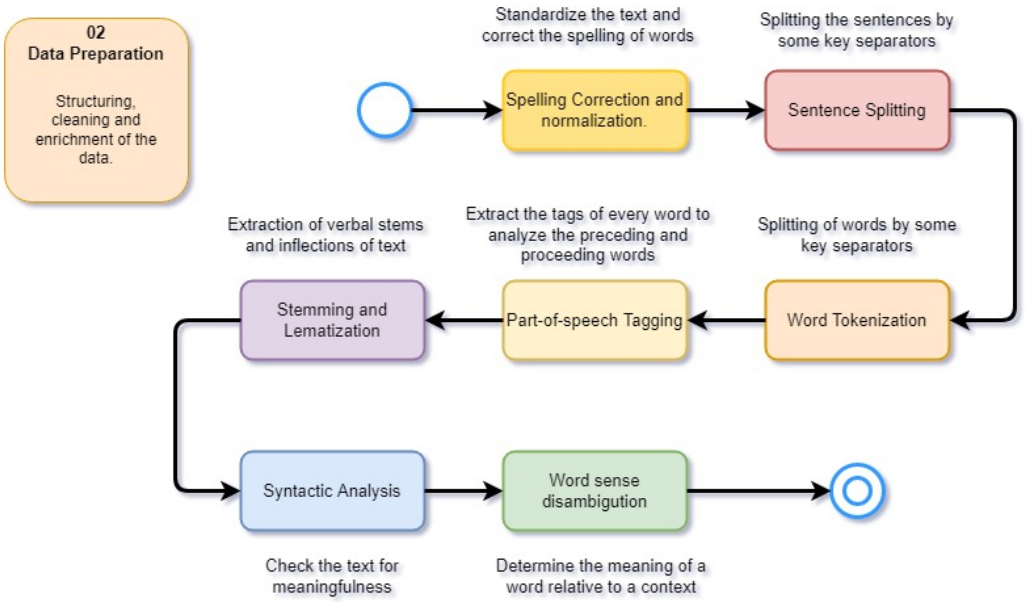


Figure 4.4: Data Preprocessing

4.3.5 Applying LDA:

Understanding LDA in its entirety necessitates a thorough understanding of sophisticated mathematical probability concepts. However, the underlying concept is simpler to grasp. LDA expects that documents are produced in the following manner: Choose a variety of themes (for example, 20This will give the dominating topics related to each group of texts. We will apply further processes on it.

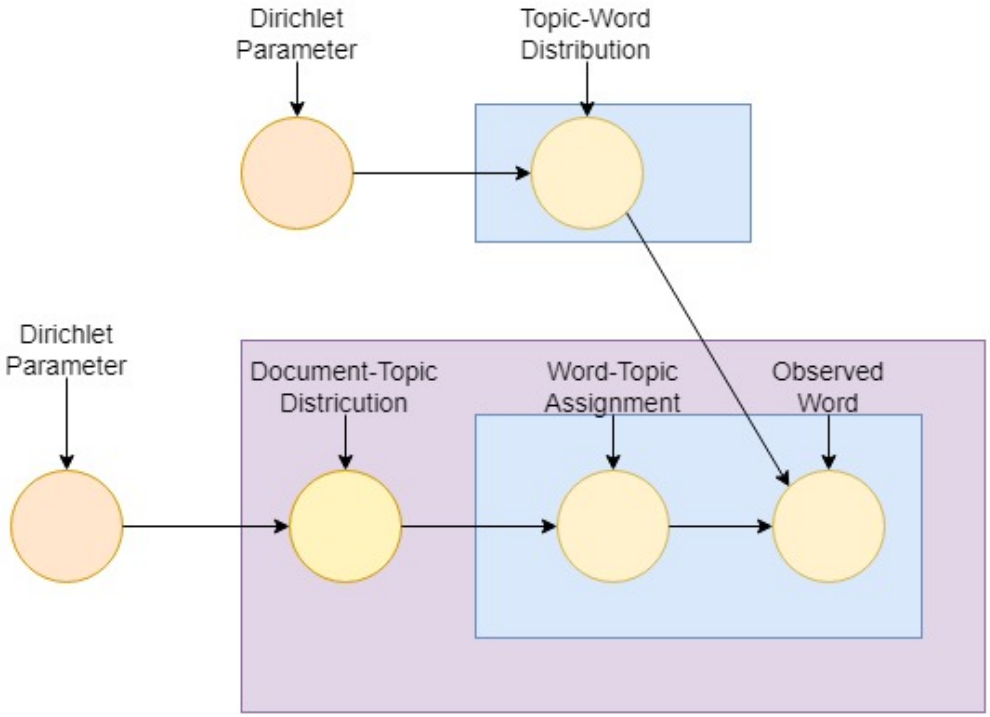


Figure 4.5: LDA Algorithm WorkFlow

Chapter 5

System Diagrams

1)System Architecture

2) Workflow

3) Data Preprocessing:

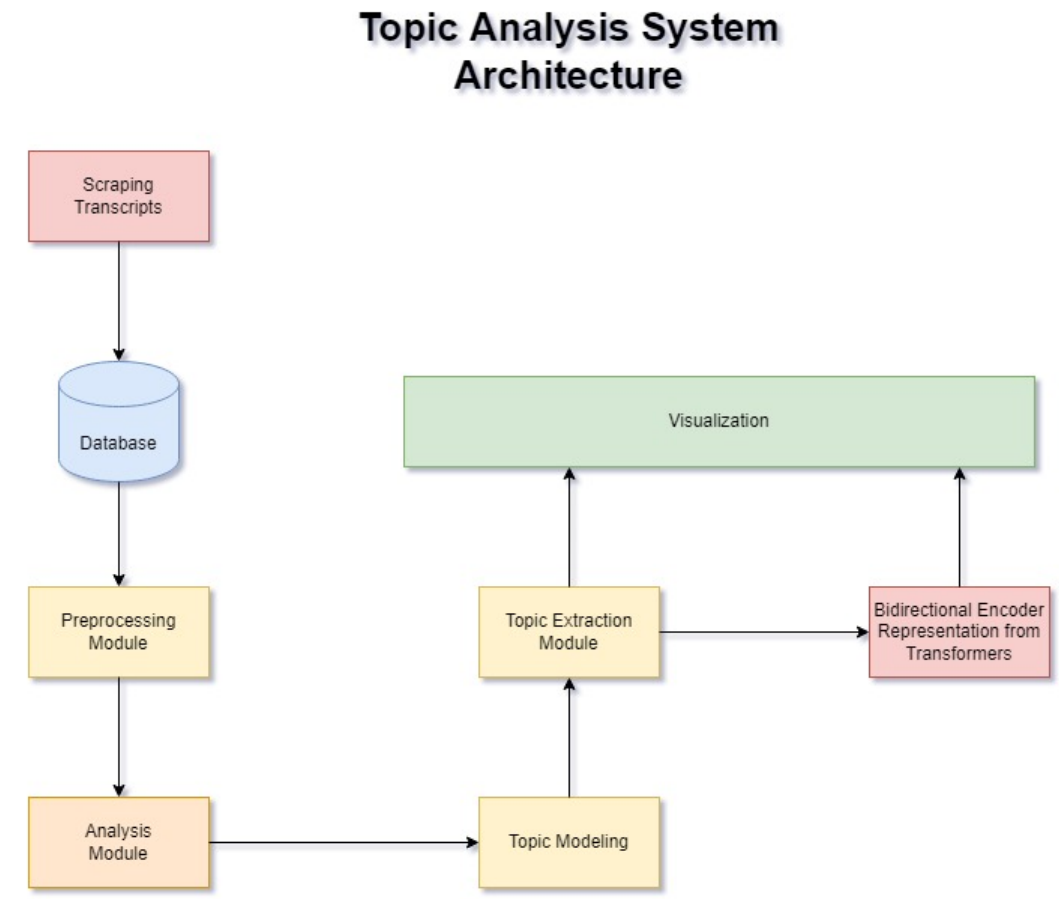


Figure 5.1: System Architecture

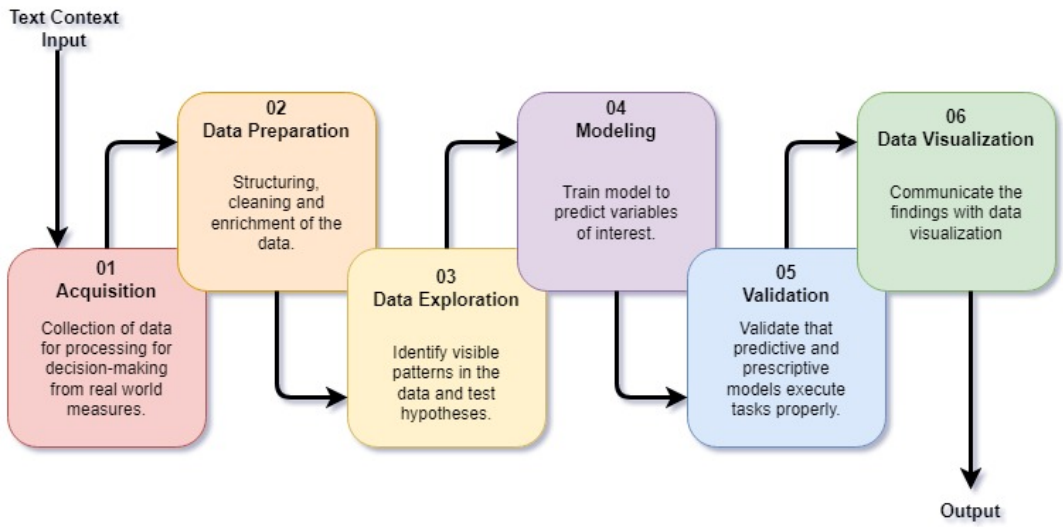


Figure 5.2: WorkFlow

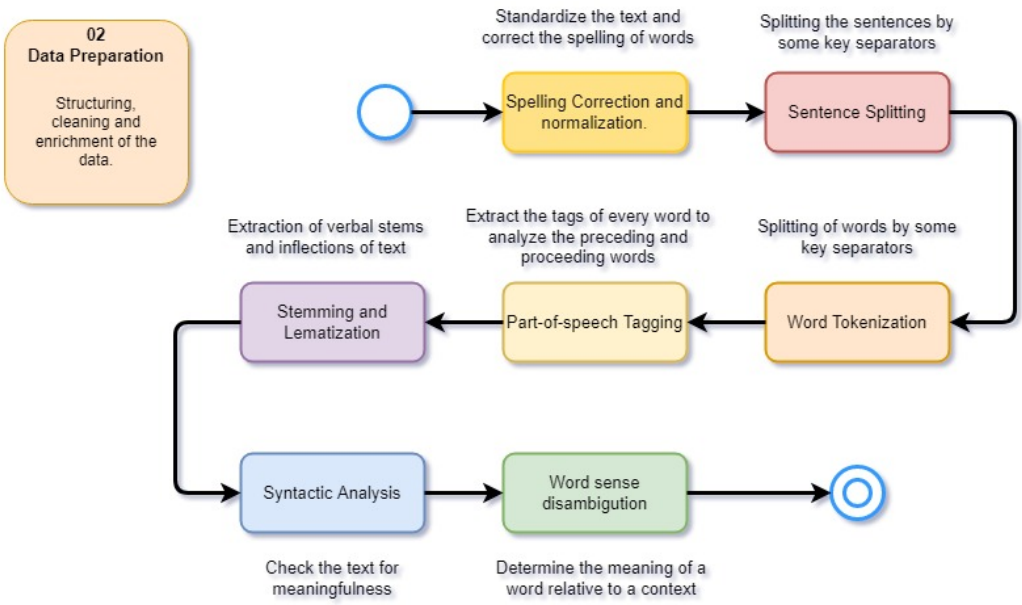


Figure 5.3: Tokenized Table Data

Chapter 6

Experimental Setup

6.1 Data Collection:

6.2 Data Preprocessing:

1) Lemmatization

2) Using Regular Expression

4)Removing Stopwards

6.3 Dominant Topics using LDA:

6.4 Merging Two or More Entities with Same Topics:

	transcript
0	good morning how
1	laughter it s great nt
2	i ve blown away whole thing in fact i m leaving laughter there three theme running conference relevant i want talk one extraordinary evidence h...
3	i find interesting if re dinner party say work education â€ actually re often dinner party frankly laughter if work education re asked lau...
4	say work education see blood run face they re like oh god know why
5	laughter my one night week laughter but ask education pin wall because s one thing go deep people i right
6	like religion money thing so i big interest education i think we huge vested interest partly s education s meant take u future ca nt grasp if t...
7	just seeing could and s exceptional i think s speak exceptional whole childhood what person extraordinary dedication found talent and contentio...
8	and girl said i m drawing picture god and teacher said but nobody know god look like and girl said they minute laughter when son four ...
9	laughter no big big story mel gibson sequel may seen laughter nativity ii but james got part joseph thrilled we considered one lead part...
10	laughter he nt speak know bit three king come
11	they come bearing gift gold frankincense myrrh this really happened we sitting i think went sequence talked little boy afterward said you ok
12	and said yeah
13	was wrong
14	they switched the three boy came fouryearolds tea towel head put box first boy said i bring gold and second boy said i bring myrrh and th...
15	they re frightened wrong i nt mean say wrong thing creative what know re prepared wrong ll never come anything original â€ re prepared wrong an...
16	i you nt think shakespeare father
17	do
18	because nt think shakespeare child
19	shakespeare seven
20	i never thought i mean seven point he somebody s english class nt
21	laughter how annoying would

Figure 6.1: Data Collection

```
In [34]: # Lemmatization
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

sentences = nltk.sent_tokenize(str1)
lemmatizer = WordNetLemmatizer()
lemm_sentence = []
# Lemmatization
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]
    sentences[i] = ' '.join(words)
#     print(words)
print(sentences)
```

```
['4.6 ( 3,348 rating ) Â |Â 99K Students Enrolled Course 1 4 Natural Language Processing Specialization This Course Video Trans
```

Figure 6.2: Web Scraping

```
In [40]: import re
import string

def clean_text_round1(text):
    '''Make text lowercase, remove text in square brackets, remove punctuation and remove words containing numbers.'''
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

round1 = lambda x: clean_text_round1(x)
```

Figure 6.3: Regular Expression

```
In [42]: # Apply a second round of cleaning
         def clean_text_round2(text):
             '''Get rid of some additional punctuation and non-sensical text that was missed the first time around.'''
             text = re.sub('[‘’“”…]', '', text)
             text = re.sub('\n', '', text)
             return text

         round2 = lambda x: clean_text_round2(x)
```

Figure 6.4: Regular Expression

```
In [46]: # We are going to create a document-term matrix using CountVectorizer, and exclude common English stop words
         from sklearn.feature_extraction.text import CountVectorizer

         cv = CountVectorizer(stop_words='english')
         data_cv = cv.fit_transform(data_clean.transcript)
         data_dtm = pd.DataFrame(data_cv.toarray(), columns=cv.get_feature_names())
         data_dtm.index = data_clean.index
         data_dtm
```

Figure 6.5: Stopword Removal

```
sent_topics_df = pd.DataFrame()
for i, row_list in enumerate(ldana[corpusna]):
        row = row_list[0] if ldana.per_word_topics else row_list
        # print(row)
        row = sorted(row, key=lambda x: (x[1]), reverse=True)
        # Get the Dominant topic, Perc Contribution and Keywords for each document
        for j, (topic_num, prop_topic) in enumerate(row):
            if j == 0:  # => dominant topic
                wp = ldana.show_topic(topic_num)
                topic_keywords = ", ".join([word for word, prop in wp])
                sent_topics_df = sent_topics_df.append(pd.Series([int(topic_num), round(prop_topic,4), topic_keywords]), ignore_
            else:
                break
sent_topics_df.columns = ['Dom_Topic', 'Topic_Contri', 'Keywords']
print(sent_topics_df)
```

	Dom_Topic	Topic_Contri	Keywords
0	4.0	0.9842	word, specialization, course, use, language, r...
1	0.0	0.9380	instructor, use, nlp, specialization, dictiona...
2	0.0	0.8853	instructor, use, nlp, specialization, dictiona...
3	4.0	0.9195	word, specialization, course, use, language, r...
4	2.0	0.9385	feature, free, check, count, look, extract, cl...
5	3.0	0.9702	tweet, sentiment, positive, word, analysis, co...
6	4.0	0.9109	word, specialization, course, use, language, r...
7	2.0	0.7333	feature, free, check, count, look, extract, cl...
8	3.0	0.9524	tweet, sentiment, positive, word, analysis, co...
9	0.0	0.9379	instructor, use, nlp, specialization, dictiona...
10	1.0	0.8664	class, word, frequency, negative, positive, nu...
11	1.0	0.8856	class, word, frequency, negative, positive, nu...
12	2.0	0.9101	feature, free, check, count, look, extract, cl...
13	3.0	0.7320	tweet, sentiment, positive, word, analysis, co...
14	1.0	0.4016	class, word, frequency, negative, positive, nu...
15	3.0	0.7331	tweet, sentiment, positive, word, analysis, co...
16	1.0	0.8397	class, word, frequency, negative, positive, nu...
17	1.0	0.5816	class, word, frequency, negative, positive, nu...
18	3.0	0.6997	tweet, sentiment, positive, word, analysis, co...
19	1.0	0.4674	class, word, frequency, negative, positive, nu...
20	1.0	0.6665	class, word, frequency, negative, positive, nu...

Figure 6.6: Output

In [21]:
```
data = {}
sentences = ""
corpus = pd.read_pickle("corpus.pkl")
corpus
len(sent_topics_df)
i=0
a=0
while(a<len(sent_topics_df)-1):
    sentences = corpus.loc[a].at['transcript']
    if(sent_topics_df.loc[a].at["Dom_Topic"] == sent_topics_df.loc[a+1].at["Dom_Topic"]):
        while((a<len(sent_topics_df)-1) and (sent_topics_df.loc[a].at["Dom_Topic"] == sent_topics_df.loc[a+1].at["Dom_Topic"])):
            sentences += corpus.loc[a+1].at['transcript']
            a+=1
    data[i] = sentences
    i+=1
    a+=1
data[i] = sentences = corpus.loc[a].at['transcript']
data
```

Out[21]:
```
{0: '4.6 ( 3,348 rating ) Â |Â 99K Students Enrolled Course 1 4 Natural Language Processing Specialization This Course Video Tr
anscript In Course 1 Natural Language Processing Specialization , : ) Perform sentiment analysis tweet using logistic regressio
n naÃ¯ve Bayes , b ) Use vector space model discover relationship word use PCA reduce dimensionality vector space visualize rel
ationship , c ) Write simple English French translation algorithm using pre-computed word embeddings locality-sensitive hashing
relate word via approximate k-nearest neighbor search .',
 1: 'By end Specialization , designed NLP application perform question-answering sentiment analysis , created tool translate la
nguage summarize text , even built chatbot !This Specialization designed taught two expert NLP , machine learning , deep learni
ng .',
 2: 'Younes Bensouda Mourri Instructor AI Stanford University also helped build Deep Learning Specialization .',
 3: 'Å\x81ukasz Kaiser Staff Research Scientist Google Brain co-author Tensorflow , Tensor2Tensor Trax library , Transformer pa
per .',
 4: 'Machine Translation , Word Embeddings , Locality-Sensitive Hashing , Sentiment Analysis , Vector Space Models 4.6 ( 3,348
rating ) HA Aug 9 , 2020 one Best course attented deeplearnig.ai last week assignment was\\n\\nto good solve cover studied enti
re course waiting course 4 nlp eagerly OA Aug 17 , 2020 Awesome .',
 5: 'The lecture exciting detailed , though little hard straight forward sometimes , Youtube helped Regression model .',
 6: 'Other , I informative fun .',
 7: 'From lesson Sentiment Analysis Logistic Regression Learn extract feature text numerical vector , build binary classifier t
weet using logistic regression !',
 8: "Instructor Instructor Senior Curriculum Developer We 'll learn generates count , use feature logistic regression classifie
r .",
 9: "Specifically , given word , want keep track number time , 's show positive class .Given another word want keep track numbe
```

Figure 6.7: Entities Merging

Chapter 7

Conclusion

This paper focuses on analysis of textual data corpus. We apply data preprocessing, topic modelling, transformation techniques to map dominant topics from the data corpus and visualise it using timeline with specific timestamp. This prototype will help in various real life applications, such as analysing main contents from educational lectures. It will help students to directly reach required content without seeing the whole video lecture.

Bibliography

[1] Mohamed Medhat Gaber-Carlos M. Dancausa Fredric Stahi Adedoyin-Olowe, Mariam and Joao Bartolo Gomes. A rule dynamics approach to event detection in twitter with its application to sports and politics. 2016.

[2] George Doddington-Jonathan Yamron Allan, Jaime G. Carbonell and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

[3] Ingrid Zukerman Andishesh Partovi, Gholamreza Haffari. Bayesian changepoint detection in textual data streams. 2015.

[4] Mohammad-Reza Feizi-Derakhshi Leili Farzinvash Mohammed-Ali Balafar Asgari-Chenaghulu, Meysam and Clina Motamed. Topic detection and tracking techniques on twitter: A systematic review. 2021.

[5] Angel Castellanos Cigarran, Juan and Ana Garicia-Serrano. A step forward for topic detection in twitter: An fca-based approach. 2016.

[6] Giuseppe Lo Re Gaglio, Salvatore and Marco Morana. Real-time detection of twitter social events from the user's perspective. 2015.

[7] Klaifer Garcia and Lilian Berton. Topic detection and sentiment analysis in twitter content related to covid-19 from brazil to usa. 2021.

[8] Yunil Wang Fragming Liao-Zachary Zanussi Samuel Larkin Goutte, Cyril and Yuri Grinberg. Evaluating change detection in online conversation. 2018.

[9] James Allan. Kumaran, Giridhar. Text classification and named entities for new event detection. 2004.

[10] Christos Emmanouilidis Cristobal Ruiz-Carcel Nomoano, Bernadin and Andrew G. Starr. Change detection in streaming data analytics: A comparison of bayesian online and martingle approaches. 2020.

[11] Sasa Petrovic Richard McCreadie-Craig Macdonald Osborne, Miles and ladhOunis. Bieber no more: First story detection using twitter and wikipedia. 2012.

[12] Symeon Papadopoulos Luca Aiello-Ryan Skarba Petkos, Georgish and Yiannis Komapatsiaris. A soft frequent pattern mining approach for textual topic detection. 2014.

[13] Yingcheng Sun and Kenneth Loapro. Topic shift detection in online discussions using structural context. 2019.

[14] Suvarna D. Tembhurnikar and Nitin N. Patil. Topic detection using bngram method and sentiment analysis on twitter dataset. 2015.

[15] Yunil Wang and Cyril Goutte. Real-time chane point detection using on-line topic models. 2018.