# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**“JNANASANGAMA” BELAGAVI - 590 018 KARNATAKA”**



**PROJECT REPORT**

ON

##### “TOPIC MODELLING USING NLP”

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF ENGINEERING IN**

**INFORMATION SCIENCE AND ENGINEERING**

Submitted by:

**1. AKASH B N**

**[1CG19IS003]**

**2. MAHESHA R**

**[1CG19IS030]**

**3. VINAY J**

**[1CG1IS058]**

**4. YASHAS T R**

**[1CG19IS059]**

**Under the guidance of: HOD:**

**Mr. Pradeep M M.Tech Dr. Thara D K Ph.D,**

Asst. Prof, Dept. of ISE, Prof & Head, Dept. of ISE,

CIT, Gubbi, Tumakuru CIT, Gubbi, Tumakuru.

**Channabasaveshwara Institute of Technology**

**(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (ISO 9001:2015 Certified Institution)**

NH 206 (B.H. Road), Gubbi, Tumkur – 572216. Karnataka

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**2022-2023**



#### D:\CIT 2\NAAC\Post SSR\Execution - Final Phase\Institute PPT\Supporting Documents\ppt images\download (2).png Channabasaveshwara Institute of Technology

**(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (ISO 9001:2015 Certified Institution)**

NH 206 (B.H. Road), Gubbi, Tumkur – 572216. Karnataka.

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**2022-2023**

# CERTIFICATE

This is to certify that the project work entitled “**TOPIC MODELLING USING NLP**” has been successfully carried out by **AKASH B N [1CG19IS003], MAHESHA R [1CG19IS030], VINAY J [1CG19IS058], YASHAS T R [1CG19IS059],** bonafide students of **CHANNABASAVESHWARA INSTITUTE OF TECHNOLOGY, GUBBI, TUMKUR,** under our supervision and guidance and submitted in partial fulfillment of the requirements for the award of Degree in **Bachelor of engineering by Visvesvaraya Technological University, Belagavi** during the academic year of 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project has been approved as it satisfies the academic requirements for the above said degree.

Guide: H.O.D:

**Mr. Pradeep M M.tech, Dr. Thara D K Ph.D,**

Asst. Prof, Dept. of ISE, Prof & Head, Dept. of ISE

CIT,Gubbi, Tumkuru. CIT,Gubbi, Tumkuru.

**Principal: Examiners:**

Dr. Suresh D S Ph.D, 1.

CIT,Gubbi,Tumkuru. 2.



#### D:\CIT 2\NAAC\Post SSR\Execution - Final Phase\Institute PPT\Supporting Documents\ppt images\download (2).pngChannabasaveshwara Institute of Technology

**(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (ISO 9001:2015 Certified Institution)**

NH 206 (B.H. Road), Gubbi, Tumkur – 572216. Karnataka.

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**2022-2023**

# UNDERTAKING

We the students **Akash B N [1CG19IS003], Mahesha R[1CG19IS030], Vinay J[1CG19IS058], Yashas T R [1CG19IS059] of VIII Semester B.E. Information Science and Engineering of CHANNABASVESHWARA INSTITUTE OF TECHNOLOGY, GUBBI, TUMAKURU** declare that project work entitled “**TOPIC MODELLING USING NLP**” has been carried out and submitted in partial fulfillment of the requirements for the award of degree in Bachelor of Engineering in Information Science and Engineering by the Visvesvaraya Technological University during this academic year 2022-2023.

1. Akash B N 3.Vinay J

[1CG19IS003] [1CG19IS058]

2. Mahesha R 4.Yashas T R

[1CG19IS030] [1CG19IS059]



#### D:\CIT 2\NAAC\Post SSR\Execution - Final Phase\Institute PPT\Supporting Documents\ppt images\download (2).pngChannabasaveshwara Institute of Technology

**(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi) (ISO 9001:2015 Certified Institution)**

NH 206 (B.H. Road), Gubbi, Tumkur – 572216. Karnataka.

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING 2022-2023**

# BONAFIEDE CERTIFICATE

This is to certify that the project work entitled “**TOPIC MODELLING USING NLP**” is a bonafied work of **Akash B N [1CG19IS003], Mahesha R [1CG19IS030], Vinay J [1CG19IS058], Yashas T R [1CG19IS003], students of VIII Semester B.E. Information Science and Engineering carried out at CHANNABASVESHWARA INSTITUTE OF TECHNOLOGY, GUBBI, TUMAKURU,** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering by the Visvesvaraya Technological University, Belagavi** under my supervision guidance.

**Signature of Guide**

**Mr. Pradeep M .,M.Tech**

Asst. Prof, Dept. of ISE

C.I.T, Gubbi, Tumkuru.

**ACKNOWLEDGEMENT**

A great of time and lot of effort has gone into completing this project report and documenting it. The number of hours spent in getting through various books and other materials related to this topic chosen by us have reaffirmed its power and utility in doing this project

Several special people have contributed significantly to this effort. First of all, I am grateful to my institution, **Channabasaveshwara Institute of Technology, Gubbi**, which provides me an opportunity in fulfilling my most cherished desire of reaching my goal.

We acknowledge and express my sincere thanks to our beloved Director & Principal, **Dr. Suresh D S**, for his many valuable suggestion and continued encouragement by supporting me in my academic endeavors.

We wish express our deep sense of gratitude to **Dr. Thara D K,** head of the department of Information science and Engineering for all the guidance and who still remains a constant driving force and motivated through innovative ideas with tireless support and advice during the project to examine and helpful suggestions offered.

We wish to express our deep sense of gratitude to our guide **Mr. Pradeep M ,** Asst.Prof , dept of ISE, CIT, for his meticulous attention to details, which has contributed immeasurably to the quality of the project report.

We would express our gratitude towards our parents and friends for their kind cooperation and encouragement which helped us in completion of this project.

Finally, we would like to thank all the teaching and non-teaching staff of dept of ISE, for their cooperation.

Thanking everyone…

**AKASH B N [1CG19IS003]**

**MAHESHA R [1CG19IS030]**

**VINAY J [1CG19IS058]**

**YASHAS T R [1CG19IS059]**

## ABSTRACT

Every day large quantities of data are collected. As more information is available, the access to what we are seeking gets challenging. We, therefore, require processes and techniques for organizing, searching, and understanding massive amounts of information. The task of topic modeling is to analyze the whole document to learn the meaningful pattern that exists in the document. It is a supervised strategy used to identify and monitor words in clusters of texts (known as the "topics"). Through the use of topic analysis models, companies can load tasks on machines rather than burden employees with too much data. In this paper, we have used Word embedding for Topic Modelling to learn the meaningful pattern of words, and k-means clustering is used to group the words that belong to one group. In this paper, we have created the nine clusters of words from the headline dataset. One of the applications of topic modeling i.e sentiment analysis using the VADER algorithm is also demonstrated in this paper.

Keywords—Topic Modeling, Word Embedding, K-means, Sentiment Analysis,VADER.

## TABLE OF CONTENTS

**SL NO. CHAPTERS PAGE NO.**

1. INTRODUCTION 1-2

1.1 [Objectives 2](#_TOC_250015)

1.2 [Problem statement 3](#_TOC_250014)

1.3 Scope of the project 3

1. LITERATURE SURVEY 4-6
2. SYSTEM ANALYSIS 7-8
   1. 3.1 Existing System 7
   2. 3.2 Proposed system 7-8
3. SYSTEM DESIGN 9-12

4.1 Flow chart 9

4.2 system architecture 10

4.3 sequence diagram 10

4.4 graphical model representation diagram 11-12

1. IMPLEMENTATION 13-22

5.1 Data collection 14

5.2 Exploratory data analytics 14

5.3 Data preprocessing 15

5.3.1 Scaling 16

5.3.2 Handling outliers 16

5.4 Model Building and training 17

5.4.1 Data pre-processing 18

5.4.2 Feature Extraction 18

5.4.3 Sentiment classification 19

5.5 Experiments and evaluations 20

5.1.1 Dataset 20

5.1.2 Data analysis 20

5.1.3 Data clearing 21

5.1.4 Clustering using word tovec embeddings 21-22

6. METHODOLOGY 23-25

7. RESULTS 26-34

CONCLUSION 35

[REFERENCES 36](#_TOC_250000)

## LIST OF FIGURE

**FIG NO. PAGE NO.**

4.1 Flow chart 09 4.2 System Architecture 10

4.3 Sequence diagram 11

4.4 graphical model representation diagram 11

5.0 Number of tweets by sentiment 18

5.1 Positive tweets 19

5.2 Negative tweets 19

5.3 Number of words in the headlines 20

5.4 Top 15 words in the headlines 21

5.5 Topic1 of cluster 21

6.0 Topic2 of cluster 22

6.1 Proposed Methodology 23

**CHAPTER 1**

##### INTRODUCTION

Natural language processing (NLP) is a challenging research in computer science to information management, semantic mining, and enabling computers to obtain mean- ing from human language processing in text-documents. Topic modeling methods are

* powerful smart techniques that widely applied in natural language processing to topic discovery and semantic mining from unordered documents [1]. In a wide perspective, Topic modeling methods based on LDA have been applied to natural language pro- cessing, text mining, and social media analysis, information retrieval. For example, topic modeling based on social media analytics facilitates understanding the reactions and conversations between people in online communities, As well as extracting use- ful patterns and understandable from their interactions in addition to what they share on social media websites such as twiiter facebook [2-3]. Topic models are prominent for demonstrating discrete data; also, give a productive approach to find hidden struc- tures(semantics) in gigantic information. There are many papers for in this field and definitely cannot mention to all of them, so we selected more signification papers. Topic models are applied in various fields including medical sciences [4-7] , software engineering [8-12], geography [13-17], political science [18-20] , etc.
* 2 For example in political science, In [20] proposed a new two-layer matrix factor- ization methodology for identifying topics in large political speech corpora over time and identify both niche topics related to events at a particular point in time and broad, long-running topics. This paper has focused on European Parliament speeches, the proposed topic modeling method has a number of potential applications in the study of politics, including the analysis of speeches in other parliaments, political man- ifestos, and other more traditional forms of political texts. In [21] suggested a new unsupervised topic model based on LDA for contrastive opinion modeling which pur- pose to find the opinions from multiple views, according to a given topic and their difference on the topic with qualifying criteria, the model called Cross-Perspective Topic (CPT) model. They performed experiments with both qualitative and quan- titative measures on two datasets in the political area that include: first dataset is statement records of U.S. senators that show political stances of senators. Also for the second dataset, extracted of world News Medias from three representative media in U.S (New York Times), China (Xinhua News) and India (Hindu). To evaluate their approach with other models, used corrIDA and LDA as two baselines.
* Another group of researchers focused on topic modeling in software engineering, in [8] for the first time, they used LDA, to extract topics in source code and perform visualization of software similarity, In other words, LDA is used as an intuitive ap- proach for calculation of similarity between source files and obtain their respective distributions of each document over topics. They utilized their method on 1,555 soft- ware projects from Apache and SourceForge that includes 19 million source lines of code (SLOC). The authors demonstrated this approach, can be effective for project or- ganization, software refactoring.

* In [22] introduced a method based on LDA for auto- matically categorizing software systems, called LACT. For evaluation of LACT, used 43 open-source software systems in different programming languages and showed LACT can categorize software systems based on type of programming language. In [23, 24] proposed an approach topic modeling based on LDA model for the pur- pose of bug localization. Their idea, applied to analysis of same bugs in Mozilla and Eclipse and result showed that their LDA-based approach is better than LSI for eval- uate and analyze bugs in these source codes.
* An analysis of geographic information is another issue that can be referred to [17]. They introduced a novel method based on multi-modal Bayesian models to describe social media by merging text features and spatial knowledge that called GeoFolk. As a general outlook, this method can be considered as an extension of Latent Dirichlet Allocation (LDA). They used the available standard CoPhIR dataset that contains an abundance of over 54 million Flickr. The GeoFolk model has the ability to be used in quality-oriented applications and can be merged with some models from Web 2.0 social. In [16], this article examines the issue of topic modeling to extract the top- ics from geographic information and GPS-related documents. They suggested a new location text method that is a combination of topic modeling and geographical clus- tering called LGTA (Latent Geographical Topic Analysis). To test their approaches, they collected a set of data from the website Flickr, according to various topics.

###### 1.1 OBJECTIVES

* + Group similar documents together based on their topic distributions.
  + Analyze customer feedback and identify the topics that are most important to customers.
  + Facilitate text classification and clustering tasks.
  + Identify trends and patterns over time using dynamic LDA.
  + Create more accurate and effective natural language processing tools and models.

**1.2 Problem statement**

1. Traditional methods of analyzing text data, such as manual tagging or keyword-based approaches, are time-consuming

**1.3 Scope of the project**

As an AI language model, I cannot know the specific scope of your project without additional information. Can you provide more details about your project, such as its goals, objectives, and requirements? With this information, I may be able to provide more targeted assistance.

### CHAPTER 2

### LITERATURE SURVEY

### 2.1 Literature Review

1. In paper[1] Word Embedding for topic modeling. They compared and assessed two distinct frameworks for unsupervised topic modeling of the CompWHoB Corpus, essentially the political-linguistic dataset, in this study. The first approach leverages the Latent Dirichlet Allocation technique, while the second framework uses the Word2Vec methodology to learn word vector representations that will later be utilized for topic modeling. The linguistic preprocessing stage was given specific attention in order to increase the quality of textual data. NLTK library is used for word tokenization, POS-tagging, and To refine the data, lemmatization was used. In the first experiment, LDA is utilized, which is a generative probabilistic model for inferring latent topics from a collection of documents. The LDA is trained on training corpus by the use of the Gensim library following the pre-processing stage Word2Vec is utilized in the second experiment. Based on the hypothesis that words that appear in similar settings have similar meanings. The Word2Vec model can be used to learn word embeddings, which are vector representations of words. The CBOW technique was used to train the model because it is better suitable for larger datasets.
2. In paper [2] Topic Modeling in Embedding Spaces, An embedded topic model is employed for the generative document model which combines conventional topic with word embedding. Even with huge vocabularies containing unusual words and stop words, the ETM finds interpretable topics. The ETM probably employs an embedding word matrix, a representation of the vocabulary in a smaller space. They can either employ pre-fitted embeddings or learn them in practice as part of their entire strategy. When the ETM learns embedding as part of the fitting technique, it simultaneously discovers topics and integration space. Like LDA, the ETM is a generative model of probability, in which every document is a mix of subjects, and a particular topic is given to each word observed. Each sentence is represented by an integration; each subject is a point within the space of the integration, and the distribution of the subject over terms is proportionate to the internal product of the embedding of the subject and the embedding of every term. One of the objectives in the ETM was to integrate word similarity into the topic model. The ETM model is the combination of LDA and word embeddings. They have employed the variation of word integration of a continuous bag of words (CBOW). The idea underlying consistency is that a well-organized theme emphasizes terms that often appear in similar writing. In other words, a high level of mutual information should most likely be found in a coherent topic. Topic models with greater cohesion are more understandable. The ETM learns a corpus in a specific embedding space when used previously fitted embeddings. This is particularly useful when the insertion comprises terms not found in the corpus. The ETM can determine the size of the words in the subjects.ETM gives better predictions and topics than LDA and the Neural Document Model. Both consistent language patterns and the accurate distribution of words should be provided by a good document model, which will require both predictability and topical interpretability to measure performance.
3. In paper[3] Semantic Augmented Topic Model over Short Text, The shorter text was proposed for a latent semantic augmented bi-term topic model(LS-BTM). The popularities of mobile equipment make short texts an important element of the information carrier. For many natural language tasks like detecting emerging topics, content analysis, question answering, sentimental analysis, automatic summary systems, recommendation systems, etc. discovering the prospective topics from the short text are significant. However, the lack of short texts results in insufficient information within the context, and it is impossible to analyze the diversity of language expression by general approaches. The classical subject models like LDA and PLSA were quite successful in a long text. However, because of the lack of word patterns, they operate badly on short texts. Topic models were explored for numerous years and utilized successfully in numerous areas. In short text topic models, however, there are significant challenges. In brief texts, for instance, a tweet message comprises a maximum of 140 characters, a maximum of more than 90% is less than 10 words and nearly half of the text-only includes one or two words. Secondly, short messages are always flexibly expressed in a language that leads to ambiguity.
4. Topic models have many applications in natural processing languages. Many articles have been published based on topic modeling approaches in various subject such as Social Network, software engineering, Linguistic science and etc. There are some works that have focused on survey in Topic modeling. In [52], the authors presented a survey on topic modeling in software engineering field to specify how topic models have thus far been applied to one or more software repositories. They focused on articles written between Dec 1999 to Dec 2014 and surveyed 167 article that using topic modeling in software engineering area. They identified and demonstrate the research trends in mining unstructured repositories by topic models. They found that most of studies focused on only a limited number of software engineering task and also most studies use only basic topic models. In [53], the authors focused on survey in Topic Models with soft clustering abilities in text corpora and investigated basic concepts and existing models classification in various categories with parameter estimation (such as Gibbs Sampling) and performance evaluation measures. In addition, the authors presented some applications of topic models for modeling text corpora and discussed several open issues and future directions.
5. In [54], introduced and surveyed the field of opinion mining and sentiment analysis, which helps us to observe a elements from the intimidating unstructured text. The authors discussed the most extensively studied subject of subjectivity classification and sentiment which specifies whether a document is opinionated. Also, they described aspect-based sentiment analysis which exploits the full power of the abstract model and discussed about aspect extraction based on topic modeling approaches. In [55], they discussed challenges of text mining techniques in information systems research and indigested the practical application of topic modeling in combination with explanatory regression analysis, using online customer reviews as an exemplary data source.
6. In [56], the authors presented a survey of how topic models have thus far been applied in Software Engineering tasks from four SE journals and eleven conference proceedings. They considered 38 selected publications from 2003 to 2015 and found that topic models are widely used in various SE tasks in an increasing tendency, such as social software engineering, developer recommendation and etc. Our research difference with other works is that, we had a deep study on topic modeling approaches based on LDA with the coverage of various aspects such as applications, tools , dataset and models.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Without additional information about the type of system you are referring to, it is difficult for me to provide a specific response. However, in general, an existing system refers to a software or hardware system that is already in place and functioning. This system may have been developed by a team or an individual and may be used for various purposes, such as data processing, automation, communication, or management.

When working on a project, it is important to understand the existing system and its limitations, strengths, and weaknesses. This can help inform decisions about whether to build upon the existing system or create a new one, and what changes or improvements need to be made to achieve the desired goals and outcomes. Additionally, understanding the existing system can help identify potential issues or challenges that may need to be addressed during the project development process.

# 3.2 PROPOSED SYSTEM

A proposed system is a new software or hardware system that is being developed to replace or improve upon an existing system, or to address a new need or opportunity. The proposed system typically involves the development of new technology, tools, or processes, and may require significant planning, design, and implementation effort.

When developing a proposed system, it is important to clearly define the project goals and objectives, as well as the requirements and specifications for the system. This can help ensure that the proposed system meets the needs of its intended users and stakeholders. Additionally, it is important to consider factors such as budget, timeline, and resource availability when planning and developing the proposed system.

The development process for a proposed system typically involves several stages, including requirements gathering, design and prototyping, testing and evaluation, and implementation and deployment. It may also involve collaboration between different teams or individuals, such as developers, designers, project managers, and stakeholders.

**Our approach to building model has following steps**

1. **Problem Definition:** Define the problem you want to solve, and identify the goals and objectives of the project.
2. **Data Collection:** Gather relevant data from various sources that can be used to train the model.
3. **Data Preprocessing:** Clean, transform, and preprocess the data to make it suitable for model training.
4. **Feature Engineering:** Select and create relevant features that can be used to train the model.
5. **Model Selection:** Choose a suitable machine learning or deep learning model that can be used to train the data.
6. **Model Training:** Train the selected model on the preprocessed and engineered data.
7. **Model Evaluation:** Evaluate the performance of the trained model using appropriate evaluation metrics and techniques.
8. **Model Optimization:** Optimize the model by tweaking the parameters, adjusting the hyperparameters, or using advanced techniques such as ensembling or transfer learning.
9. **Model Deployment:** Deploy the optimized model in a production environment or integrate it with other software systems.
10. **Model Monitoring:** Continuously monitor the performance of the deployed model and retrain or update it as needed.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Flowchart:**

A flow chart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step by step approach to solving a task. The flow chart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

Tokenistion and stemming

Load Data

TF-IDF

K-means clusturing

Topic Modeling Latent Dirichlet Allocation

Discussion

**Fig 4.1 Flowchart**

**4.2 system architecture:**

Topic 0

Topic modeling method

Data pre processing

Input

Topic 1

Topic 2

Topic -k

**Fig: 4.2 System architecture**

**4.3 Sequence diagram**

Sequence diagrams are a type of UML (Unified Modeling Language) diagram that represents the interactions between objects in a system or software application. They are used to visualize the flow of events or messages between objects in a software system, and are particularly useful for understanding complex systems and how they work.

Sequence diagrams are made up of several elements, including actors (people or systems that interact with the system), objects (the elements in the system that perform specific functions), messages (the information or instructions that are passed between objects), and lifelines (the vertical lines that represent the lifetime of an object).

To create a sequence diagram, you typically start by identifying the actors and objects in the system, and then determine the sequence of events or messages that occur between them. You can then use the UML notation to represent these elements and their interactions in a visual diagram.

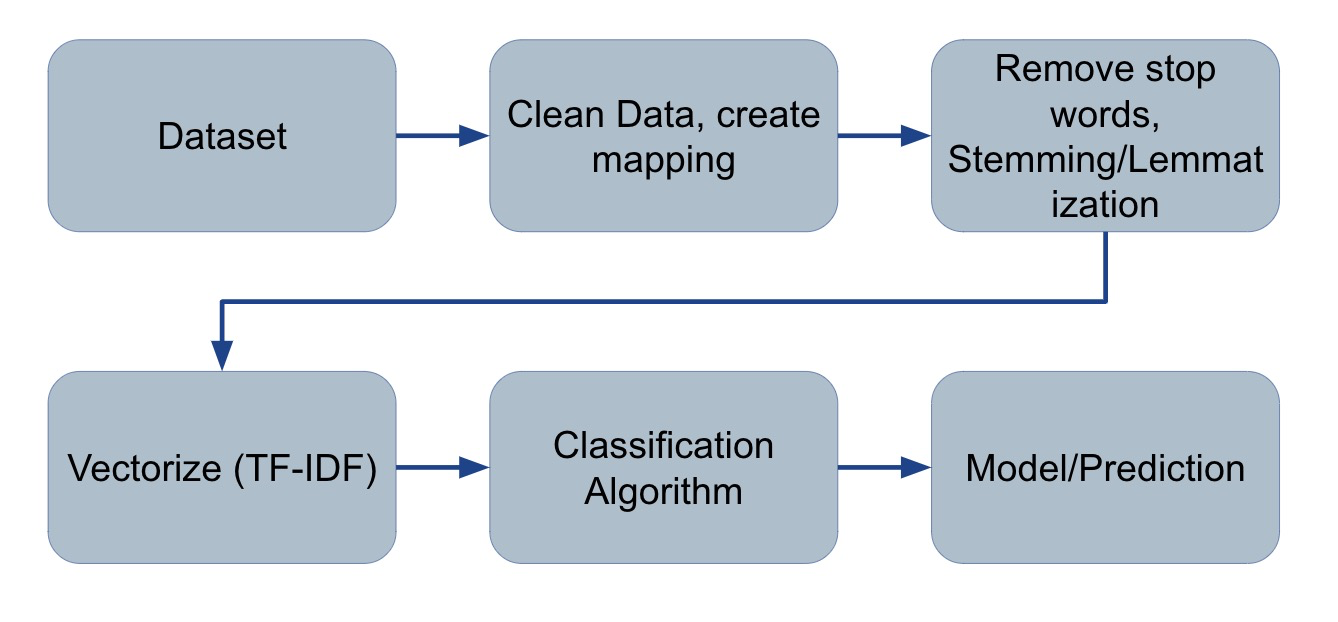
Code: Actor -> Object: Message

Object -> Object: Message

Object -> Actor: Message

In this example, the first line represents an actor sending a message to an object. The second line represents an object sending a message to another object. The third line represents an object sending a message back to the actor.

Sequence diagrams can be a powerful tool for designing, testing, and documenting complex software systems, and are commonly used in software development and engineering projects.



**Fig 4.3: Sequences diagram**

**4.4 graphical model representation diagram**

**β**

K

Q W N

M

**Fig: 4.3: graphical model representation diagram**

A graphical model representation diagram is a type of diagram used in machine learning and artificial intelligence to represent a statistical model or system. These diagrams can help to visualize the structure of a model, the relationships between different variables or components, and the flow of information or data through the system.

There are several types of graphical model representation diagrams, including:

1. **Bayesian networks:** A Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies using a directed acyclic graph (DAG). The nodes in the graph represent random variables, and the edges represent the conditional dependencies between them.
2. **Markov models:** A Markov model is a stochastic model that represents a system that transitions between different states over time. These models are represented using a directed graph, where the nodes represent the states and the edges represent the probability of transitioning between states.
3. **Decision trees:** A decision tree is a type of diagram that represents a set of decisions and their possible consequences. The tree structure represents the sequence of decisions and their outcomes, and can be used to make predictions or classifications based on input data.
4. **Neural networks:** A neural network is a type of model that is inspired by the structure of the human brain. It is composed of layers of interconnected nodes (neurons) that process information and make predictions based on input data.
5. **Flowcharts:** A flowchart is a type of diagram that represents the flow of information or data through a system. It is composed of a set of symbols and arrows that represent different actions, decisions, and data flows in the system.

**CHAPTER 5**

### IMPLEMENTATION

**5.1 Data collection**

### Data collection is the process of gathering information or data from various sources to support research or analysis. It is an important step in the research process and can involve a variety of methods, including surveys, interviews, experiments, observations, and secondary data sources.

### The following are some common methods used for data collection:

### Surveys: Surveys are one of the most common methods for collecting data. They involve asking a series of questions to a group of individuals or organizations, either in person, over the phone, or online.

### Interviews: Interviews are another method for collecting data. They involve one-on-one conversations with individuals or groups to gather information and insights on a specific topic.

### Experiments: Experiments involve manipulating one or more variables to observe the effects on a particular outcome. They are often used in scientific research to test hypotheses and identify causal relationships.

### Observations: Observations involve watching and recording the behavior of individuals or groups in a natural or controlled setting. They can provide valuable insights into behavior and decision-making processes.

### Secondary data sources: Secondary data sources involve gathering data from existing sources, such as government reports, academic journals, or company financial statements. This can be a cost-effective and time-efficient way to gather data, but the quality and relevance of the data can vary.

### 

**5.2 Exploratory Data Analytics**

Exploratory data analysis (EDA) is the process of analyzing and summarizing data to gain insights and understanding about the underlying patterns, relationships, and trends in the data. It is an important step in any data analysis project, as it helps to identify potential issues with the data, explore the distribution of variables, and generate hypotheses for further investigation.

The process of EDA typically involves several steps, including:

1. **Data Cleaning:** This involves identifying and addressing missing values, outliers, and other anomalies in the data.
2. **Univariate Analysis:** This involves analyzing individual variables in the data, such as their distribution, central tendency, and variability.
3. **Bivariate Analysis:** This involves exploring the relationship between two variables in the data, such as their correlation, association, or causality.
4. **Multivariate Analysis:** This involves examining the relationships among multiple variables in the data, using techniques such as factor analysis or cluster analysis.
5. **Visualization:** This involves creating graphical representations of the data, such as histograms, scatter plots, or heatmaps, to better understand the patterns and relationships in the data.
6. Overall, EDA is an important tool for data scientists and analysts, as it can help to generate insights and guide further analysis, modeling, and decision-making.

**5.3 Data preprocessing**

Data preprocessing is a crucial step in data analysis, where raw data is cleaned, transformed, and prepared for further analysis. It involves various techniques to transform data into a more manageable and usable format.

The following are some common techniques used in data preprocessing:

1. **Data cleaning:** This involves identifying and handling missing values, outliers, duplicates, and inconsistent data.
2. **Data transformation:** This involves transforming data into a more suitable format for analysis. This can include converting categorical variables into numeric variables or scaling data to have a similar range.
3. **Data integration:** This involves combining data from multiple sources into a single dataset.
4. **Data reduction:** This involves reducing the number of variables or observations in a dataset, while maintaining the integrity of the data.
5. **Data normalization:** This involves scaling data to a common range to make it easier to compare and analyze.
6. **Feature selection:** This involves selecting a subset of relevant features or variables to use in analysis, to reduce the dimensionality of the dataset and improve accuracy.

**5.3.1 Scaling**

Scaling is a technique used in data preprocessing to transform data so that it has a similar range, distribution, and variance. This is done to improve the performance and accuracy of certain machine learning algorithms, such as those that use distance-based metrics, and to make it easier to compare and analyze different variables.

There are several common scaling techniques:

1. **Min-max scaling:** This involves scaling data so that it falls within a specified range, typically between 0 and 1. This is done by subtracting the minimum value and dividing by the range.
2. **Z-score normalization:** This involves scaling data so that it has a mean of 0 and a standard deviation of 1. This is done by subtracting the mean and dividing by the standard deviation.
3. **Robust scaling:** This involves scaling data using the interquartile range (IQR) instead of the mean and standard deviation. This can be more effective when there are outliers present in the data.
4. **Log transformation:** This involves transforming data by taking the logarithm of the values. This can help to reduce the influence of extreme values and make the distribution of the data more normal.

# 5.3.2 Handling outliers

# Outliers are data points that are significantly different from the other observations in a dataset. They can occur due to various reasons such as measurement errors, data corruption, or rare events. Outliers can affect the accuracy and reliability of data analysis, so it is important to handle them appropriately. Here are some common techniques to handle outliers:

# Identify the outliers: The first step is to identify the outliers in the dataset using various statistical techniques, such as box plots, scatter plots, or z-scores.

# Remove the outliers: One approach is to remove the outliers from the dataset, especially if they are due to measurement errors or data corruption. However, it is important to carefully consider the impact of removing outliers, as it can affect the overall distribution and skewness of the data.

# Transform the data: Another approach is to transform the data using techniques such as log transformation or square root transformation, which can reduce the impact of outliers and make the distribution more normal.

# Treat the outliers: Another approach is to treat the outliers by replacing them with a more appropriate value. This can be done by using the mean, median, or interpolation techniques to estimate a more reasonable value for the outlier.

# Use robust statistical techniques: Robust statistical techniques, such as median or trimmed mean, are less sensitive to outliers than other techniques and can be used to handle outliers in the data.

**5.4 Model Building and Training**

Model building and training is the process of developing a machine learning model and training it on a dataset to make accurate predictions or classifications. Here are some common steps involved in model building and training:

1. **Data preprocessing:** The first step is to preprocess the data, which involves cleaning, transforming, and preparing the data for modeling. This can include handling missing values, scaling the data, and encoding categorical variables.
2. **Model selection:** The next step is to select an appropriate machine learning algorithm based on the nature of the data and the problem at hand. This can include supervised learning algorithms such as regression or classification, or unsupervised learning algorithms such as clustering or dimensionality reduction.
3. **Model architecture:** Once the algorithm is chosen, the model architecture needs to be defined. This involves selecting the appropriate features or variables to use in the model, and deciding on the number and type of layers or nodes in the model.
4. **Model training:** The next step is to train the model on a dataset. This involves dividing the data into training and validation sets, and using the training set to optimize the model parameters using an appropriate optimization algorithm such as gradient descent.
5. **Model evaluation:** Once the model is trained, it needs to be evaluated on a separate test set to assess its performance. This can be done using various performance metrics such as accuracy, precision, recall, or F1-score.
6. **Model tuning:** If the model performance is not satisfactory, it may be necessary to tune the model parameters or adjust the model architecture to improve its performance.
7. **Deployment:** Once the model is trained and tested, it can be deployed for real-world use, either by integrating it into an application or using it to make predictions on new data.

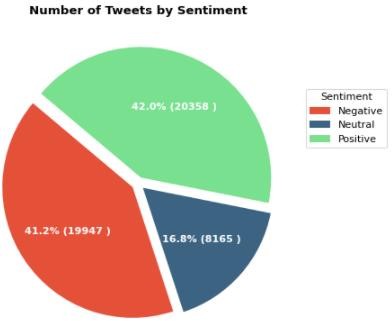
**5.4.1 Data Pre-processing:**

In order to transform data in a useable format for future processing, data processing is performed. In this we first remove the duplicate tweet if any exist, then we delete the extra spaces from the text data. After this, we remove the hashtag and ‘@’ from the tweet data. Then delete all the links and the text whose length is less than 3 from the tweet as it doesn’t carry any information. At the end we remove all the stop words, punctuations then the tweet is converted into tokens.

**5.4.2 Feature Extraction:**

In this step we extract the all the useful feature that is needed for next step.

Applying VADER Sentiment Analyzer and Sentiment Score Calculation:

In this, we import the pre-trained VADER algorithm of Sentiment Analyzer. The sentiment value is derived on the basis of every word's intensity.

**Fig. 5.0: Number of Tweets by sentiment**

**5.4.3 Sentiment Classification:**

Once score is calculated the sentiment is classify as positive,negative,neutral.

**Fig. 5.1: positive tweets**

# 

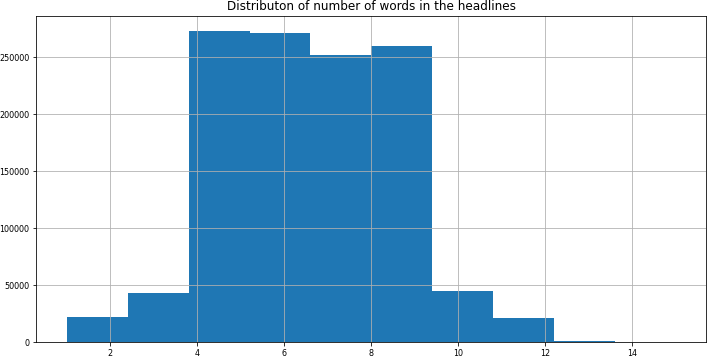
**Fig.5.2: Negative tweets**

**5.5 EXPERIMENTS AND EVALUATIONS**

**5.1.1. Dataset:**

We'll extract topics from a million news headlines collected from the respectable Australian news outlet ABC(Australian Broadcasting Corp.) in this exercise. The dataset can be found on Kaggle. Dataset content three column: publish\_date , headline\_text and Start Date.

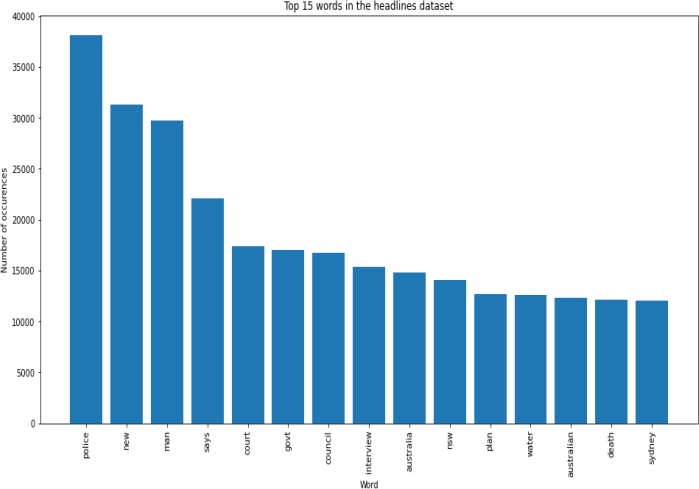
**5.1.2 Data Analysis:**

We'll start explora tory data analysis now that we've imported our data to get a better idea of what's in it.

**Fig. 5.3: N umber of words in the headline**

**5.1.3. Data Cleaning:**

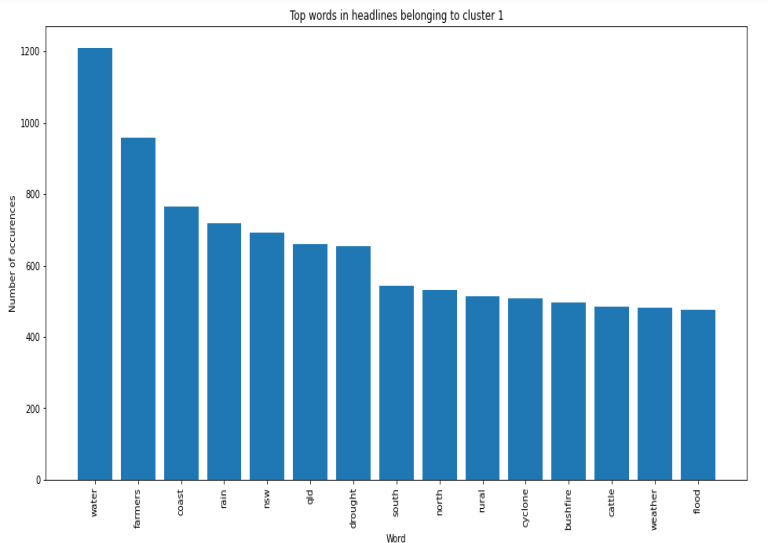
Let's continue to clean up the headlines by converting each word to lowercase, deleting punctuation, and deleting non-ASCII characters that aren't important to modeling topics.

****

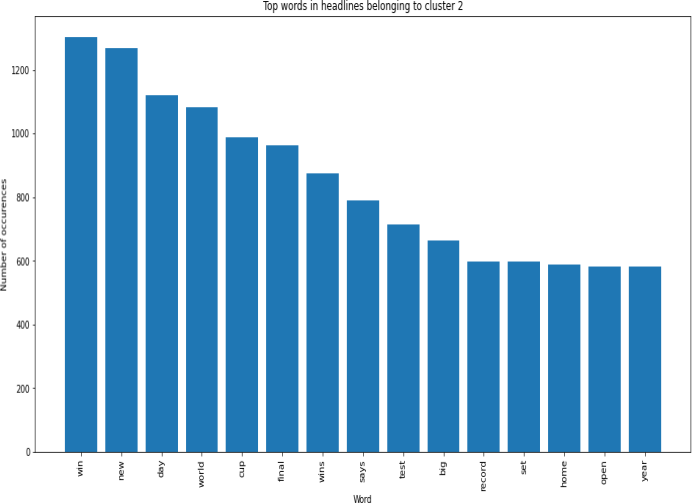
**Fig. 5.4 :Top 15 words in the headlines**

**5.1.4. Clustering using ‘wordtovec’ embeddings:**

We will import the word embeddings from the pre-trained deep NN on google news and then represent each headline with the mean of word embeddings for each word in that headline. Now, we will randomly sample 20% of the data because of the memory constraints and then build the clustering model using the word embeddings we just imported. Now we have 22343 headlines and each headline has 300 features. Let us use KMeans CLustering to cluster them into 8 clusters.



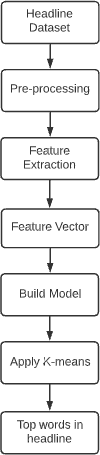
**Fig. 5.5: Topic1 of Cluster**

****

**Fig. 6.0: Topic 2 of Cluster**

# CHAPTER 6

**METHODOLOGY**



**Fig. 6.1: Proposed Methodology**

**Word Embedding :** Word Embeddings are numerical representations of texts that have been translated. The same text may be represented numerically in multiple ways. Individual words in word embeddings are represented as real-valued vectors in a vector space. It can recognize a word's context in a document, as well as semantic and grammatical similarities, relationships with other words, and so on. It is a method of providing a dense representation of words. Due to its training speed and performance, Word2Vec is one of the most common strategies for learning word embeddings. The word2vec tool generates vector representations of words by employing the continuous bag-of-words (CBOW) and skip-gram models. These representations can then be employed in a variety of natural language processing applications.

**How does word2vec work: Word** vectors are generated from a text corpus using the word2vec tool. It learns the vector representation of words after creating a vocabulary from the training text input. The generated word vector file has the advantage of being usable in a wide range of (NLP)natural language processing and (ML)machine learning applications.

Finding the closest terms to a user-specified term is a simple method for investigating the learned representations. This is what the distance tool is for. If we type in 'Iraq,' for example, the distances between them and 'Iraq' will be displayed. It has recently been demonstrated that word vectors capture numerous linguistic regularities, such as vector operations.

**Continuous Bag of words:** The way CBOW works is that it predicts the probability of a word given a context. A single word or several words could be a context.

An input layer, a hidden layer, and an output layer comprise a three-layer neural network.

The flow is as follows:

1. The target layer and the input layer, both single-hot encoded of size [1 X V].
2. Two weights are available. One is between the input and the layer that has been hidden, and the second between the layer that has been hidden and the output.
3. The size of the input-hidden layer matrix is [V X N], and the size of the hidden-output layer matrix is [N X V].

Where N denotes the number of dimensions in which we want our word to be represented. It is an arbitrary hyper-parameter for a Neural Network. In addition, N represents the number of neurons in the hidden layer.

4. The input is multiplied by the input-hidden weights and called hidden activation.

5 . The output is calculated after multiplying the hidden input by the hidden-output weights.

6. It determines and re-adjusts the differences between the output and the target error.

7. As a vector representation of the word, the weight between the hidden layer and the output layer will be assumed.

**Skip – Gram model:** It simply flips the architecture of CBOW on its head. Skip- gram is a technique for predicting the context of a word. Skip- gram input vectors are comparable to a 1-context CBOW. The differences will be found in the target variable. Two single-hot coded target variables are available and two corresponding outputs because both sides have a context window of one. For each target variables, two independent errors are created, resulting in an element-by-element error vector that is propagated back to updating weights. Element by element. After training, the input and hidden layers weights are used to create a word vector representation. The objective or loss function is the same as in the CBOW model.

**K means Clustering Algorithm:** Clustering is one of the most often exploratory data analysis approaches for gaining a sense of the data's structure. The k- means algorithm identifies clusters in the dataset so that data points in the same cluster are quite identical while data points in other clusters are quite dissimilar.

K-means Algorithm segment the dataset into k pre-defined clusters(subgroups) with every data point belonging to a single category.

K-.means algorithm is given below:

1. the first step is to select k no. of the cluster.
2. As centroids, pick k random no. of points from the given data.
3. Allocate all points to the closest cluster centroid.
4. Determine again the centroids of the newly generated clusters.
5. Repeat steps 3 and 4.

To stop the K-means algorithm, three conditions can be used: 1.Newly created cluster centers do not alter.

1. The points stay in the same group. 3.Iterations shall reach the maximum number.

# CHAPTER 7

# RESULTS

# C:\Users\Admin\Downloads\Screenshot (80).png

# C:\Users\Admin\Downloads\Screenshot (82).png

# C:\Users\Admin\Downloads\Screenshot (83).png

# C:\Users\Admin\Downloads\Screenshot (84).png

# C:\Users\Admin\Downloads\Screenshot (85).png

# C:\Users\Admin\Downloads\Screenshot (86).png

# C:\Users\Admin\Downloads\Screenshot (87).png

# C:\Users\Admin\Downloads\Screenshot (89).png

# C:\Users\Admin\Downloads\Screenshot (90).png

# C:\Users\Admin\Downloads\Screenshot (91).png

# C:\Users\Admin\Downloads\Screenshot (92).png

# C:\Users\Admin\Downloads\Screenshot (93).png

# Conclusion

Topic models have an important role in computer science for text mining. In Topic modeling, a topic is a list of words that occur in statistically significant methods. A text can be an email, a book chapter, a blog posts, a journal article and any kind of unstructured text. Topic models cannot understand the means and concepts of words in text documents for topic modeling. Instead, they suppose that any part of the text is combined by selecting words from probable baskets of words where each basket corresponds to a topic. The tool goes via this process over and over again until it stays on the most probable distribution of words into baskets which call topics. Topic modeling can provide a useful view of a large collection in terms of the collection as a whole, the individual documents, and the relationships between the documents. In this paper, we investigated scholarly articles highly (between 2003 to 2016) related to Topic Modeling based on LDA in various science. Given the importance of research, we believe this paper can be a significant source and good opportunities for text mining with topic modeling based on LDA for researchers and future works.

# References

1. Blei, D.M., A.Y. Ng, and M.I. Jordan, Latent dirichlet allocation. Journal of machine Learning research, 2003. 3(Jan): p. 993-1022.

2. Sun, S., C. Luo, and J. Chen, A review of natural language processing techniques for opinion mining systems. Information Fusion, 2017. 36: p. 10-25.

3. Xu, Z., et al., Crowdsourcing based social media data analysis of urban emergency events. Multimedia Tools and Applications, 2017. 76(9): p. 11567-11584.

4. Zhang, Y., et al., iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. Future Generation Computer Systems, 2017. 66: p. 30-35.

5. Jiang, Z., et al. Using link topic model to analyze traditional chinese medicine clinical symptom-herb regularities. in e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th Inter- national Conference on. 2012. IEEE.

6. Paul, M.J. and M. Dredze, You are what you Tweet: Analyzing Twitter for public health. Icwsm, 2011. 20: p. 265-272.

7. Wu, Y., et al. Ranking gene-drug relationships in biomedical literature using latent dirichlet allocation. in Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing. 2012. NIH Public Access.

8. Linstead, E., et al. Mining concepts from code with probabilistic topic models. in Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. 2007. ACM.

9. Gethers, M. and D. Poshyvanyk. Using relational topic models to capture coupling among classes in object-oriented software systems. in Software Maintenance (ICSM), 2010 IEEE International Confer- ence on. 2010. IEEE.

10. Asuncion, H.U., A.U. Asuncion, and R.N. Taylor. Software traceability with topic modeling. in Pro- ceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1. 2010. ACM.

11. Thomas, S.W. Mining software repositories using topic models. in Proceedings of the 33rd Interna- tional Conference on Software Engineering. 2011. ACM.

12. Thomas, S.W., et al. Modeling the evolution of topics in source code histories. in Proceedings of the 8th working conference on mining software repositories. 2011. ACM.

13. Cristani, M., et al. Geo-located image analysis using latent representations. in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. 2008. IEEE.

14. Eisenstein, J., et al. A latent variable model for geographic lexical variation. in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 2010. Association for Computa- tional Linguistics.

15. Tang, H., et al., A multiscale latent Dirichlet allocation model for object-oriented clustering of VHR panchromatic satellite images. IEEE Transactions on Geoscience and Remote Sensing, 2013. 51(3): p. 1680-1692.