Topic Modelling using Deep Learning Framework

by

Tumuluri Krishna Madhav ***411986***

Veerabathina Rufus ***411991***

Jyotin Kumar Madas ***411934***

Under the guidance of

Dr. S Nagesh Bhattu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534102, INDIA

MAY - 2023

Topic Modelling using Deep Learning Framework

Thesis submitted to

National Institute of Technology Andhra Pradesh

For the award of the degree

Of

Bachelor of Technology

By

Tumuluri Krishna Madhav-(Roll No: 411986)

Veerabathina Rufus-(Roll No: 411991)

Jyotin Kumar Madas -(Roll No: 411934)

Supervisor

Dr. S Nagesh Bhattu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ANDHRA PRADESH -534102, INDIA

APPROVAL SHEET

This thesis/dissertation/report entitled “Topic Modelling using Deep Learning Framework” by Tumuluri Krishna Madhav,Veerabathina Rufus,Madas Jyotin Kumar is approved for the degree of “Bachelor of Technology”.

Supervisor (s)

_____

_____

Examiners

_____

_____

_____

Chairman

_____

Date: _____

Place: Tadepalligudem

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Tumuluri Krishna Madhav	Veerabathina Rufus	Jyotin Kumar Madas
Roll No: 411986	Roll No: 411991	Roll No: 411934
Date:	Date:	Date:

CERTIFICATE

It is certified that the work contained in the thesis titled **"Topic Modelling using Deep Learning Framework**," by "Tumuluri Krishna Madhav, Veerabathina Rufus, Jyotin Kumar Madas, bearing Roll No's: 411986,411991,411934" respectively has been carried out under my supervision and that this work has not been submitted elsewhere for a degree*

Signature of Supervisor
Name: Dr. S Nagesh Bhattu
Department: COMPUTER SCIENCE AND ENGINEERING
N.I.T. Andhra Pradesh
May,2023

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. S Nagesh Bhattu, Assistant Professor at Department of Computer Science and Engineering, NIT Andhra Pradesh, for his guidance and help extended at every stage of this project work.

We are greatly thankful to all the staff members of the Computer Science department and all my well wishers, classmates and friends for their inspiration and help.

Tumuluri Krishna Madhav	Veerabathina Rufus	Jyotin Kumar Madas
Roll No: 411986	Roll No: 411991	Roll No: 411934
Date:	Date:	Date:

ABSTRACT

Topic modeling is an important task in natural language processing, which involves identifying the underlying topics or themes present in a collection of text documents. Traditionally, topic modeling has been implemented using statistical methods such as Latent Dirichlet Allocation (LDA),etc., which assume that the topics are generated by a probabilistic process.A sparse topical coding(STC) abstract is a concise representation of text data that captures the most relevant topics using a limited number of codes.Bidirectional Poison Topic Model (BPTM) is a probabilistic model that assumes each document is generated from a mixture of different topics, and each word in the document is generated from a specific topic.

Inference process of BPTM needs to be customized based on the model's complexity, and it is difficult to automate the design of inference processes.The inference processes of conventional BPTMs are difficult to scale efficiently on large text collections or leverage parallel computing facilities like GPUs.It is usually inconvenient to integrate BPTMs with other deep neural networks (DNNs) for joint training.

In Bidirectional Adversarial Topic model(BATM)[32], Adversarial training is employed in neural topic models.In this project, we propose a deep neural framework for topic modeling based on the BATM and STC framework . This involves training two neural networks: a generator network, which is a re-model of STC, that produces topics from the input text, and a discriminator network that determines whether the generated topics are real or fake.

The proposed framework attempts to improve upon traditional topic modeling techniques by incorporating more complex and nuanced representations of the underlying topics. By utilizing a deep neural network architecture, the framework can potentially capture more subtle patterns and relationships in the data that may not be apparent to traditional statistical models.

To evaluate the performance of the proposed framework, we will train and test it on a large corpus of text data. We will evaluate the framework based on its ability to accurately identify and classify topics in unseen data, as well as its computational efficiency and scalability.The results of this project that are obtained are presented in the experimental section.

TABLE OF CONTENTS

Content	Page No
TITLE	i
APPROVAL SHEET	ii
DECLARATION	iii
CERTIFICATE	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xii

CONTENTS

1.	**Introduction**	**1**
2.	**Literature Review**	**3**
	2.1 Latent Dirichlet Allocation	3
	2.2 Sparse Topical Coding	4
	2.2.1 Hierarchical Sparse Coding	6
	2.3 Latent Aspect Mining via Exploring Sparsity	7
	2.4 Adversarial Neural Topic Model	9
	2.5 Neural Topic Modeling with Bidirectional Adversarial Training	10
	2.6 Neural STC	12
3.	**Proposed Methodology**	**14**
	3.1 Datasets	14
	3.2 Data Preprocessing	14
	3.2.1 Spacy Tokenizer	15
	3.3 NSTC Model with Restructured Generator(Proposed Model)	15

3.3.1 Model Architecture 17
4. Experimental Setup and Implementation 17
5. Results and Analysis 18
6. Conclusion and future work 24
Reference 25

LIST OF FIGURES

FIGURE NO	NAME	PAGE NO
2.1	Graphical model representation of Latent Dirichlet Allocation (LDA).	**4**
2.2	Sparse Topical Coding (STC) model.	**5**
2.3	Sparse Aspect Coding Model (SACM).	**8**
2.4	The framework of the Adversarial-neural Topic Model (ATM)	**10**
2.5	The framework of the Bidirectional Adversarial Topic (BAT) model.	**12**
2.6	Schematic overview of NSTC	**13**

LIST OF TABLES

Table NO	NAME	PAGE NO
Table 1	Topic coherence metrics comparison to BATM-model and proposed model based on epoch settings(50,100,150,200) for 20Newsgroup dataset	**18**
Table 2	Topic coherence metrics comparison to BATM-model and proposed model based on number of topics(K) settings(20,50,75,100) for 20Newsgroup dataset.	**19**
Table 3	Table 3: Topic coherence metrics comparison to BATM-model and proposed model based on epoch settings(50,100,150,200) for 20Newsgroup dataset with Modified LOSS.	**20**
Table 4	Table 3: Topic coherence metrics comparison to BATM-model and proposed model based on number of topics(K) settings(20,50,75,100) for 20Newsgroup dataset with Modified LOSS.	**21**

Table 5	Topic coherence metrics for BAT model with alpha value 10 for (20,50,75,100) topics for the 20Newsgroup dataset.	22
Table 6	Topic coherence metrics for BAT model with alpha value 1 for (20,50,75,100) topics for the 20Newsgroup dataset.	22
Table 7	Topic coherence metrics for BAT model with alpha value 0.1 for (20,50,75,100) topics for the 20Newsgroup dataset.	23
Table 8	Topic coherence metrics for BATmodel with alpha value 0.01 for (20,50,75,100) topics for the 20Newsgroup dataset.	23
Table 9	Topic coherence metrics for BAT model with alpha value 0.001 for (20,50,75,100) topics for the 20Newsgroup dataset.	24

LIST OF ABBREVIATIONS

NOTATION		MEANING
LDA	-	Latent Dirichlet Allocation
STC	-	Sparse Topical Coding
ATM	-	Adversarial-neural Topic Model
BATM	-	Bidirectional Adversarial Training Model
NSTC	-	Neural Sparse Topical Coding
SACM	-	Sparse Aspect Coding Model
LARAM	-	Latent Aspect Rating Analysis Model
MAP	-	Maximum A Posteriori
TF	-	Term Frequency
IDF	-	Inverse Document Frequency
BPTM	-	Bayesian Probabilistic Topic Models
GAN	-	Generative Adversarial Network
NLP	-	Natural Language Processing

Chapter 1

Introduction

Topic modeling is a machine learning technique used in natural language processing (NLP) to discover hidden topics or themes in a collection of documents. The goal of topic modeling is to automatically identify patterns and groupings in text data, without the need for human annotation or labeling.

The most well-known and effective series of models is the Bayesian probabilistic topic models (BPTMs), of which latent Dirichlet allocation (LDA) is an example[31].Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus[4].

A BPTM often describes a probabilistic generative model that produces data for a document using a structure of latent variables drawn from pre-specified distributions and linked by the Bayes theorem. These latent variables capture topics.A variational inference (VI) and Monte Carlo Markov Chain sampling are two examples of (Bayesian) inference processes that can be used to learn a BPTM[31].

Latent Dirichlet allocation (LDA)[4] is a generative probabilistic model of a corpus. The fundamental concept is that documents are viewed as random mixtures of latent topics, with each topic being represented by a distribution over words[1,4].

Sparse Topical Coding (STC)[5] is a topic modeling technique that uses a sparse coding framework to represent documents as a linear combination of a small number of "topic atoms." By using a sparse representation, STC[5] can effectively capture the most important words and topics in a document, while ignoring irrelevant or noise words. STC[5] has been shown to outperform traditional topic modeling techniques like LDA[4].

Neural topic models, a novel and rapidly prominent study field created when topic modeling and deep neural networks collided, have over a hundred models established and a variety of uses in neural language processing(NLP), including text recognition [31].

The Adversarial Neural Topic Model (ATM)[23] is a topic modeling technique that uses a generative adversarial network (GAN) to learn the topic distributions of a corpus of text data. In ATM, the generator network produces a set of topic distributions for each document, while the discriminator network tries to distinguish between the real and generated topic distributions.

Bidirectional adversarial training (BAT)[32] is a technique used to improve the performance of generative models such as GANs. BAT involves training two discriminators, one that evaluates the quality of the generated data and another that evaluates the quality of the real data. By training the generator to produce data that can fool both discriminators, BAT can improve the overall quality of the generated data.

Neural Sparse Topical Coding (NSTC)[22] is a topic modeling technique that combines the strengths of sparse topical coding(STC)[5] and deep neural networks. In NSTC[22], each document is represented as a sparse linear combination of a small number of "topic atoms" learned from the data. However, unlike traditional sparse topical coding, the topic atoms in NSTC[22] are learned through a neural network, allowing for a more flexible and expressive model.

Our proposed methodology, Modified-BATM is based on these models (BATM[32],NSTC[22],STC[5]) is an attempt to collide the advantages of neural networks like GAN with STC[5] topic model and as STC[5] has proved that it can provide better metrics in compared to LDA[4] and also as deep learning architecture has various advantages in combination with topic-modeling. We have compared our results to the BAT[32] paper and achieved a few promising results on datasets(20Newsgroup and NYTimes dataset).

Chapter 2

Literature Review

2.1 Latent Dirichlet Allocation

LDA is a generative model, which means it conjures up a hypothetical procedure to mimic how text is produced. The foundation of LDA is the idea that each document consists of a variety of topics, and that each topic is a distribution over a collection of words [1].

Each topic is represented as a probability distribution over words in LDA, while a document is represented as a probability distribution over topics [4]. The model presupposes that each word in a text is produced by first choosing a topic from the topic distribution of the document, and then choosing a word from the word distribution of the topic.

Various methods have been proposed to estimate LDA parameters, such as variational method [9], expectation propagation [10] and Gibbs sampling [11].

Gibbs sampling is a Monte Carlo Markov-chain algorithm, a powerful technique in statistical inference, and a method of generating a sample from a joint distribution when only conditional distributions of each variable can be efficiently computed. According to our knowledge, researchers have widely used this method for the LDA. Some works related based on LDA and Gibbs, such as [12,13].

Bayesian inference is a statistical method used by LDA to estimate the model's parameters [2]. Finding the subject and word distributions that best explain the observed data (i.e., the words in the texts) is the objective. The distributions are modified iteratively throughout this process until they reach a stable outcome.The assumptions are as:

- Documents are mix of topics
- Topics are a mixture of tokens (or words).

Hyperparameters:

- Document-topic density factor ‘α’.
- topic-word density factor ‘β’ [4].

The joint distribution of a topic mixture θ,a set of N topics z,and a set of N words is given by:

$$p(\theta,z,w|\alpha,\beta)=p(\theta|\alpha)\prod_{n=1}^{N}p(z_n|\theta)p(w_n|z_n,\beta)\ [4]$$

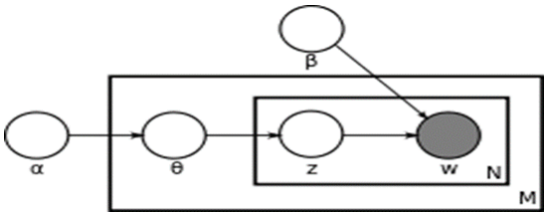


Figure 2.1: Graphical model representation of Latent Dirichlet Allocation (LDA) [4].

Generative process proceeds as: choose N ~ Poisson(ξ) and then choose θ ~ Dir(α). For each of the N word W_n we choose a topic Z_n ~ Multinomial(θ) and choose a word W_n from $p(W_n|Z_n,\beta)$, a multinomial probability conditioned on the topic Z_n [4].

2.2 Sparse Topical Coding

Natural language processing (NLP) uses the Sparse Topical Coding (STC) technique for document modeling and representation. It is a technique for breaking down a group of documents into a sparse assortment of latent subjects.

Sparse topical coding (STC), a non-probabilistic topic model framework for identifying latent representations of big data sets. STC reduces the normalization constraint of admixture proportions and the constraint of constructing a normalized likelihood function, in contrast to probabilistic topic models [5].

In STC, each latent topic is a distribution over words, and each document is represented as a sparse linear combination of latent topics. The objective of STC is to determine the set of latent topics that, while fostering sparsity in the representation, can most effectively account for the observed data (i.e., the words in the documents) [5].The STC's sparsity requirement promotes the model to describe a document with a minimal number of topics. This aids in reducing the representation's dimensionality and enhances the themes that result's interpretability.

STC is predicated on the idea that a process that involves choosing a small number of latent topics and then producing words from those topics produces the observed data (i.e., the words in the texts). The model determines the optimum collection of latent themes by applying a statistical method known as Bayesian inference to estimate the model's parameters.

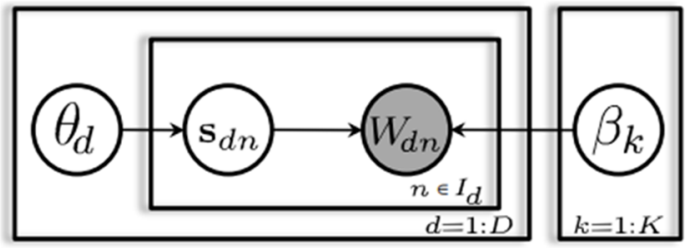


Figure 2.2 :Sparse Topical Coding (STC) model [5].

- V-Vocabulary={1,2,...,N}
- K-number of topics
- N-number of words in each topic
- β -Dictionary =R_{KxN}
- θ-Document code
- S_n-word code
- W_n-known word

Assumptions

- For each document the word codes S_n are conditionally independent given its document code θ.
- The observed word counts are independent given their latent representations S.

θ represents the topic distribution of a document in sparse topical coding, while β represents the word distribution of a topic

- A dictionary β is sampled from a uniform distribution on P. The following technique can then be used to explain how each document was created:
- sample the document code θ from a prior p(θ).
- for each observed word $n \in I$
    a. sample the word code S_n from a conditional distribution $p(S_n|\theta)$.
    b. b.sample the observed word count W_n from a distribution with the mean being $S_n^T\beta_n$ [5].

The above generating procedure defines a joint distribution

$p(\theta,s,w|\beta) = p(\theta)\mathbf{\Pi}_{n\in I}\, p(S_n|\theta)p(W_n|S_n, \beta)$.

2.2.1 Hierarchical sparse coding:

This step involves finding the codes Θ when β is fixed. There are two steps [5]:

1. $\text{Min}_{sn}\, l(sn, \beta) + \gamma||sn - \theta||^2_2 + \rho \Sigma_k s_{nk}$, s.t. : $sn \geq 0$ (θ is fixed optimize over s)
2. $\min \theta\ \lambda||\theta||_1 + \gamma\ \Sigma_{n\in I}||sn - \theta||^2_2\, 2$, s.t. : $\theta \geq 0$. (s is fixed to optimize over θ)

Dictionary learning:

After we have inferred the latent representations (θ, s) of all the documents, we update the dictionary β by minimizing the log-Poisson loss, which is convex and can be efficiently solved with a high-performance method, such as projected gradient descent [6].

2.3 Latent Aspect Mining via Exploring Sparsity

Information mining and extraction from review texts have received a lot of scholarly attention, including sentiment analysis [14,15], opinion summarization and identification [16,17].However, the majority of these models only attempt to analyze the overall sentiment of review texts. It is vital to find additional fine-grained information about the things in order to give users more thorough insights of various reviews [8]. Aspect-based sentiment analysis has been conducted for this event [18,19].

The inability of probabilistic topic models, such as LDA-based models, to deal with aspect sparsity in texts, is one of its drawbacks [7].

Aspect sparsity is the observation that most reviews' text content only briefly touches on a few elements rather than covering them fully. In fact, the aspect sparsity problem is frequently seen in real-world reviews. For example, let's look at the hotel industry, which has a variety of factors like Price, Room, Location, Cleanliness, Food, Service, etc. Users frequently remark on some aspects of a review but not always all of the aspects. Another example involves a certain hotel with a stellar reputation for its cuisine. It is more likely that a normal review of this hotel will focus on the food, while omitting to highlight other factors like value and room, especially for short reviews.It is more likely that a normal review of this hotel will focus on the food, while omitting to highlight other factors like value and room, especially for short reviews [8].

The main obstacle for traditional probabilistic topic models such as LDA in LARAM [20] is to handle aspect sparsity is that topic or aspect proportions are modeled as normalized distributions, namely, the sum of each aspect proportion should be one, so applying a sparsity inducing l1-regularizer as in lasso [21] is not helpful. Non-probabilistic sparse coding techniques, such as the Sparse Topical Coding (STC) model [7], can tackle the above sparsity issue.The model SACM is an application of STC mentioned in [5].

Latent variables- user intrinsic aspect and item intrinsic aspect. User intrinsic aspect interest denotes the intrinsic interest for each aspect of a particular user [8].This notion is different from the notion of aspect weight defined in [9]. Item intrinsic aspect quality represents the intrinsic quality for each aspect of a particular item. Uses MAP Optimization problem and Block-Coordinate gradient descent. The document code θd is modeled as the Hadamard product between the user intrinsic aspect interest t_{ud} and the item intrinsic aspect quality q_{hd} [8].

- $Y^{A}_{dk} \sim N(q_{hd}k, \alpha^{2} t^{2}_{udk})$

$\min \Sigma_{d,n \in Id} (\gamma||sn - \theta||^{2}_{2} + \rho||sn||_{1}) + \Sigma_{d,n \in Id} l(sdn, \beta) +$
$\sum_{d} (1 \mid 2c^{2}) (Y_{d} - \sum_{k} \eta_{dk} Y^{A}_{dk})^{2} + \sum_{d} \sum_{k} [\log(\alpha t_{udk}) + 1/2\alpha^{2} t^{2}_{udk} (Y^{A}_{dk} - q_{hdk})^{2}] + \sum_{u} \lambda // tu //_{1}$

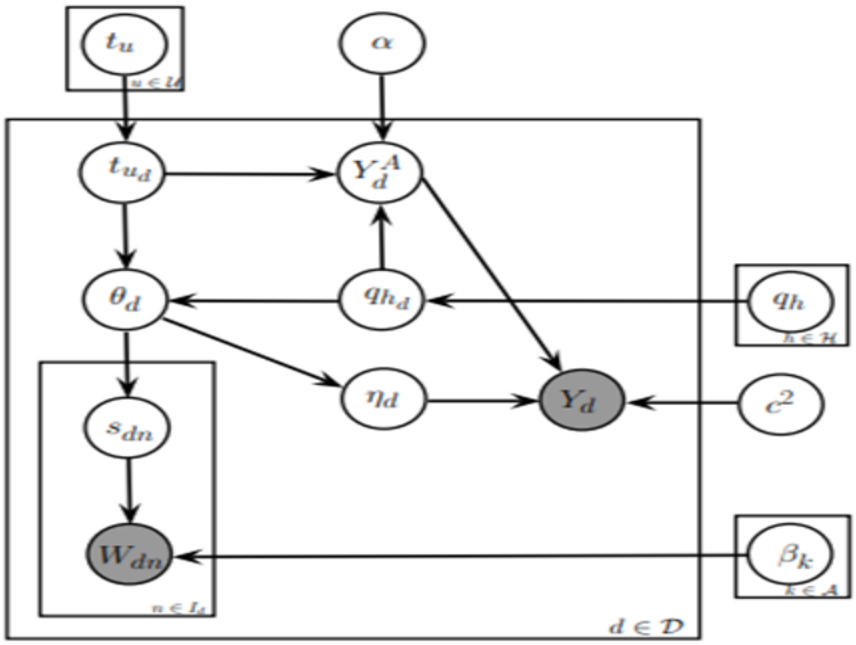


Figure 2.3: Sparse Aspect Coding Model (SACM) [8].

2.4 Adversarial-neural Topic Model(ATM model)

Thematic structure in literature is frequently discovered using topic models. Traditional topic models, however, frequently call for specialized inference techniques for the relevant tasks. Additionally, they are not intended to produce semantic word representations. Adversarial-neural Topic Model (ATM) is a neural topic modeling method based on the Generative Adversarial Nets (GANs) [23].

The resulting embeddings encode numerous semantic relations (similarity or analogies) and are helpful for NLP tasks[24, 25].Due to their increased effectiveness in encoding words as continuous vectors in a low-dimensional space, word embeddings (such as Word2vec[10], GloVe[28], fastText[29, 30], and probabilistic fastText[31]) have attracted growing interest in recent years.

The Adversarial Neural Topic Model (ANTM) is a machine learning algorithm that combines the power of generative models with the ability to learn from adversarial examples. This model is particularly useful for analyzing large datasets that contain unstructured text data, such as social media posts, news articles, and research papers [22].

It has the following layers:

1. Generator: The G network contains three layers, the K-dimensional document-topic distribution layer, the S-dimensional embedding layer and the V -dimensional document-word distribution layer as seen in Figure 2.4 .
2. Discriminator: The D network employs the d_f and the d_r as input and outputs a scalar as shown in Figure 2.4. A higher D_{out} means that the discriminator is prone to consider the input data as a real document and vice versa.

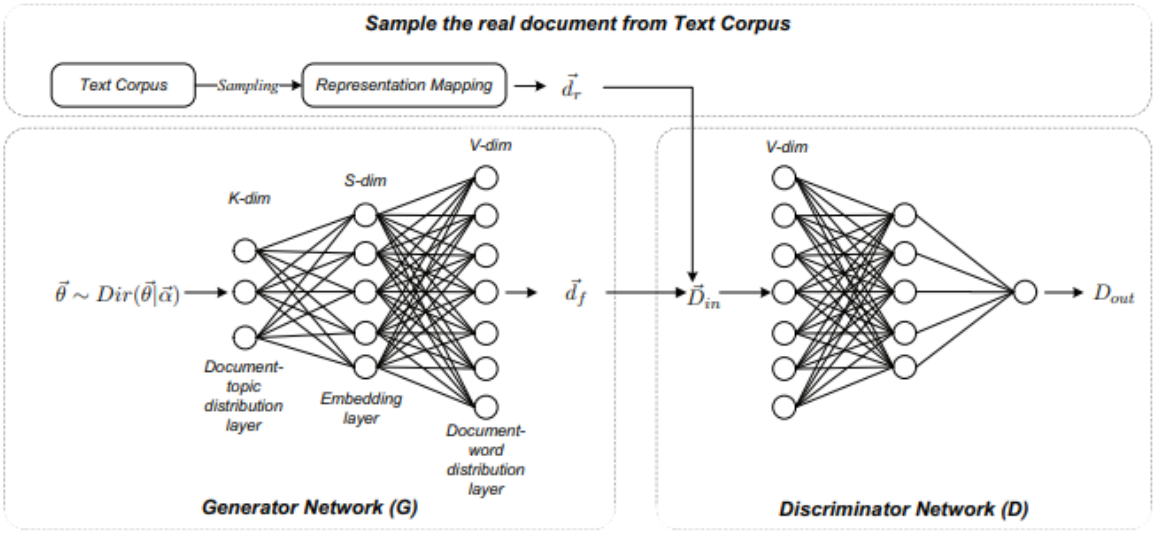


Figure 2.4: The framework of the Adversarial-neural Topic Model (ATM) [22].

2.5: Neural Topic Modeling with Bidirectional Adversarial Training (BAT model)

In this research, a brand-new model was introduced that marks the first time bidirectional adversarial training has been used for neural topic modeling. The suggested BAT [32] creates a two-way projection between the distribution of document topics and words. It employs an encoder for topic inference and a generator to extract the semantic patterns from texts.

A normalized V-dimensional vector weighted by TF-IDF serves as the representation for each document d.

The suggested model consists of three parts:

- Encoder: The encoder learns a mapping function to convert the distribution of document words to distribution of document topics. A V-dimensional document-word distribution layer, an S-dimensional representation layer, and a K-dimensional document-topic distribution layer are all present, as illustrated in the figure below, where V and K stand for vocabulary size and topic number, respectively.

- Generator: Unlike an encoder, a generator offers an inverse projection from document-topic distribution to document-word distribution and consists of layers for document-topics in K dimensions, representations in S dimensions, and document-word distribution layers in V dimensions. The dirichlet distribution is used to determine document-topic distribution.
- Discriminator: As illustrated in the accompanying image, the discriminator D is made up of three layers: a V + K-dimensional joint distribution layer, an S-dimensional representation layer, and an output layer. Real distribution pair pr and fake distribution pair pf are used as input, while Dout is output to determine whether the input sources are real or phony. A greater Dout value, in concrete terms, indicates that D is more likely to forecast the input as real, and vice versa.
- $tf_{i,d} = n_{i,d} / \Sigma_v \, n_{v,d}$
- $idf_i = \log |C| / |C_i|$
- $\text{tf-idf}_{i,d} = tf_{i,d} \times idf_i$
- $d^i{}_r = \text{tf-idf}_{i,d} / \Sigma_v \, \text{tf-idf}_{v,d}$

There are two components to TF-IDF. IDF (inverse document frequency) and TF (term frequency).The way term frequency works is by examining how frequently a specific phrase is used in relation to the document. There are various frequency definitions or metrics, including:

How frequently (or infrequently) a term appears in the corpus is examined via inverse document frequency. The following formula is used to compute IDF, where t is the term (word) whose frequency we want to gauge and N is the total number of documents (d) in the corpus (D). The number of papers that contain the phrase "t" serves as the denominator.

We require IDF since terms like "of," "as," "the," etc. commonly arise in an English corpus and need to be corrected. Therefore, by using inverse document frequency, we can reduce the weighting of frequent terms while increasing the impact of infrequent terms.Finally, IDFs can be obtained from the dataset being used in the experiment at hand or from a background corpus, which corrects for sampling bias.

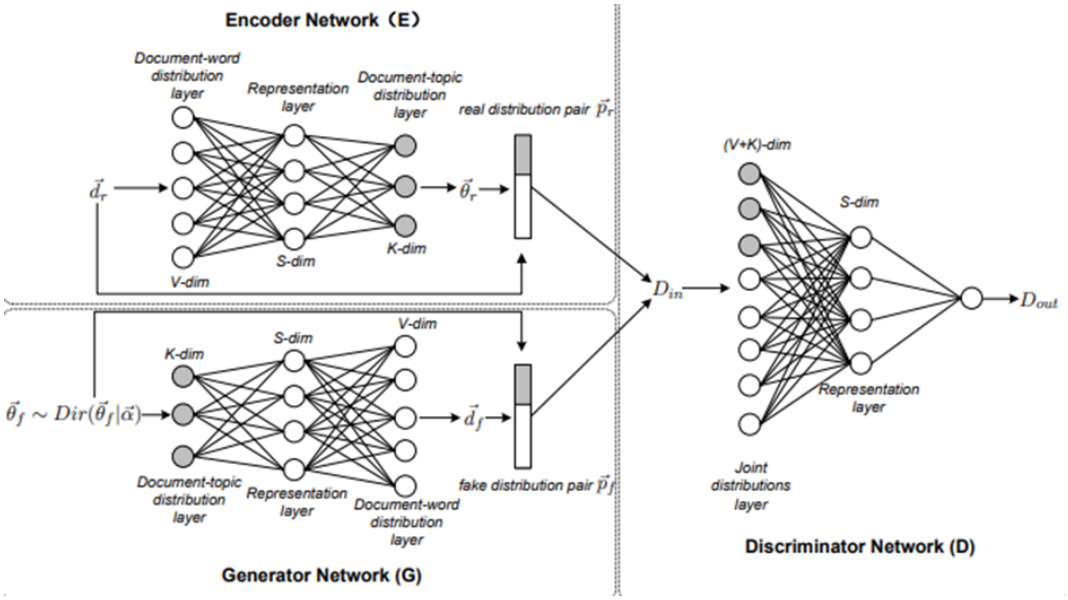


Figure 2.5: The framework of the Bidirectional Adversarial Topic (BAT) model [32].

2.6 :Neural STC

A particular kind of topic model called Neural Sparse Topical Coding (NSTC)[22] seeks to find latent themes in a group of documents while enforcing sparsity on the topic representations that are produced. In this STC is re-modelled by Neural-Networks for easy inference and back-propagation.

The variants and applications of these probabilistic topic models are, however, constrained by the extensions of these models, which necessitate carefully crafted graphical models and re-deduced inference methods. Neural Sparse Topical Coding (NSTC) sparsity-enhanced topic model is based on the Sparse Topical Coding (STC) sparsity-enhanced topic model. It focuses on using back propagation to replace the intricate inference process, making it simple to investigate model extensions [22].

Neural Sparse Topical Coding (NSTC) jointly utilizes word embeddings and neural networks with a sparsity-enhanced topic model, Sparse Topical Coding (STC).

In this paper, STC uses a neural network to streamline its backpropagation inference process. After generating the topic dictionary from neural network, our model follows the generative story below for each document d:

1. For each word n in document d:

   (a) Sample a latent variable word code $S_{d,n} \sim f_g(d, n)$.

   (b) Sample the observed word count $W_{d,n}$ from $p(W_{d,n}|S_{d,n}, \beta_n) \sim \text{Poisson}(S_{d,n} * \beta_n)$

The layers:

1. Input layer (n, d): A word n of document $d \in D$, where D is a document set.
2. Word embedding layer ($W\ E \in R^{N\times 300}$): Supposing the word number of the vocabulary is N, this layer devotes to transform each word to a distributed embedding representation.
3. Word code layers ($s_d \in R^{N\times K}$): These layers generate the K-dimensional word code of input word n in document d.

   $s(d, n) = f_s(d, n)$

   where f_s is a multilayer perceptron**.**
4. Topic dictionary layers ($\beta \in R^{N\times K}$): These layers aim at converting W E to a topic dictionary similar to the one in STC.

   $\beta(n) = f_\beta(W\ E)$, where f_β is a multilayer perceptron. We normalize each column of the dictionary via the simplex projection as follow:

   $\beta_{.k} = \text{project}(\beta_{.k}), \forall k$
5. Score layer ($C_{d,n} \in R^{1\times 1}$): NSTC outputs the matching score of a word n and a document d with the dot product of s(d, n) and β(n) in this layer. The output score is utilized to approximate the observed word count wd,n. $C(d, n) = s(d, n) * \beta(n)$

   Given the count wd,n of word n in document d, we can directly use it to supervise the training process. According to the architecture of our model, for each word n and each document d, the cost function is: $L = l(w_{d,n}, C(d, n)) + \lambda||s_{d,n}||_1$, where l is the log-Poisson loss, λ is the regularization factors.

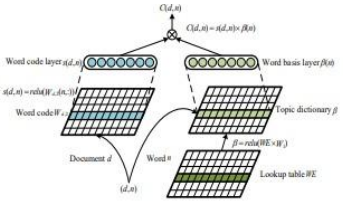


Figure 2.6 :Schematic overview of NSTC

Chapter 3

Proposed Methodology

3.1 Datasets

In this work, experimentation is done on 20Newsgroups dataset . This dataset iswidely used in natural language processing research.

The 20 Newsgroups dataset consists of text documents that were originally posted on different newsgroups on the internet. Each document in the dataset is categorized into one of 20 different topics, such as politics, religion, or sports. The documents are relatively short, typically ranging from a few sentences to a few paragraphs in length.

3.2:Data Preprocessing

In Natural Language Processing (NLP), cleaning, altering, and preparing the text data for subsequent analysis is known as data preparation. Here are a few typical NLP preprocessing methods:

- Tokenization: The process of isolating particular words or phrases from a text.
- Stop words can be eliminated from the text since they are frequent terms like "the," "a," "an," and "in" that have little to no meaning.
- Lowercase: Changing every word to lowercase to maintain text uniformity.
- Lemmatization and stemming: Getting words back to their basic structure. Lemmatization employs a dictionary to map words to their basic form.
- Spell Check: Fixing grammatical and spelling mistakes in the text.
- Punctuation Removal: Eliminate all punctuation from the text.
- Eliminating HTML tags: Getting rid of HTML tags from the content.
- Remove special characters from the text. This brings us to number eight.

- Taking out URLs: Taking out URLs from the text.
- Removing Numbers: Erasing the text's use of numbers.
- Eliminating emojis from the text is option 11.
- These preprocessing methods are frequently combined, and the precise order of processes can change based on the particular use case and the type of text input.

3.2.1: Spacy Tokenizer:

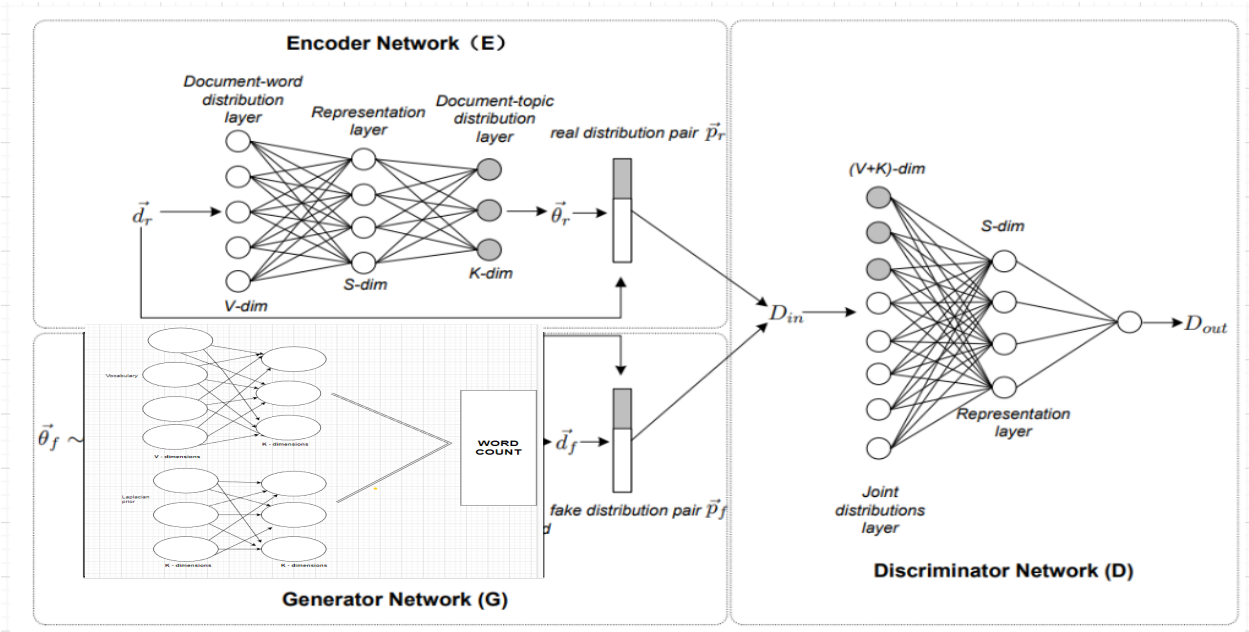
Popular open-source Python natural language processing (NLP) library Spacy is widely used. It offers a variety of functions for part-of-speech tagging, named entity recognition, dependency parsing, tokenization, and text data preparation.

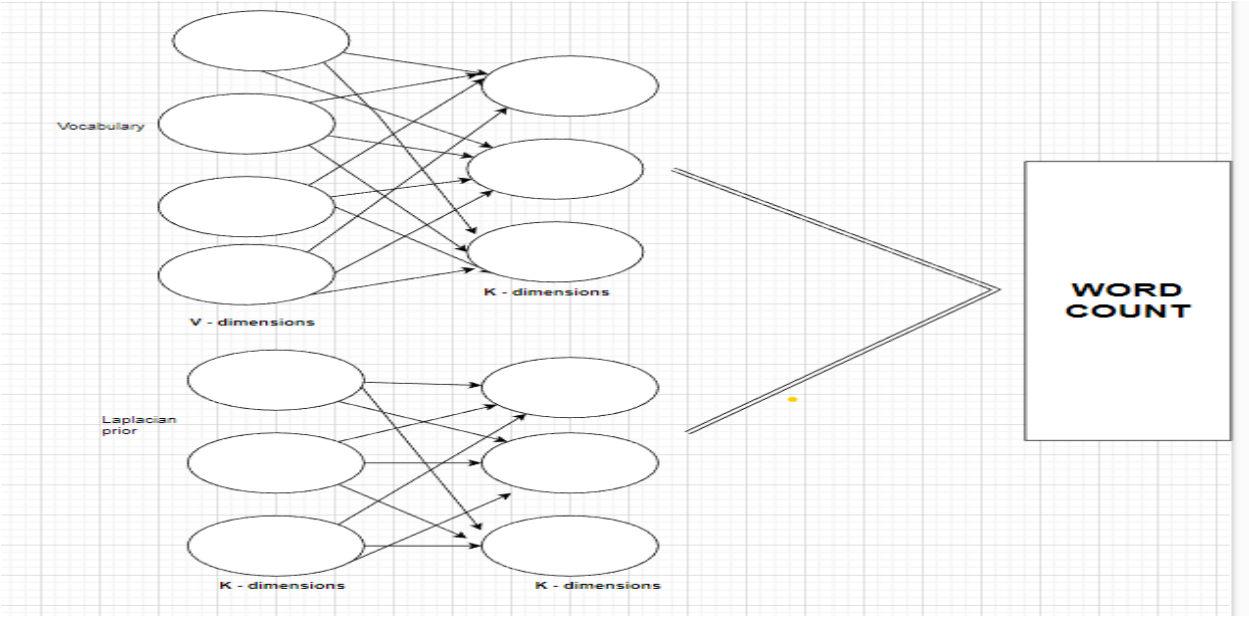
Tokenization is the process of dividing the text into discrete tokens, or units of meaning. The Tokenizer class is used in Spacy to do tokenization.

Spacy offers other preprocessing features in addition to tokenization. A list of stop words in a given language can be obtained using the stop_words attribute, for instance, and those words can then be removed from the text.

3.3 NSTC Model with Restructured Generator(Proposed Model):

In our proposed methodology, we are trying to incorporate the STC model in BATM paper architecture.The Generator has been re-structured from NSTC.To In this deep learning framework we introduce the Neural Network layers in the Generator module. This Module is based on STC architecture where we sample from the output of a neural network here whereas in STC we sample from distributions.


Encoder Network（E）
Document-word distribution layer
Representation layer
Document-topic distribution layer
real distribution pair $\vec{p}_r$
$\vec{d}_r$
$\vec{\theta}_r$
V-dim
S-dim
K-dim
$\vec{\theta}_f \sim$
WORD COUNT
$\vec{d}_f$
fake distribution pair $\vec{p}_f$
Generator Network (G)
D_{in}
(V+K)-dim
S-dim
Representation layer
D_{out}
Joint distributions layer
Discriminator Network (D)


Vocabulary
V - dimensions
K - dimensions
Laplacian prior
K - dimensions
K - dimensions
WORD COUNT

3.3.1 Model Architecture

The Generator module is divided into 2 parts. One is the Neural Network where Laplacian prior is given as input and word code is provided as output. The dimension of input is K and output dimension is K. The second part of Neural network is

4 Experimental Setup and Implementation:

The implementation is done with batch_size of 512 and the vocabulary size of 20NewsGroup dataset is 13290. The Generator module takes Laplacian Distribution as input and gives the word code in output. The second part of the Generator Module takes Vocabulary as input and gives β which is used to find word count. ReLU is applied after every layer to have positive values.

The parameters sent to run this model are no_below 0.5 and no_below 5 .We have used different epochs and different n_topics to explore the outputs.

4.2 Loss Function

L1 Normalisation is added to adversarial loss function used in BATM[32] model.This L1 Norm is added in Loss function of Generator which induces sparsity.

```
L1_norm=sum(p.abs().sum() for p in self.generator.parameters())
Loss_G=original_loss_g+(L1_lambda*L1_norm)
```

4.3 Topic Coherence Evaluation

To compare the performance of the proposed approach, experiments are conducted on 20NewsGroup with five topic number settings [20, 50, 75, 100]. The coherence values are listed in Table 1 and Table 2 with different epochs and different n_topics .Table 3 and Table 4 contains the topic coherence values with modified loss (addition of L1 normalization) with different epochs and different n_topics [20, 50, 75, 100].

RESULTS AND ANALYSIS:-

Metrics	BATM-Model				Proposed Model			
epochs	50	100	150	200	50	100	150	200
topic diversity	0.52	0.1566	0.0733	0.0667	0.0633	0.0633	0.06	**0.06**
c_v	0.2846	0.4981	0.5591	0.5373	**0.3894**	0.2954	0.3666	0.3391
c_uci	-7.4228	0.7777	0.3526	0.3329	**-2.3799**	-2.4424	-1.8783	-2.5951
c_npmi	-0.2623	0.0152	0.0475	0.0447	-0.0488	-0.0542	-0.0373	-0.0540
topic coherence	-302.7113	-194.4665	-179.1165	-182.1935	-261.7979	-252.1754	-277.0636	-257.9103

Table 1: Topic coherence metrics comparison to BATM[32]-model and proposed model based on epoch settings(50,100,150,200) for 20Newsgroup dataset.

Metrics	BATM_Model				Proposed-Model			
No of topics	20	50	75	100	20	50	75	100
topic diversity	0.52	0.1146	0.0897	0.074	0.0633	0.036	0.0302	0.0286
c_v	0.2846	0.4040	0.4164	0.4064	**0.3894**	0.3303	0.3008	0.3571
c_uci	-7.4228	-0.1966	-0.3809	-0.4157	**-2.3799**	-3.2107	-5.1150	-4.7908
c_npmi	-0.2623	-0.0028	-0.0158	-0.0106	**-0.0488**	-0.1038	-0.1704	-0.1629
topic coherence	-302.7113	-242.0611	-239.4626	-263.3751	**-261.7979**	-249.0781	-328.3226	-284.3133

Table 2: Topic coherence metrics comparison to BATM[32]-model and proposed model based on number of topics(K) settings(20,50,75,100) for 20Newsgroup dataset.

Metrics	BATM-Model				Proposed Model			
epochs	50	100	150	200	50	100	150	200
topic diversity	0.8366	0.323	0.8466	0.8266	0.06	0.06	0.0533	**0.0533**
c_v	0.653	0.617	0.656	0.6787	0.319	0.332	0.3344	0.3466
c_uci	-9.86	-9.792	-9.656	-9.753	-3.757	-3.042	-2.22	-1.937
c_npmi	-0.35	-0.351	-0.347	-0.351	-0.115	-0.078	-0.031	-0.027
topic coherence	-194.9	-236.84	-183.21	-190.124	-245.365	-257.27	-261.16	-274.500

Table 3: Topic coherence metrics comparison to BATM[32]-model and proposed model based on epoch settings(50,100,150,200) for 20Newgroup dataset with Modified LOSS.

Metrics	BATM-Model				Proposed-Model			
No of topics	20	50	75	100	20	50	75	100
topic diversity	0.8366	0.0933	0.057	0.0413	0.06	0.044	0.0231	0.016
c_v	0.653	0.656	0.681	0.6144	0.319	0.341	0.358	0.343
c_uci	-9.86	-9.74	-10.19	-10.22	-3.757	-4.47	-4.5	-3.76
c_npmi	-0.35	-0.35	-0.367	-0.366	-0.115	-0.152	-0.141	-0.126
topic coherence	-194.9	-186.64	-215.366	-231.951	-245.365	-253.765	-307.026	-232.156

Table 4: Topic coherence metrics comparison to BATM[32]-model and proposed model based on number of topics(K) settings(20,50,75,100) for 20Newsgroup dataset with modified-LOSS

No. of topics	topic diversity	c_v	c_uci	c_npmi	Topic coherence
20	0.5366	0.6028	-9.7209	-0.3317	-216.7035
50	0.98	0.3727	-8.0962	-0.2547	-282.4945
75	0.8506	0.3540	-7.6788	-0.2505	-321.8112
100	0.6393	0.3555	-11.3397	-0.2364	-288.3714

Table 5: topic coherence metrics for BAT[32] model with alpha 10, number of topics (20,50,75,100) for 20Newsgroup dataset

No. of topics	topic diversity	c_v	c_uci	c_npmi	Topic coherence
20	0.58	0.5583	-9.6611	-0.3273	-229.72
50	0.97	0.3817	-8.2993	-0.2762	-270.0412
75	0.8533	0.3761	-7.7837	-0.2525	-224.6494
100	0.6606	0.3624	-7.1522	-0.2288	-266.5948

Table 6: Topic coherence metrics for BAT[32] model with alpha 1, number of topics (20,50,75,100) for 20Newsgroup dataset.

No. of topics	topic diversity	c_v	c_uci	c_npmi	Topic coherence
20	0.2466	0.48660	-9.4546	-0.3270	-238.9361
50d	0.9746	0.3720	-8.2966	-0.2739	-269.2993
75	0.8497	0.3739	-7.5566	-0.2419	-207.6769
100	0.65	0.3604	-7.1559	-0.2290	-299.9057

Table 7 :Topic coherence metrics for BAT[32] model with alpha 0.1, number of topics (20,50,75,100) fro 20Newsgroup dataset

No. of topics	topic diversity	c_v	c_uci	c_npmi	Topic coherence
20	0.6	0.4633	-8.6241	-0.2917	-244.4803
50	0.9826	0.3896	-8.4761	-0.2848	-306.7358
75	0.83466	0.3679	-7.8013	-0.2553	-252.7259
100	0.646	0.3583	-7.0434	-0.2232	-341.6614

Table 8:Topic coherence metrics for BAT[32] model with alpha 0.01, number of topics (20,50,75,100) for 20Newsgroup dataset.

No. of topics	topic diversity	c_v	c_uci	c_npmi	Topic coherence
20	0.5866	0.5383	-9.6387	-0.3213	-246.9454
50	0.9813	0.3758	-8.0621	-0.2665	-298.2084
75	0.8595	0.3665	-7.9164	-0.2590	-292.2930
100	0.656	0.3564	-7.0435	-0.2239	-228.2980

Table 9 :Topic coherence metrics for BAT[32], alpha 0.001, number of topics (20,50,75,100) for 20Newsgroup dataset.

Conclusion and Future Work:

In this proposed work, we have explored the incorporation of Sparse Topic Coding(STC) in a deep learning framework. Literature study has suggested that Topic Modelling with neural networks gives us promising results and better advantages in comparison to the traditional Bayesian probabilistic models. For this work, we have used the models as described in literature like STC,NSTC,BATM and we attempt to propose a new methodology .

We have done our experimentation on 20Newsgroup dataset and presented our results in the experimentation section. In future work, we hypertune the parameters to achieve better results in comparison to the existing models.

References

[1] Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, Liang Zhao.

[2] Zhang, Y., et al., iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. Future Generation Computer Systems, 2017.

[3] Zhai, K., et al. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. in Proceedings of the 21st international conference on World Wide Web. 2012. ACM.

[4] David M. Blei, Andrew Y. Ng, Michael I. Jordan on "Latent Dirichlet Allocation".

[5] Jun Zhu, Eric P. Xing on "Sparse Topical Coding".

[6] J. Duchi, S. Shalev-Shwartz, Y. Singer, & T. Chandra. Efficient projections onto the `1-ball for learning in high dimensions. In ICML, 2008.

[7] Yinqing Xu Tianyi Lin Wai Lam Zirui Zhou Hong Cheng Anthony Man-Cho So on "Latent Aspect Mining via Exploring Sparsity and Intrinsic Information".

[8] H. Wang and Y. Lu C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In KDD, pages 783–792, 2010.

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

[10] Minka, T. and J. Lafferty. Expectation-propagation for the generative aspect model. in Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence. 2002. Morgan Kaufmann Publishers Inc.

[11]Griffiths, T.L. and M. Steyvers, Finding scientific topics. Proceedings of the National academy of Sciences, 2004. 101(suppl 1): p. 5228-5235.

[12] . Tian, K., M. Revelle, and D. Poshyvanyk. Using latent dirichlet allocation for automatic categorization of software. in Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on. 2009. IEEE.

[13] Rao, Y., Contextual sentiment topic model for adaptive social emotion classification. IEEE Intelligent Systems, 2016. 31(1): p. 41-47.

[14] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In ACL, pages 417–424, 2002.

[15] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In ACL, pages 115–124, 2005.

[16] M. Hu and B. Liu. Mining and summarizing customer reviews. In KDD, pages 168–177, 2004.

[17] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In WWW, pages 519–528, 2003.

[18] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In WWW, pages 111–120, 2008.

[19] M. Hu and B. Liu. Mining opinion features in customer reviews. In AAAI, volume 4, pages 755–760, 2004.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. JMLR, 3:993–1022, 2003.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society., pages 267–288, 1996.

[22] Min Peng, Qianqian Xie, Yanchun Zhang, Hua Wang, Xiuzheng Zhang, Jimin Huang, and Gang Tian on "Neural Sparse Topical Coding".

[23] Rui Wang, Deyu Zhou and Yulan He on “Adversarial-neural Topic Model”.

[24] Francis C. Fernández-Reyes, Jorge Hermosillo Valadez, and Manuel Montes-y-Gómez. A prospect-guided global query expansion strategy using word embeddings. Inf. Process. Manage., 54(1):1–13, 2018.

[25] Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. Inf. Process. Manage., 54(6):969–984, 2018.

[26] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Proceedings of the 31 st International Conference on Machine Learning, pages 1188–1196, Beijing, China, 2014.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014), pages 1532–1543, Doha, Qatar, 2014.

[28] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL, pages 3–7, 2017.

[29] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[30] Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. Probabilistic fasttext for multisense word embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1–11, Melbourne, Australia, 2018. Association for Computational Linguistics.

[31] He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, Wray Buntine on “Topic Modeling Meets Deep Neural Networks: A Survey”.

[32] Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He ,Yuxuan Xiong, Chenchen Ye, Haiyang Xu on “Neural Topic Modeling with Bidirectional Adversarial Training”.