Topic Modeling in NLP

Two-Week Summer Internship Program

in

Computer Science and Engineering (Data Science)

by

Rohitha Ponnapalli

20951A6736

Chief Mentor : Dr. C V R Padmaja

Co-Ordinator : Dr. Indu

Institute of Aeronautical Engineering, Dundigal, 500043

Career Development Center (CDC)

MAY 2023

CERTIFICATION

This is to certify that the project report entitled **Topic Modeling in NLP** submitted by **Rohitha Ponnapalli** to the **Institute of Aeronautical Engineering, Dundigal**, in partial fulfillment for the award of the degree of B.Tech in **CSE (Data Science)** is a record of project work carried out by her in our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Signature	Signature
Dr. C V R Padmaja	Head Of The Department
Department of CSE	Department of CSE (DS)

DECLARATION

I declare that this project report entitled **Topic Modeling in NLP** submitted in partial fulfillment of the degree of **B.Tech in CSE (Data Science)** is a record of original work carried out by me under the supervision of Dr. C V R Padmaja and has not formed the basis of the award of any other degree or diploma, in this or any other Institute or University. In keeping with the ethical practice in reporting scientific information, acknowledgements have been made wherever the findings of others have cited.

Rohitha.P

Signature of the Student

Rohitha Ponnapalli

20951A6736

02-06-2023

ACKNOWLEDGMENTS

All acknowledgments are to be included here. Please restrict it to two pages. The name of the candidate shall appear at the end, without signature.

I take this opportunity to thank Sri M. Rajasekhar Reddy, Director - IARE, Dr. C. V. R. Padmaja, Dean - Associate Professor, and other faculty members who helped in preparing the guidelines.

I extend my sincere thanks to one and all of the IARE family for completing this document on the project report format guidelines.

Rohitha Ponnapalli

ABSTRACT

Topic modeling is a prominent technique in Natural Language Processing (NLP) that aims to uncover latent thematic structures within large collections of textual data. It has gained significant attention due to its applicability in various domains, such as information retrieval, sentiment analysis, recommender systems, and content analysis. This paper presents a comprehensive review of topic modeling methods and their applications in NLP. The paper presents a detailed analysis of the challenges associated with topic modeling, such as selecting the optimal number of topics, handling domain-specific data, and addressing the problem of topic sparsity. Moreover, the review investigates the application areas of topic modeling in NLP. It showcases how topic modeling has been employed in tasks such as text summarization, document clustering.

Keywords: Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Non-negative Matrix Factorization (NMF), Hierarchical Dirichlet Process (HDP).

TABLE OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human language. With the explosion of textual data in various domains, there is a growing need to extract meaningful information and discover hidden patterns within these vast collections of text. Topic modeling has emerged as a powerful technique in NLP, enabling researchers and practitioners to uncover latent themes and structures within textual data.

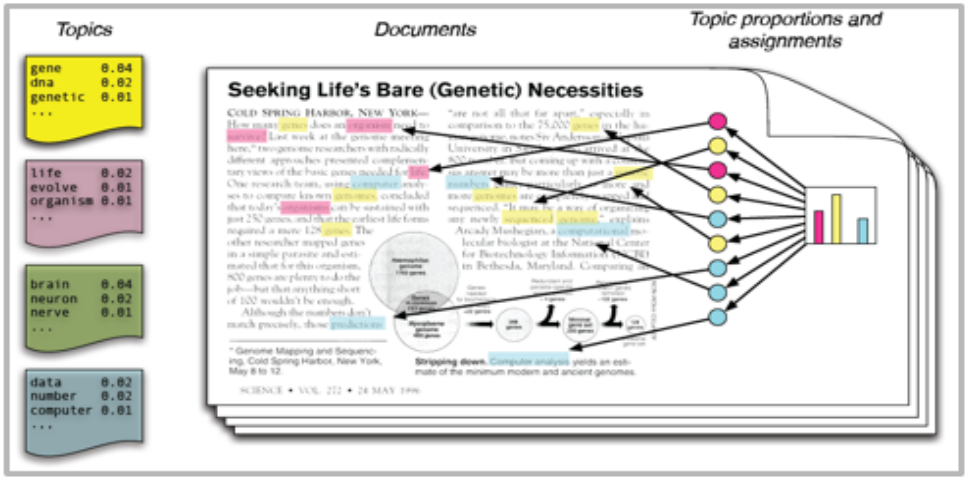

Fig 1.1 Text Classification

1.1 AIM

Topic modeling aims to automatically identify topics or themes present in a collection of documents, without requiring any prior knowledge or supervision. It provides a means to organize and understand large volumes of unstructured text, facilitating tasks such as information retrieval, document clustering, summarization, and sentiment analysis.

At its core, topic modeling treats text documents as a mixture of underlying topics, with each topic characterized by a distribution of words. The underlying assumption is that documents are generated based on the mixture of topics, and the goal is to infer these latent topics and their corresponding word distributions.

One of the most widely used topic modeling techniques is Latent Dirichlet Allocation (LDA), proposed by Blei, Ng, and Jordan in 2003. LDA assumes that each document is a mixture of topics, and each topic is a distribution over words. It uses probabilistic inference to estimate the topic proportions for each document and the word distributions for each topic.

Another popular technique is Probabilistic Latent Semantic Analysis (PLSA), which is similar to LDA but does not include a prior distribution on the topic proportions. Instead, it directly models the joint distribution of words and documents based on the observed data.

Topic modeling has seen significant advancements over the years, with the development of extensions and variations of LDA and PLSA. Non-negative Matrix Factorization (NMF) is a technique that approximates the document-term matrix using non-negative matrices, leading to interpretable topics. Hierarchical Dirichlet Process (HDP) allows for the automatic determination of the number of topics based on a hierarchical Bayesian framework.

In recent years, deep learning-based approaches have also been applied to topic modeling, leveraging neural networks to capture intricate patterns and dependencies within textual data. These approaches include Neural Variational Inference (NVI) and Neural Topic Models (NTM), which offer more flexibility and expressiveness in modeling complex relationships between words and topics.

Topic modeling has found applications across various domains. It has been used for document clustering, where documents are grouped based on their topic distributions. It has also been employed in sentiment analysis, where the sentiment of a document or a sentence is analyzed within the context of its underlying topics. In addition, topic modeling has been utilized in recommendation systems to generate personalized recommendations based on users' interests and preferences.

Topic modeling is a technique used in Natural Language Processing (NLP) to discover latent topics or themes present in a collection of documents. It allows us to automatically extract meaningful and coherent topics from unstructured text data. Python provides several libraries and tools that make topic modeling implementation straightforward.

1.2 METHODOLOGY

Here is an introduction to topic modeling in NLP using Python:

Preprocessing Text Data: Before performing topic modeling, it is essential to preprocess the text data. This typically involves steps like tokenization, removing stopwords, stemming or lemmatization, and handling other text-specific preprocessing tasks.

Document-Term Matrix: The first step in topic modeling is representing the text data in a numerical format. The Document-Term Matrix (DTM) or Term-Document Matrix (TDM) captures the frequency of words or terms in each document. Python libraries such as scikit-learn or gensim provide efficient methods to create the DTM from a corpus of documents.

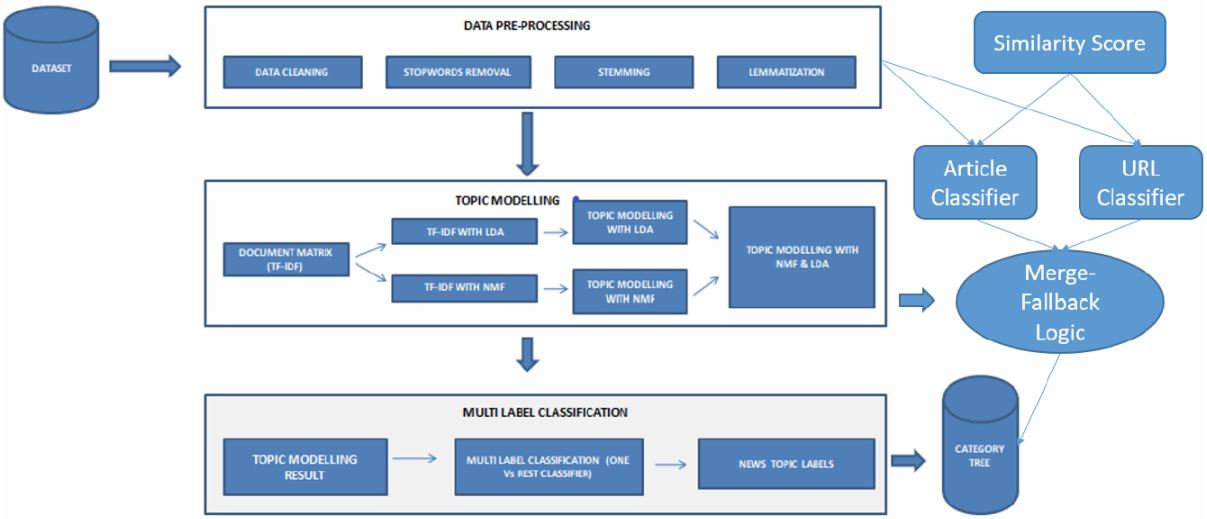

Fig 1.2 Methodology

Latent Dirichlet Allocation (LDA): LDA is one of the most popular and widely used algorithms for topic modeling. It is a generative probabilistic model that assumes each document is a mixture of a few topics, and each word in the document is generated from one of those topics. The goal of LDA is to discover the latent topics and their associated word distributions.

Implementation using Gensim: Gensim is a popular Python library for topic modeling. It provides an easy-to-use implementation of LDA and other topic modeling algorithms. The steps include creating a dictionary from the preprocessed text corpus, converting the corpus to a document-term matrix, training the LDA model using the corpus and dictionary, and extracting the topics and associated word distributions.

2. RELATED WORK

Base Paper : A Phrase Topic Model for Large-scale Corpus

DOI : 10.1109/ICCCBDA.2019.8725681

Topic Modeling is an unsupervised learning model, one of the important tools for large-scale corpus analysis, widely used in information retrieval, natural language processing, and machine learning.

This paper proposes a phrase topic model based on the LDA model, which integrates a regular expression constraint condition. This model makes the topic more meaningful and interpretable based on a limited increase in the dimensions of the vocabulary.

They perform part-of-speech tagging and adopt the noun phrase longest matching principle when extracting phrases at the document preprocessing stage. As in the example where stores become stores, they turn plural nouns into singular nouns.

Due to the use of the noun phrase longest matching principle, our algorithm may treat "modern machine learning method" as a single phrase. However, in order to prevent the lexical matrix from being excessively sparse, we remove the phrase that appears below a certain threshold number of times and discards phrases that exceed a certain threshold length by setting a threshold for the length of the regular expression. These assumptions are critical to the model reasoning.

For the realization of the LPTM, they integrated a new constraint, regular expressions, on top of LDA and established the relationship between topic and regular expression. Specifically, each topic has a distribution over regular expression vocabulary, just as a distribution topic over vocabulary in traditional LDA. Intuitively, each phrase m in the document d corresponds to a regular expression r, and the regular expression r is related to the topic again.

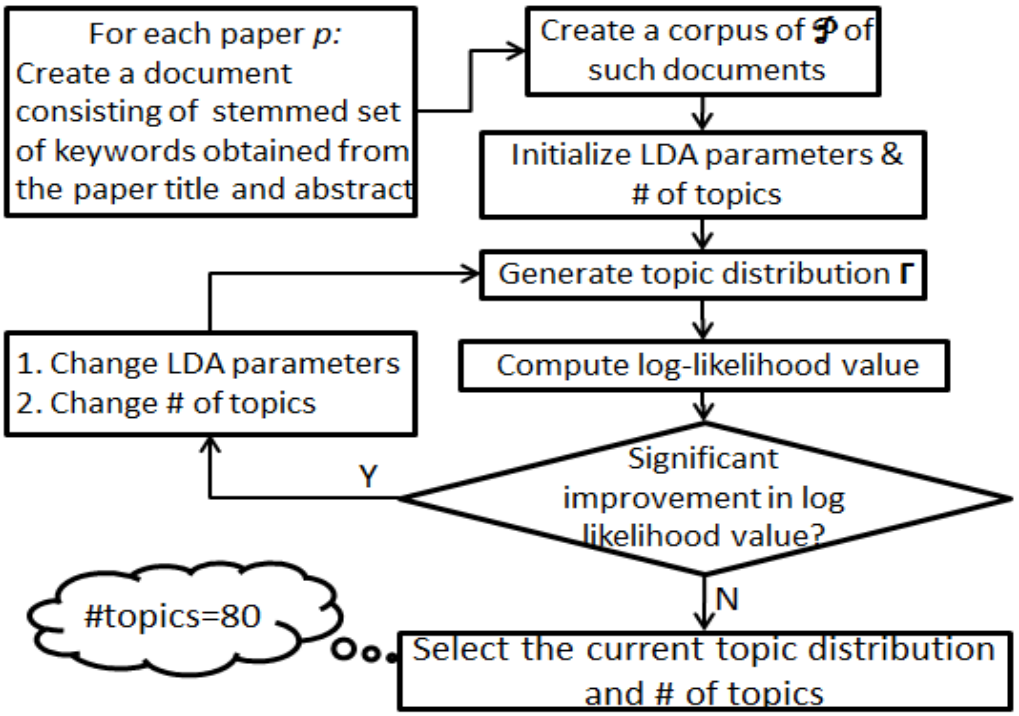

Fig:2.1 Flow chart of existing system

The topic model is an unsupervised learning model, one of the important tools for large-scale corpus analysis, widely used in information retrieval, natural language processing, and machine learning. Traditional topic models, such as Latent Dirichlet Allocation (LDA), ignore the order of words. However, in many text-mining tasks, word order and phrases are often crucial for capturing the meaning of texts efficiently.

2.1 ALGORITHMS USED

The paper proposes a phrase topic model based on the LDA model, which integrates a regular expression constraint condition. The model makes the topic more meaningful and interpretable based on a limited increase in the dimensions of the vocabulary. The experimental results show that the algorithm can find meaningful phrases and have generic applicability in our test data set.

In this paper, the proposed model is a new LDA-based phrase topic model (LPTM) regarding the problems above, which lies in between the aforementioned categories 1 and 3. In the LPTM, we consider the part-of-speech of a term and phrase structure.

They first adopted the noun phrase longest matching principle to extract noun phrases (if adjacent words are adjectives or nouns, they are combined to form a phrase), and then establish regular expression rules based on the parts-of-speech information of phrases. Then they improve the model by integrating a new regular expression constraint to the LDA. They deem that each phrase is guided by a regular expression and we enumerate the types of regular expressions that appear in the corpus.

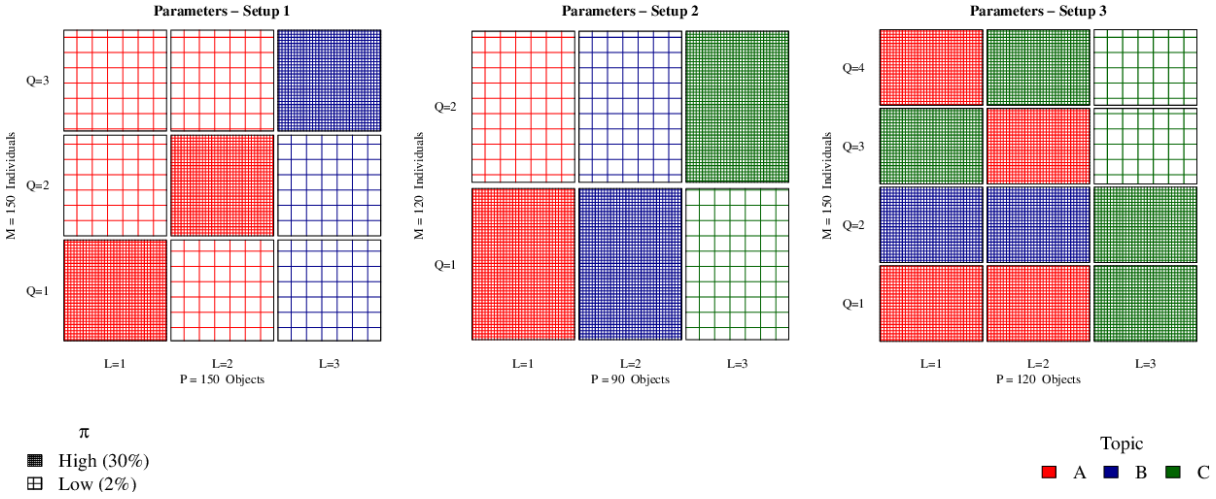

Fig:2.2 Latent Phrase Topic Block Model (LPTM)

They applied the LPTM and other baselines topic models on the NIPS corpus. Extensive experiments validate the effectiveness of our approach. The proposed approach outperforms the other topical phrase models in terms of perplexity and generates more interpretable topics.

The contributions of this paper mainly include the following points: we propose a generic topical phrase model whose use is not restricted by the discipline. They integrated new constraints-regular expressions based on the LDA topic model, based on consideration of the corpora's own characteristic information. Moreover, they still use the storage of unigram to generate vocabulary, since we introduced regular expressions.

3. METHODOLOGY

3.1 USE CASES

The proposed system aims to develop an effective topic modeling solution for NLP, capable of accurately identifying and extracting meaningful topics from text documents. The system will leverage advanced techniques from the field of natural language processing and machine learning to achieve its objectives.

The system will extract relevant features from the preprocessed text, such as word frequency, TF-IDF (Term Frequency-Inverse Document Frequency). These features will capture the semantic and contextual information necessary for topic modeling.

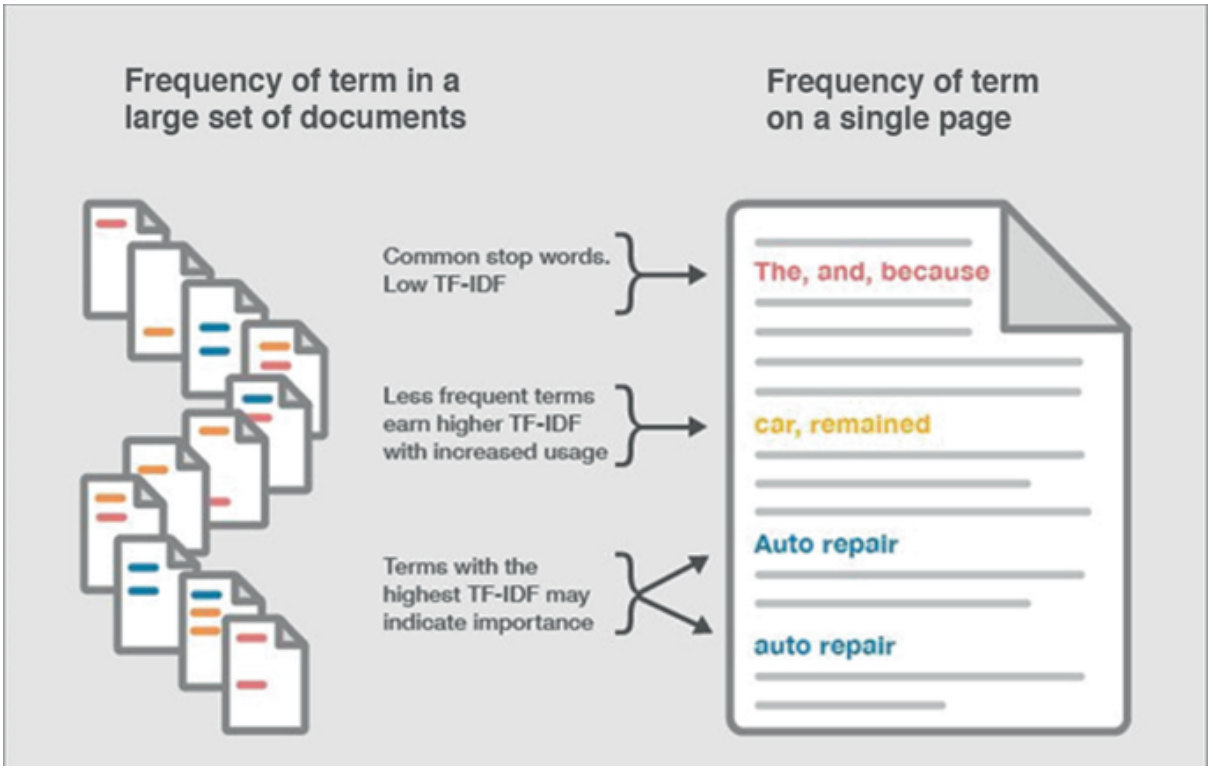

Fig:3.1 Term Frequency-Inverse Document Frequency (TF-IDF)

The system will employ a suitable topic modeling algorithm such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), or Hierarchical Dirichlet Process (HDP) based on the specific requirements and characteristics of the data.

3.2 ALGORITHMS

The proposed system includes the following algorithms:

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a widely used probabilistic topic modeling algorithm that was introduced by David Blei, Andrew Ng, and Michael Jordan in 2003. LDA assumes that each document in a collection is a mixture of different topics, and each topic is characterized by a distribution of words. The goal of LDA is to automatically discover these latent topics and their corresponding word distributions.

Implementations of LDA:

Gensim: Gensim is a Python library that provides efficient implementations of various topic modeling algorithms, including LDA. It offers a user-friendly interface for training LDA models on large text corpora, as well as tools for preprocessing text data and visualizing the resulting topics.

Scikit-learn: Scikit-learn is a popular machine learning library in Python that includes a module for topic modeling. Although Scikit-learn does not provide a direct implementation of LDA, it offers an implementation of Non-negative Matrix Factorization (NMF), which can be used for topic modeling tasks.

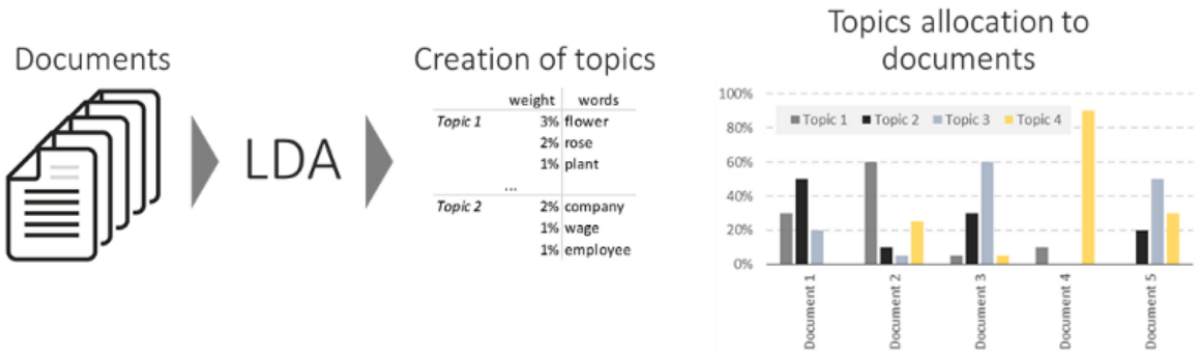

Fig: 3.2 Working of LDA

Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a technique used in Natural Language Processing (NLP) for analyzing relationships between a set of documents and the terms they contain. It is a mathematical approach that aims to uncover latent or hidden semantic structures within textual data. LSA is based on the idea that words that are used in similar contexts tend to have similar meanings.

Latent Semantic Analysis has several applications in NLP, including information retrieval, document clustering, text classification, and question-answering systems. By capturing latent semantic structures, LSA can help in understanding the meaning and relationships between words and documents, even in the presence of synonymy, polysemy, and other linguistic variations.

Key steps involved in Latent Semantic Analysis:

Document-Term Matrix Construction: The matrix elements typically denote the frequency of occurrence or some other measure of the term's presence in the document.

Singular Value Decomposition (SVD): SVD is applied to the document-term matrix to decompose it into three matrices: U, Σ, and V^T. U represents the left singular vectors, Σ is a diagonal matrix containing the singular values, and V^T represents the right singular vectors.

Dimensionality Reduction: The dimensionality of the matrix is reduced by truncating the matrices U, Σ, and V^T by keeping only the top k singular values and their corresponding vectors.

Similarity Calculation: The reduced matrices are used to calculate the similarity between documents and terms. Similarity measures such as cosine similarity are commonly used to quantify the relationships between the vectors. The higher the cosine similarity value, the more similar the documents or terms are considered to be.

Implementations of LSA:

Gensim: Gensim is a Python library that offers efficient implementations of LSA and other topic modeling algorithms. It provides an easy-to-use interface for training LSA models, transforming documents into their latent semantic representations, and performing similarity calculations.

Scikit-learn: Scikit-learn, a popular machine learning library in Python, includes a module for Latent Semantic Analysis. It provides functions for computing document-term matrices, performing SVD, and conducting dimensionality reduction.

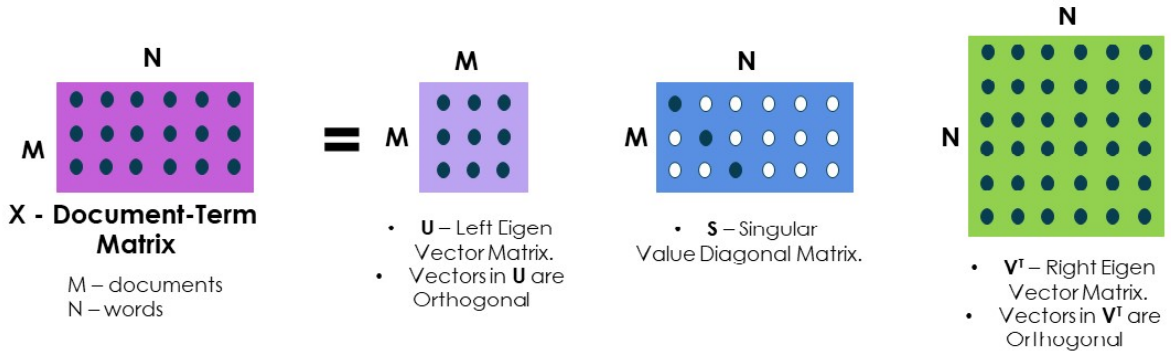

Fig: 3.3 Working of LSA

Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic Latent Semantic Analysis (PLSA) is a probabilistic generative model used for topic modeling and dimensionality reduction in Natural Language Processing (NLP). In PLSA, each document is represented as a mixture of latent topics, and each topic is characterized by a distribution over words. The model assumes that the observed word frequencies in a document are generated probabilistically based on these latent topics. The goal is to estimate the parameters of the model, including the topic-word distributions and the document-topic proportions, that maximize the likelihood of the observed data.

Implementations of PLSA:

Gensim: Gensim is a Python library that offers efficient implementations of various topic modeling algorithms, including PLSA. It provides an easy-to-use interface for training PLSA models, transforming documents into their latent topic representations, and performing similarity calculations.

Scikit-learn: Scikit-learn, a popular machine learning library in Python, includes a module for Latent Dirichlet Allocation (LDA), which is a related topic modeling algorithm. Although not specifically dedicated to PLSA, LDA can be seen as a variant of PLSA and can be used for similar tasks.

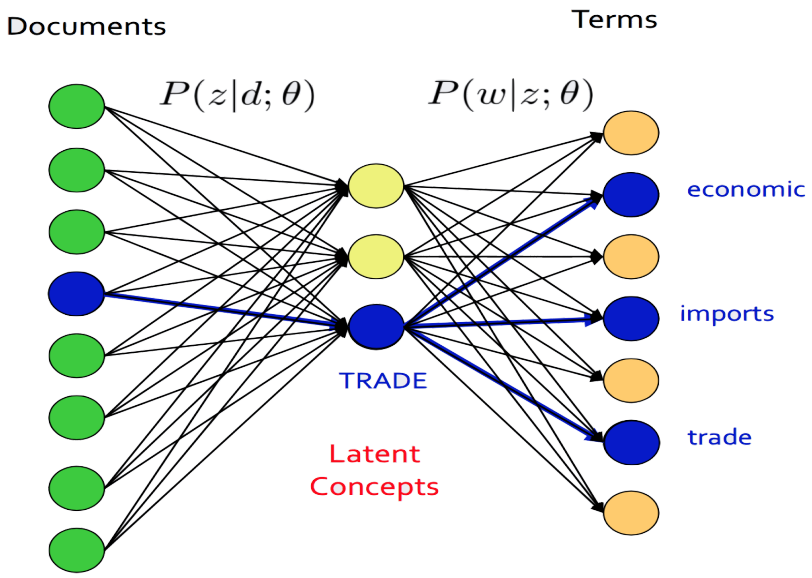

Fig: 3.4 Working of PLSA

4. IMPLEMENTATION

Steps involved in the implementation of the Algorithm:

Step 1: Data Preprocessing: Clean and preprocess the text data, including tasks like tokenization, removing stopwords, and stemming/lemmatization.

Step 2: Create a Document-Term Matrix: Convert the preprocessed documents into a numerical representation such as a bag-of-words or TF-IDF matrix.

Step 3: Initialize Model Parameters: Set the number of topics (K) to extract and initialize the topic-word and document-topic distributions.

Step 4: Gibbs Sampling: Iteratively update the topic assignments for each word in each document using Gibbs sampling, which involves sampling from the conditional probability distributions.

Step 5: Estimate Model Parameters: After convergence, estimate the final topic-word and document-topic distributions using the collected samples.

Step 6: Inference: Use the estimated model to infer the most probable topics for new, unseen documents.

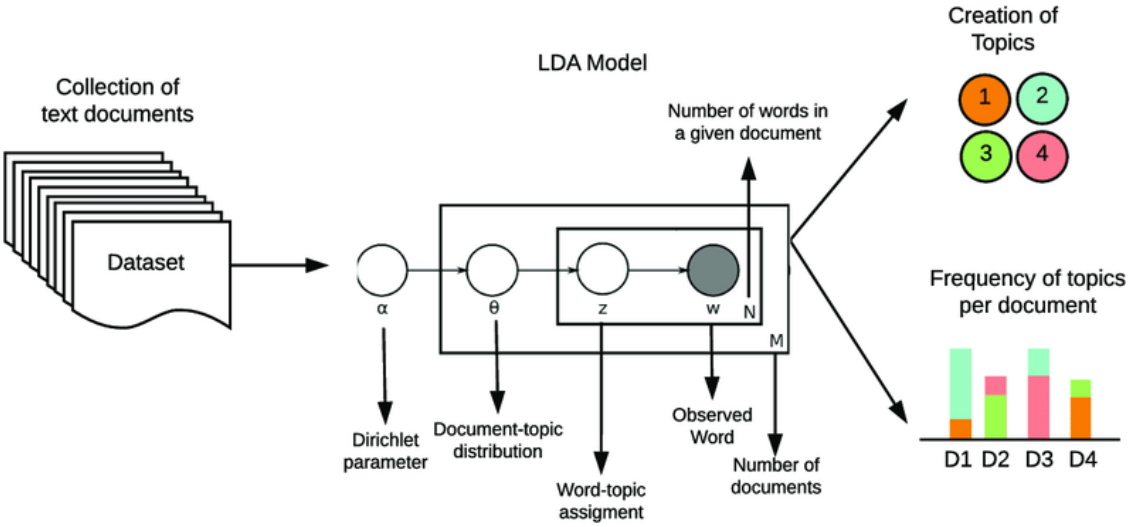

Fig:4.1 Working Model

Word tokenization and lemmatization are important preprocessing steps in the implementation of LDA (Latent Dirichlet Allocation) for topic modeling. Here's a brief explanation of these steps:

4.1 WORD TOKENIZATION

Word tokenization is the process of splitting a text into individual words or tokens. In the context of topic modeling with LDA, tokenization is performed on each document in the corpus. The goal is to break down the text into its constituent words, as LDA operates on the assumption that the input is a collection of documents composed of words.

For example, given the sentence "The cat is sitting on the mat," tokenization would produce the tokens: ["The", "cat", "is", "sitting", "on", "the", "mat"].

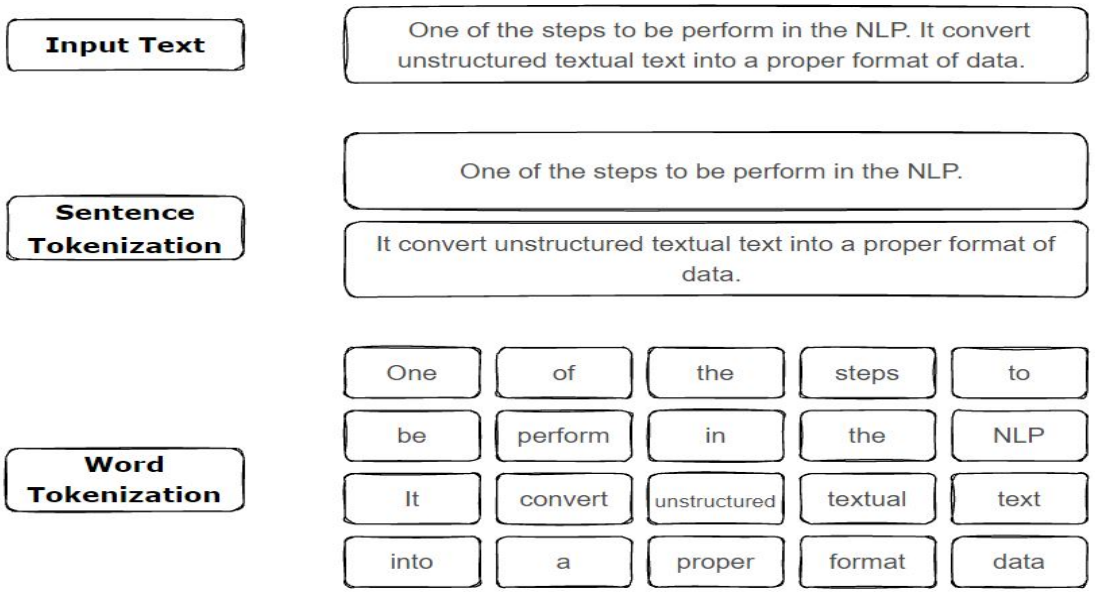

Fig:4.2 Tokenization

4.2 LEMMATIZATION

Lemmatization is the process of reducing words to their base or root form, known as the lemma. It aims to normalize different inflected forms of a word to a common base form. This step helps to consolidate words with similar meanings and reduces the dimensionality of the vocabulary.

For example, the words "running," "runs," and "ran" would be lemmatized to their base form "run".

Lemmatization is typically performed after tokenization. It involves applying linguistic rules to identify the lemma of each word. Libraries such as NLTK (Natural Language Toolkit) or spaCy provide lemmatization functionalities in various languages.

In the context of LDA, lemmatization can help in improving the quality of topics by reducing the sparsity and capturing the essence of words with the same lemma. It is often used in conjunction with other preprocessing steps like stop word removal and lowercasing.

By incorporating word tokenization and lemmatization in the preprocessing pipeline of LDA, you can effectively prepare your text data for topic modeling, enhancing the interpretability and coherence of the resulting topics.

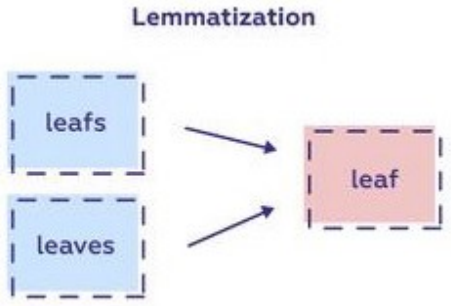

Fig:4.3 Lemmatization

4.3 TRAINING THE MODEL

To generate word clouds based on the topics extracted using LDA (Latent Dirichlet Allocation) in topic modeling, you can follow these steps:

1. Train an LDA model using your document-term matrix or corpus. Ensure that you have already preprocessed your text data, including tokenization, lemmatization, and removing stopwords.

2. Extract the most probable words for each topic in the LDA model. You can do this by accessing the topic-word distribution obtained from the trained model.

3. Calculate the relevance score for each word in a topic. This score can be based on the probability of the word in the topic or other metrics such as TF-IDF or coherence scores.

4. Sort the words in each topic based on their relevance scores in descending order.

5. Select the top N words from each topic to be included in the word cloud. The value of N depends on the desired number of words in the word cloud and can be adjusted based on the length and complexity of the topics.

6. Generate a word cloud for each topic using a word cloud visualization library like word cloud or matplotlib.

pyLDAvis is a powerful library that allows you to visually explore and interpret the results of an LDA (Latent Dirichlet Allocation) topic model. It provides an interactive visualization that enables you to understand the topics, their interrelationships, and the most relevant terms.

The main use of pyLDAvis is to generate an interactive visualization that consists of the following components:

Topic-Term Matrix: A bar chart that displays the most relevant terms for each topic. The width of the bars represents the term frequency within the topic.

Inter-Topic Distance Map: A 2D scatter plot that shows the proximity between topics. Topics with similar content are located closer to each other.

Word Clouds: Word clouds are displayed for each topic, showing the most important and distinctive terms in the topic.

Topic Selection: By clicking on a topic in the visualization, you can view the most relevant terms and their frequencies in the selected topic.

The pyLDAvis library helps in interpreting and exploring the LDA topic model by providing an intuitive and interactive interface. It aids in identifying and understanding the topics, analyzing their relationships, and discovering the key terms associated with each topic.

5. RESULTS

In [5]:
```
def lemmatize(docs, allowed_postags = ["NOUN","ADJ","VERB","ADV"]):
    #nlp = spacy.load("en_core_web_sm", disable = ["parser","ner"])
    nlp = en_core_web_md.load(disable=['parser', 'ner'])
    lemmatized_docs = []
    for doc in docs:
        doc = nlp(doc)
        tokens = []
        for token in doc:
            if token.pos_ in allowed_postags:
                tokens.append(token.lemma_)
        lemmatized_docs.append(" ".join(tokens))
    return (lemmatized_docs)
```

In [6]:
```
def tokenize(docs):
    tokenized_docs = []
    for doc in docs:
        tokens = gensim.utils.simple_preprocess(doc,deacc=True)
        tokenized_docs.append(tokens)
    return (tokenized_docs)
```

n [17]:
```
# Fetch 20newsgropus dataset
docs = fetch_20newsgroups(subset='all',remove=('headers','footers','quotes'))

# Pre-process input: lemmatization and tokenization
lemmatized_docs = lemmatize(docs)
tokenized_docs = tokenize(lemmatized_docs)

# Mapping from word IDs to words
id2word = corpora.Dictionary(tokenized_docs)

# Prepare Document-Term Matrix
corpus = []
for doc in tokenized_docs:
    corpus.append(id2word.doc2bow(doc))
```

Fig:5.1 Pseudo Code for LDA

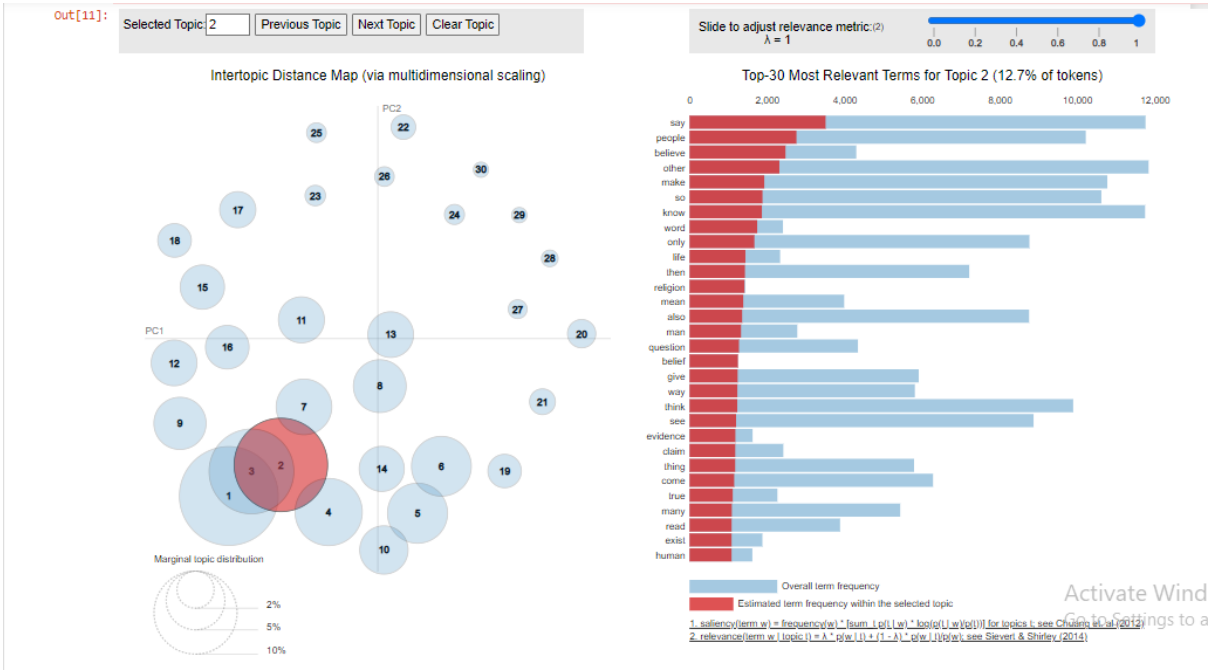

Fig:5.2 Visualization using PyLDAvis

```
In [18]: news_df = pd.DataFrame({'News': docs.data,'Target': docs.target})
         news_df['Target_name'] = news_df['Target'].apply(lambda x: docs.target_names[x])
```

```
In [23]: from wordcloud import WordCloud
         import matplotlib.pyplot as plt
         wordcloud = WordCloud(background_color='white',
                               max_words=200).generate(str(news_df['News']))
         fig = plt.figure(figsize=[7,10])
         plt.title('WordCloud of News')
         plt.axis('off')
         plt.imshow(wordcloud)
         plt.show()
```

Fig:5.3 Word Cloud Generation for LDA

```
In [17]:
from gensim.parsing.preprocessing import remove_stopwords, strip_punctuation, preprocess_string, strip_short, stem_text

# preprocess given text
def preprocess(text):
    CUSTOM_FILTERS = [lambda x: x.lower(),
                                remove_stopwords,
                                strip_punctuation,
                                strip_short,
                                stem_text]
    text = preprocess_string(text, CUSTOM_FILTERS)
    return text

df['reviewText'] = df['reviewText'].astype(str)
df['cleanText'] = df['reviewText'].apply(preprocess)
```

```
In [18]:
df.head()
```

Out[18]:

	reviewerID	reviewText	cleanText
0	A2IBPI20UZIR0U	Not much to write about here, but it does exac...	[write, here, exactli, suppos, filter, pop, so...
1	A14VAT5EAX3D9S	The product does exactly as it should and is q...	[product, exactli, afford, realiz, doubl, scre...
2	A195EZSQDW3E21	The primary job of this device is to block the...	[primari, job, devic, block, breath, produc, p...
3	A2C00NNG1ZQQG2	Nice windscreen protects my MXL mic and preven...	[nice, windscreen, protect, mxl, mic, prevent,...
4	A94QU4C90B1AX	This pop filter is great. It looks and perform...	[pop, filter, great, look, perform, like, stud...

Fig:5.4 Pseudo Code for LSA

```
In [28]:
df_topic0 = df_topic[df_topic['Topic']==0]
df_topic1 = df_topic[df_topic['Topic']==1]
print('Sample text from topic 0:\n {}'.format(df_topic0.sample(1, random_state=2)['Text'].values))
print('\nSample text from topic 1:\n {}'.format(df_topic1.sample(1, random_state=2)['Text'].values))
```

```
Sample text from topic 0:
 ["4 stars instead of 5 only because this thing won't tune your guitar or carry your amp. I use this solo, in a duo with a bass player, in a trio with drums and bass. I
have a regular gig at a medium sized club that is basically one large room. I use it through a Pyle Pro self contained Pa into which I also plug my Martin GRP elec/acous
tic. The sound quality is excellent. The volume seems limited only bny what the Pyle Pro will put out. I would have no hesitation using this for bigger rooms with a full
band and PA. I haven't heard a hint of feedback and the on/off switch is a plus.The 36.99 price might seem to good to be true but think of the things for which we are pa
ying less for than we used to. I bought a 32&#34; Toshiba flat screen for around 230 bucks, Think of what that would have cost even five years ago or less. The same is t
rue of a lot of Audio Gear. A lot of companies are bringing high quality elctronics and music to the market but it's great to have a company like Shure make something so
good and so affordable. They have always made a high quality product and now they have one that almost anyone can afford. I cannot recommend it strongly enough."]

Sample text from topic 1:
 ["While this is not a $300 tape delay...it's a solid pedal that gets it done. It's way better built than other pedals in this price range. I'm sold. I've purchased seve
ral JoYo pedals and all of them are built extremely well and perform as good as if not better than top name pedals."]
```

Fig:5.5 Sampling after Semantic Analysis

```
In [27]:
from wordcloud import WordCloud
import matplotlib.pyplot as plt
wordcloud = WordCloud(background_color='white',
                      max_words=200).generate(str(df_topic['Text']))
fig = plt.figure(figsize=[7,10])
plt.title('WordCloud of Musical Instruments')
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```

Fig:5.6 Word Cloud Generation for LSA

6. CONCLUSION

Through the use of topic modeling algorithms such as Latent Dirichlet Allocation (LDA), Probabilistic Latent Semantic Analysis (PLSA), and Hierarchical Dirichlet Process (HDP), researchers and practitioners have been able to extract meaningful topics from large text corpora automatically. These topics provide insights into the underlying themes and concepts present in the documents, facilitating the organization, exploration, and analysis of textual data.

One of the significant advantages of topic modeling is its ability to handle high-dimensional and unstructured text data. By reducing the dimensionality of the data and representing it in terms of latent topics, topic modeling helps overcome the challenges posed by the vast amount of textual information available today.

Moreover, topic modeling techniques have evolved to incorporate additional sources of information, such as incorporating external knowledge and integrating multi-modal data (e.g., images, audio, video) with text. These advancements have further enhanced the performance and interpretability of topic models, allowing for richer and more comprehensive analysis of diverse types of data.

Here are some key takeaways regarding topic modeling with LDA:

Unsupervised Approach: LDA is an unsupervised learning algorithm, meaning it discovers topics without the need for labeled data. It automatically identifies topics based on the statistical patterns of word co-occurrences in the documents.

Probabilistic Model: LDA is a probabilistic model that assigns probabilities to each word being generated from a particular topic and each document containing a combination of multiple topics. This probabilistic framework provides flexibility and robustness in capturing the underlying structure of the data.

Interpretability: LDA produces interpretable results by associating each topic with a distribution of words. By analyzing the topic-word distributions, we can understand the most significant terms that define each topic and gain insights into the main themes present in the documents.

Applications: Topic modeling using LDA has various applications in NLP, including text mining, document clustering, information retrieval, sentiment analysis, and recommendation systems. It helps in organizing and summarizing large document collections, facilitating efficient information retrieval and knowledge discovery.

Python Libraries: Python provides several libraries, such as Gensim, scikit-learn, and NLTK, that offer easy-to-use implementations of LDA and other topic modeling algorithms. These libraries provide functionalities for creating document-term matrices, training topic models, extracting topics, and visualizing the results.

7. REFERENCES

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.

Griffiths, T. L., & Steyvers, M. (2004). Finding Scientific Topics. Proceedings of the National Academy of Sciences, 101(Supplement 1), 5228-5235.

Mei, Q., Shen, X., & Zhai, C. (2007). Automatic Labeling of Multinomial Topic Models. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 490-499.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic Evaluation of Topic Coherence. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 100-108.

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 248-256.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing Semantic Coherence in Topic Models. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 262-272.

Wang, C., Blei, D. M., & Heckerman, D. (2009). Continuous-Time Dynamic Topic Models. Proceedings of the 26th International Conference on Machine Learning, 1121-1128.

Sievert, C., & Shirley, K. E. (2014). LDAvis: A Method for Visualizing and Interpreting Topics. Proceedings of the International Conference on the Advances in Intelligent Data Analysis, 391-405.

Chen, Z., Mukherjee, A., Liu, B., & Hsu, M. (2015). Discovering Coherent Topics Using General Knowledge. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1352-1362.

Nguyen, D. Q., Boyd-Graber, J., Resnik, P., & Phillips, A. T. (2015). Tea Party in the House: A Hierarchical Ideal Point Topic Model and Its Application to Republican Legislators in the 112th Congress. The Journal of Artificial Intelligence Research, 53, 315-354.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic Evaluation of Topic Coherence. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 100-108.

Sievert, C., & Shirley, K. E. (2014). LDAvis: A Method for Visualizing and Interpreting Topics. Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 63-70.

Chang, J., Boyd-Graber, J. L., Gerrish, S., Wang, C., & Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. Neural Information Processing Systems, 288-296.

Blei, D. M., Lafferty, J. D., & Smyth, P. (2007). Supervised Topic Models. Advances in Neural Information Processing Systems, 601-608.

Roberts, M. E., Stewart, B. M., & Tingley, D. (2016). stm: An R Package for Structural Topic Models. Journal of Statistical Software, 1-40.

F. Esposito, A. Corazza, F. Cutugno, Topic Modelling with Word Embeddings, in IEEE Transactions on Content Mining, Vol.7, April 2018.

Adji B. Dieng, Francisco J. R. Ruiz, David M. Blei, "Topic Modeling in Embedding Spaces," in ACL Information Retrieval (cs.IR); Computation and Language (cs.CL); Machine Learning (cs.LG); Machine Learning (stat.ML), Vol. 6, 8 Jul 2019.

Lingyun Li; Yawei Sun; Cong Wang,Semantic Augmented Topic Model over Short Text., 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS).Nov. 2018.

C.J. Hutto, Eric Gilbert,VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text, in IEEE 8th Eighth International Conference on Weblogs, Nov2014.

Bhagyashree Vyankatrao Barde, Anant Madhavrao Bainwad,An overview of topic modeling methods and tools, in IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), Jan 2018.

P. Anupriya, S. Karpagavalli,LDA based topic modeling of journal abstracts, in IEEE International Conference on Advanced Computing and Communication Systems, Nov 2015.

Dandan Song; Jingwen Gao, Jinhui Pang, Lejian Liao, Lifei Qin, Knowledge Base Enhanced Topic Modeling,in IEEE International Conference on Knowledge Graph (ICKG), Sept 2020.

Yang Gao, Yue Xu; Yuefeng Li, Pattern-Based Topic Models for Information Filtering,in IEEE 13th International Conference on Data Mining Workshops,March 2014.

Biao Wang, Yang Liu, Zelong Liu, Maozhen Li, Man Qi,Topic selection in latent dirichlet allocation,in 11th Internatinal Conference on Fuzzy Systems and Knowledge Discovery (FSKD),Dec 2014.

Zhenzhong Li, Wenqian Shang, Menghan Yan ,News text classification model based on topic model,in IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),June 2016.

David Alfred Ostrowski,Using latent dirichlet allocation for topic modelling in twitter,in Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015),March 2015.