**University trier**

**12401603|Digital Humanities |sose 2024**

**Prof.Dr.schoch**

**Topic:Analyzing And Topic Modeling for Good-Read Quotes**

**Muhammad Ahmed**

**1678814**

[**s4muahme@uni-trier.de**](mailto:s4muahme@uni-trier.de)

**Masters NLP**

Abstract:

This research explores the application of Latent Dirichlet Allocation (LDA) and advanced topic modeling techniques to analyze thematic patterns and topics within large collections of textual data, specifically focusing on short-text corpora. Given the inherent sparsity of short texts, traditional models like LDA struggle with topic coherence due to the limited context. This study leverages pseudo-document aggregation to overcome these challenges, building on recent advancements in short-text topic modeling to enhance interpretability and coherence of the topics generated. Through a comprehensive analysis of thematic patterns, key topics are identified, providing insights into the distribution of themes across various datasets. Furthermore, this study evaluates the effectiveness of LDA-based topic modeling in capturing both positive and negative sentiments, common themes, and thematic diversity across topics. The findings demonstrate the model's capacity to handle the sparsity of short texts, offering significant improvements in topic coherence and diversity. This research contributes to the growing body of work on short-text topic modeling, highlighting its potential applications in areas such as social media analysis, content categorization, and information retrieval.

**I hereby clarify that I wrote this work independently and did not use any resource other than those specified anything taken from external sources is identified I agree to this electronic plagiarism check for this work.**

**Muhammad Ahmed**

**15-10-2024**

Table of Contents

[1.1 Overview of Goodreads and Literary Quotes: 4](#_Toc179829154)

[1.2 Importance of Analyzing User-Generated Content: 4](#_Toc179829155)

[1.3 Literature Review 4](#_Toc179829156)

[2. Data 6](#_Toc179829157)

[2.1 Data Source Description: 6](#_Toc179829158)

[2.2 Structure of the Dataset: 6](#_Toc179829159)

[3. Tools and Methodology 7](#_Toc179829160)

[3.2 Data Visualization Tools: 7](#_Toc179829161)

[4. Topic Modeling 9](#_Toc179829162)

[4.1 Latent Dirichlet Allocation (LDA) Methodology 9](#_Toc179829163)

[4.2 Corpus Creation and Preprocessing 9](#_Toc179829164)

[4.3 Topic Extraction Using LDA 10](#_Toc179829165)

[. 12](#_Toc179829166)

[4.4 Results: Topics Extraction 12](#_Toc179829167)

[4.5 Coherence Assessment 13](#_Toc179829168)

[4.6 Top 10 Most Liked Quotes: 19](#_Toc179829169)

[4.7 Top 10 Most Liked Categories: 19](#_Toc179829170)

[4.8 Top 10 Most Liked Authors: 20](#_Toc179829171)

[5 . Results 21](#_Toc179829172)

[5.1 Topic Distribution and Visualization 21](#_Toc179829173)

[5.2 Most Common Themes or Genres in Quotes 23](#_Toc179829174)

[5.3 Most Negative or Critical Themes in Quotes 26](#_Toc179829175)

[5.4 Most Positive or Uplifting Themes in Quotes 27](#_Toc179829176)

[5.5 Most Common Themes Across Topics 28](#_Toc179829177)

[5.6 Diversity of Themes by Topic 29](#_Toc179829178)

[6. Discussion 31](#_Toc179829179)

[6.1 Insights from Keyword and Trend Analysis 31](#_Toc179829180)

[6.2 Influence of Authors and Themes on Goodreads 31](#_Toc179829181)

[7 Conclusion 33](#_Toc179829182)

[7.1 Summary of Findings 33](#_Toc179829183)

[7.2 Potential Applications of the Research 33](#_Toc179829184)

[8 . Reference 34](#_Toc179829185)

**1. Introduction**

* 1. **Overview of Goodreads and Literary Quotes:**

Goodreads is one of the largest social platforms dedicated to book enthusiasts, where users can share, discover, and engage with quotes from books they’ve read. These literary quotes often reflect meaningful moments from novels, poetry, and non-fiction works. Goodreads allows users to upvote quotes they find inspiring or thought-provoking, creating a rich repository of content that spans across different genres, authors, and themes. The quotes are tagged by users with relevant keywords, further helping to categorize and identify the core messages of the shared content.

## **1.2 Importance of Analyzing User-Generated Content:**

User-generated content, such as the quotes found on Goodreads, holds significant value for researchers in the digital humanities. This content provides a snapshot of the collective consciousness of readers—showing what themes, topics, and authors resonate most with the public. By analyzing this dataset, one can uncover patterns in literary appreciation, revealing what types of quotes tend to gather the most attention and which themes dominate user interactions. Studying these trends offers insight into both contemporary cultural values and timeless literary sentiments. Moreover, such analyses help bridge the gap between digital content and traditional literary scholarship, offering new ways to study the evolution of reader preferences and engagement in a digital age.

## **1.3 Literature Review**

Recent advancements in topic modeling have significantly enhanced the ability to analyze large collections of textual data, particularly in short-text domains. The foundational work by **Blei, Ng, and Jordan (2003)** introduced **Latent Dirichlet Allocation (LDA)**, a probabilistic model that revolutionized how researchers infer latent topics from collections of discrete data such as text corpora. By representing documents as mixtures of topics and each topic as a mixture of words, LDA has become a cornerstone in text modeling, offering an interpretable framework for organizing and understanding large textual datasets. However, LDA was initially designed for longer documents, and its performance degrades when applied to short texts like tweets or news headlines, where the limited context often results in poor topic coherence.

Addressing these limitations, **Hong and Davison (2010)** proposed aggregating short texts into pseudo-documents to create more meaningful co-occurrence structures, thus improving the topic modeling process for sparse data. Building on this concept, **Zuo et al. (2016)** introduced the **Pseudo-Document-based Topic Model (PTM)**, which aggregates short texts into pseudo-documents to enhance topic extraction and improve coherence. Their model demonstrated superior performance in handling sparse short-text data, marking a notable advancement in the field.

In addition to probabilistic models, recent surveys, such as **Xun et al. (2021)**, have highlighted the emergence of **neural topic models** as a promising approach to overcome the limitations of traditional models like LDA. Neural models, which leverage pre-trained language models, provide more robust representations for complex text data. These advancements have shifted the focus towards developing scalable, flexible models that can handle diverse datasets more effectively.

In the field of bioinformatics, topic models have also found novel applications. **Kho et al. (2017)** employed LDA to classify gene expression data, focusing on cancer research. Their approach used LDA for clustering and classification of cancerous and healthy tissues, achieving high accuracy in distinguishing between the two. This demonstrates LDA’s utility beyond traditional text analysis, showcasing its potential in genomic research by identifying key genetic markers and pathways involved in cancer differentiation.

Further extending the application of LDA, **Riedl and Biemann (2012)** introduced **TopicTiling**, an LDA-based text segmentation algorithm that stabilizes topic assignments and improves segmentation accuracy. Their algorithm, which is computationally efficient, outperformed previous segmentation methods and demonstrated the applicability of LDA in document segmentation tasks. Similarly, **Scaiano et al. (2010)** explored text segmentation for movie subtitles using an adapted version of the **TextTiling** algorithm, showing the utility of text segmentation for improving information retrieval from films.

The broader utility of topic modeling and its integration with human expertise was explored by **Chuang et al. (2015)**, who examined how interactive topic models could supplement traditional content analysis methods. They emphasized the need for reproducibility and reliability in computer-assisted content analysis, proposing interactive tools that allow users to explore and refine machine-generated topics. Their work highlighted the importance of a human-in-the-loop approach in ensuring the interpretability of computational analyses.

Additionally, **Potrus et al. (2014)** introduced a novel hybrid method for Arabic text recognition using a combination of **Genetic Algorithms (GA)** and **Harmony Search (HS)** algorithms. Their method achieved high recognition accuracy for Arabic text, demonstrating the versatility of topic modeling and related techniques across different languages and character sets.

Finally, **Barak, Floyd, and Goldberg (2019)** extended a feature-based computational model to address the complexities of polysemy and homonymy in word learning. Their model, which distinguishes between multiple meanings of a word, aligns better with human performance in word-learning tasks, emphasizing the importance of handling word ambiguity in natural language models.

In conclusion, the evolution of topic modeling from probabilistic models like LDA to modern neural techniques reflects ongoing efforts to tackle challenges related to text data sparsity, model interpretability, and scalability. These developments have enabled topic modeling to extend its reach across disciplines, offering powerful tools for both traditional text analysis and more specialized applications like bioinformatics and content analysis.

# 

# **2. Data**

## **2.1 Data Source Description:**

The dataset utilized for this research was obtained from the publicly available Kaggle dataset titled "Goodreads Quotes" (available at: [Goodreads Quotes Dataset](https://www.kaggle.com/datasets/abhishekvermasg1/goodreads-quotes/data)). This dataset was sourced from Goodreads, a widely known platform where millions of users share, engage with, and discuss quotes from various literary works. Goodreads has become a significant hub for literary appreciation, with user-generated content reflecting the diverse interests and sentiments of its global user base. The dataset selected for this research focuses on quotes that have gathered substantial attention and engagement, indicated by user interactions like likes or upvotes. As such, it provides a rich source for examining prevalent themes, notable authors, and the impact of particular quotes within the Goodreads community.

* 1. **Structure of the Dataset:**

The dataset consists of five key columns that offer essential information about the quotes and their context:

* 1. **Index:** A unique identifier assigned to each quote entry, which helps differentiate and reference individual records in the dataset.
  2. **Quote:** The primary textual content, containing the quote itself. These quotes vary in length and cover a wide range of literary styles, genres, and sentiments.
  3. **Author:** The name of the individual or author who is attributed to the quote, spanning both classical and contemporary literary figures.
  4. **Tags:** User-generated keywords or themes associated with each quote. Tags are separated by semicolons and help categorize the quotes based on central ideas or emotions (e.g., "love," "inspiration," "wisdom").
  5. **Likes:** The number of likes or upvotes that each quote has received from the Goodreads community. This metric serves as an indicator of the quote's popularity and resonance with readers.

# **3. Tools and Methodology**

* 1. **Data Analysis Tools:**  
     The analysis of the dataset will be conducted using a combination of Python libraries that are well-suited for handling, processing, and analyzing text-based data. The primary tools include:
  2. **Pandas:** A highly efficient data manipulation library that will be used for importing, cleaning, and structuring the dataset into a workable format. It is essential for managing the data and performing initial explorations.
  3. **NumPy:** This library will assist in numerical operations and data handling, especially for any calculations required during the analysis.
  4. **Gensim:** For topic modeling, **Gensim** will be employed to implement **Latent Dirichlet Allocation (LDA)**. This method helps in extracting and identifying hidden themes or topics across the dataset, which will provide insights into the central ideas and sentiments in the quotes.
  5. **BERTopic:** In addition to LDA, **BERTopic** will be used as an alternative modeling approach. BERTopic leverages pre-trained BERT embeddings to identify clusters of semantically related words, which can work effectively even with smaller datasets, such as the one in this study.
  6. **CoherenceModel (Gensim):** This model will be used to assess the coherence of the topics generated during the topic modeling process. Higher coherence scores will indicate more meaningful and interpretable topics.

## **3.2 Data Visualization Tools:**

Visualization is a crucial part of understanding the data and presenting results. To create meaningful visual representations of the insights, the following libraries will be used:

* 1. **Matplotlib:** This is a comprehensive plotting library used to generate a variety of static, animated, and interactive plots. It will be used to produce bar charts, line graphs, and other plots that visually summarize the dataset and topic modeling results.
  2. **Seaborn:** Built on top of Matplotlib, **Seaborn** offers enhanced data visualization capabilities, particularly for statistical plots. It will be used to create attractive and informative plots, making it easier to understand patterns, distributions, and relationships in the data.
  3. **WordCloud:** This library will be used to generate **word clouds** for both the quotes and tags columns, visually representing the frequency of words by displaying commonly used words in a larger and bolder font. Word clouds help quickly grasp the dominant themes in the text.
  4. **pyLDAvis:** This tool is specifically designed for visualizing topics generated by the LDA model. It provides interactive visualizations that show the distribution of topics and their relationships, helping in the interpretation of the model's output and making the results more accessible.

# 

# **4. Topic Modeling**

## **4.1 Latent Dirichlet Allocation (LDA) Methodology**

For this study, we employed **Latent Dirichlet Allocation (LDA)**, a widely-used probabilistic model that discovers latent semantic structures within large collections of text. LDA treats each document as a distribution over a fixed number of topics, and each topic as a distribution over words. In our case, the dataset comprised thousands of quotes, with the objective of uncovering meaningful patterns and themes embedded within the text.

LDA was chosen for its ability to model complex structures in the data by assuming that documents exhibit multiple topics in varying proportions. The model helps uncover hidden themes by assigning a probabilistic distribution over topics for each document and over words for each topic. This flexibility makes LDA a robust choice for analyzing unstructured text such as quotes, where the diversity of themes can be high.

## **4.2 Corpus Creation and Preprocessing**

Preprocessing is a crucial step in any topic modeling task to ensure that the text data is clean and suitable for analysis. The corpus for this study was generated by applying the following preprocessing steps:

1. **Tokenization**: Each quote was broken down into individual words (tokens) to analyze its content at the word level.
2. **Lemmatization**: Words were reduced to their base or root form (lemmas), allowing for a more generalized analysis. For example, the words “running” and “ran” were converted to “run.”
3. **Stop Word Removal**: Commonly used words such as "the," "is," and "in" that add little semantic value were removed to focus on meaningful content.
4. **Punctuation Removal**: Non-alphabetical characters were removed, ensuring that punctuation marks did not interfere with the analysis.
5. **Bigrams and Trigrams**: To capture more meaningful phrases, bigrams (two-word combinations like "hard work") and trigrams (three-word combinations like "fall in love") were generated. This allowed us to retain context that would have been lost in single-word tokenization.

After preprocessing, the dataset consisted of a **bag-of-words representation**, where each document (quote) was represented by the frequency of its terms, and the terms were filtered to include only those appearing frequently enough to provide meaningful insights. Additionally, terms that occurred in too many documents (i.e., extremely common words) were excluded to reduce noise.

## **4.3 Topic Extraction Using LDA**

For topic extraction, the processed dataset was fed into the LDA model using **Gensim's LdaMulticore**, a parallelized implementation that allows efficient computation on multi-core systems. This was critical given the large size of the dataset. The following hyperparameters were used:

* **Number of Topics**: The model was configured to extract five topics, representing distinct semantic themes within the corpus. This number was selected based on initial tests and the goal of balancing interpretability with model complexity.
* **Iterations**: To fine-tune the word distributions across topics, the model was run for **62 iterations**, ensuring the convergence of topic-word probabilities.
* **Passes**: A single pass over the corpus was performed, which is typical when using LdaMulticore for efficient training.
* **Chunk Size**: The data was processed in chunks of 62 documents to optimize memory usage while maintaining computational efficiency.

The model's output consists of a distribution of topics across documents and the most prominent words associated with each topic. These topics represent distinct clusters of quotes that share common themes. For instance, one topic might predominantly feature terms related to "success" and "hard work," while another might focus on themes of "love" and "relationships."

We applied the LDA model using the Gensim library, which is an unsupervised machine learning technique designed to identify underlying topics in large corpora. The LDA algorithm assumes that each document in the dataset is a mixture of various topics and each topic is a mixture of words. In this way, the algorithm assigns a probability distribution to each word in the document for each topic.

The quotes dataset was preprocessed through the following steps:

1. **Tokenization and Lemmatization**: The text data was tokenized, and lemmatization was applied to reduce the words to their base forms. Stop words and punctuation were removed during this process.
2. **N-grams Creation**: Bigrams were created to capture meaningful word pairings within the dataset, such as “hard work” or “United States,” which could better represent the semantics of the content.
3. **Dictionary and Corpus Creation**: A dictionary was created to map unique tokens in the dataset, followed by a bag-of-words representation of each document. This corpus was then fed into the LDA algorithm

import pandas as pd

import spacy

from gensim import corpora, models

from gensim.models import CoherenceModel

import pyLDAvis.gensim\_models

import matplotlib.pyplot as plt

from nltk import ngrams

import multiprocessing

# Load the English NLP model

nlp = spacy.load('en\_core\_web\_sm')

# Load your data

data\_path = '/Users/muhammadahmed/Desktop/DM/quotes.csv'

df = pd.read\_csv(data\_path)

# Check for missing values and handle them

df['quote'] = df['quote'].fillna('') # Replace NaN with empty string

# Assuming your quotes are in a column named 'quote'

documents = df['quote'].tolist()

# Function to create bigrams or trigrams

def create\_ngrams(tokens, n=2):

return list(ngrams(tokens, n))

# Preprocess the text

def preprocess(text):

if isinstance(text, str) and text: # Ensure it's a non-empty string

doc = nlp(text)

tokens = [token.lemma\_ for token in doc if not token.is\_stop and not token.is\_punct]

bigrams = create\_ngrams(tokens, n=2) # You can change this to 3 for trigrams

return [' '.join(bigram) for bigram in bigrams] # Join bigrams into strings

return [] # Return empty list for non-string inputs

# Preprocess documents in parallel

def parallel\_preprocess(documents):

with multiprocessing.Pool(processes=multiprocessing.cpu\_count()) as pool:

processed\_docs = pool.map(preprocess, documents)

return processed\_docs

if \_\_name\_\_ == '\_\_main\_\_':

# Process documents

processed\_docs = parallel\_preprocess(documents)

# Create a dictionary representation of the documents

dictionary = corpora.Dictionary(processed\_docs)

# Filter extremes to limit vocabulary (further reduced by 50%)

dictionary.filter\_extremes(no\_below=1, no\_above=0.015625) # Adjusted values

# Create a corpus

corpus = [dictionary.doc2bow(text) for text in processed\_docs if text] # Ensure non-empty texts

# Use LdaMulticore for better performance on multi-core machines

num\_topics = 5 # Keep the same number of topics

chunksize = 62 # Reduced from 125 to 62

passes = 1 # Remain at 1

iterations = 2 # Reduced from 4 to 2

# Perform LDA using LdaMulticore

lda\_model = models.LdaMulticore(corpus, num\_topics=num\_topics, id2word=dictionary,

passes=passes, iterations=iterations,

chunksize=chunksize, workers=multiprocessing.cpu\_count())

# Save the LDA model

lda\_model.save('/Users/muhammadahmed/Desktop/DM/lda\_model.gensim')

# Print the topics found by LDA

topics = lda\_model.print\_topics(num\_words=4)

for topic in topics:

print(topic)

# Coherence Score

coherence\_model\_lda = CoherenceModel(model=lda\_model, texts=processed\_docs, dictionary=dictionary, coherence='c\_v')

coherence\_lda = coherence\_model\_lda.get\_coherence()

print(f'Coherence Score: {coherence\_lda}')

# Visualizing the topics and saving as HTML

vis = pyLDAvis.gensim\_models.prepare(lda\_model, corpus, dictionary)

pyLDAvis.save\_html(vis, '/Users/muhammadahmed/Desktop/DM/lda\_visualization.html')

## .

## **4.4 Results: Topics Extraction**

After training the LDA model, five distinct topics were extracted from the dataset. Each topic is represented by a combination of top words with their respective probabilities. Below are the results:

1. **Topic 0**: This topic centers around themes of success and personal growth, with terms like *"good thing," "hard work,"* and *"fall love."* It suggests a focus on striving for success through effort and personal relationships.
2. **Topic 1**: This topic is characterized by a broader theme of life experiences, with terms like *"live life," "year old,"* and *"good bad."* It captures a mix of reflections on life’s ups and downs, as well as the passage of time.
3. **Topic 2**: Health and societal themes dominate this topic, with terms like *"look like," "know know,"* and *"health care."* It points toward reflections on health, well-being, and understanding one’s surroundings.
4. **Topic 3**: A strong sense of patriotism and perseverance is evident in this topic, with terms like *"feel like," "United States,"* and *"work hard."* This suggests a focus on national identity and individual effort in overcoming challenges.
5. **Topic 4**: This topic revolves around themes of love, relationships, and personal growth, with terms such as *"human being," "love love,"* and *"young people."* The topic emphasizes interpersonal connections and the experience of youth.

## **4.5 Coherence Assessment**

**1. Definition of Coherence in Topic Modeling**  
Coherence in topic modeling refers to the degree to which the words in a topic make sense when presented together. A high coherence score indicates that the words frequently co-occur in the corpus and are semantically related, making the topic more interpretable. In contrast, a low coherence score suggests that the words may not be strongly connected, resulting in less meaningful topics.

Coherence is a crucial metric in evaluating topic models like Latent Dirichlet Allocation (LDA) because, while LDA attempts to find statistical patterns in the data, it doesn't inherently ensure that the discovered topics are humanly interpretable. The coherence score serves as a quantitative measure to assess the usefulness and quality of the topics generated.

**2. Coherence Measure Used**  
In this research, I utilized the **c\_v coherence measure**, which is one of the most widely used metrics in topic modeling. The c\_v coherence score evaluates the semantic similarity between high-scoring words in each topic by considering word co-occurrences in a sliding window across the corpus. It combines co-occurrence statistics with indirect confirmation measures, which leads to more reliable and interpretable results.

**3. Calculation of the Coherence Score**  
The coherence score was computed using **Gensim's CoherenceModel**, which allows for an efficient and accurate calculation of topic coherence. The steps involved in computing the coherence score include:

* Extracting the top words for each topic.
* Evaluating how frequently these words appear together in the documents.
* Assigning a score based on the extent to which these words co-occur and are semantically related.

**4. Coherence Score Results**  
The LDA model generated for this research achieved a **coherence score of 0.67**. This score reflects the overall interpretability and meaningfulness of the topics extracted from the corpus of quotes. A coherence score typically ranges from 0 to 1, where:

* Scores closer to 1 indicate highly coherent and easily interpretable topics.
* Scores closer to 0 suggest that the topics are less coherent, and the words within a topic may not be as strongly related.

In this context, a score of 0.67 suggests that the topics discovered by the LDA model are fairly coherent and interpretable. The extracted topics exhibit a reasonable level of semantic consistency, meaning that the words grouped within each topic tend to co-occur in a meaningful way. However, there is still some room for improvement in terms of further refining the topics to achieve a higher coherence score.

**5. Visualization of Coherence Score**

To visualize the coherence score, a bar chart was generated (Figure X) with the coherence score plotted on the y-axis. The plot clearly displays the LDA model's coherence score of 0.67. This graphical representation helps in communicating the model’s performance and provides a straightforward visual summary of the quality of the topics.

import pandas as pd

import spacy

from gensim import corpora, models

from gensim.models import CoherenceModel

import matplotlib.pyplot as plt

import multiprocessing

import os

# Load the Spacy model and disable unnecessary pipeline components

nlp = spacy.load('en\_core\_web\_sm', disable=['ner', 'parser'])

# Load your processed documents

data\_path = '/Users/muhammadahmed/Desktop/DM/quotes.csv'

df = pd.read\_csv(data\_path)

documents = df['quote'].fillna('').tolist() # Ensure non-empty strings

# Preprocess the text in parallel

def preprocess(text):

if isinstance(text, str) and text: # Ensure it's a non-empty string

doc = nlp(text)

tokens = [token.lemma\_ for token in doc if not token.is\_stop and not token.is\_punct]

return tokens # Return tokens directly

return [] # Return empty list for non-string inputs

def parallel\_preprocess(docs):

with multiprocessing.Pool(processes=multiprocessing.cpu\_count()) as pool:

return pool.map(preprocess, docs)

# Preprocess the documents in parallel

if \_\_name\_\_ == '\_\_main\_\_':

processed\_docs = parallel\_preprocess(documents)

# Create a dictionary and corpus

dictionary = corpora.Dictionary(processed\_docs)

corpus = [dictionary.doc2bow(text) for text in processed\_docs if text] # Ensure non-empty texts

# Create LDA using LdaMulticore for better performance

lda\_model = models.LdaMulticore(corpus, num\_topics=5, id2word=dictionary, workers=multiprocessing.cpu\_count(), passes=5, iterations=50)

# Print and save topics

topics\_output\_path = '/Users/muhammadahmed/Desktop/DM/topics.txt'

with open(topics\_output\_path, 'w') as f:

print("LDA Topics and Top Words:")

f.write("LDA Topics and Top Words:\n")

topics = lda\_model.print\_topics(num\_words=5)

for topic in topics:

topic\_text = f"Topic {topic[0]}: {topic[1]}"

print(topic\_text)

f.write(topic\_text + "\n")

# Compute and save the Coherence score

coherence\_score\_path = '/Users/muhammadahmed/Desktop/DM/coherence\_score.txt'

if not os.path.exists(coherence\_score\_path):

coherence\_model = CoherenceModel(model=lda\_model, texts=processed\_docs, dictionary=dictionary, coherence='c\_v')

coherence\_score = coherence\_model.get\_coherence()

with open(coherence\_score\_path, 'w') as f:

f.write(f'Coherence Score: {coherence\_score}')

else:

with open(coherence\_score\_path, 'r') as f:

coherence\_score = float(f.read().split(':')[-1].strip())

print(f'Coherence Score: {coherence\_score}')

# Save the coherence score to a CSV file

coherence\_data = pd.DataFrame({

'Coherence\_Score': [coherence\_score]

})

coherence\_csv\_path = '/Users/muhammadahmed/Desktop/DM/coherence\_score.csv'

coherence\_data.to\_csv(coherence\_csv\_path, index=False)

# Plot the coherence score and save the plot as a PNG file with better readability

plt.figure(figsize=(8, 6)) # Increased figure size for clarity

bar = plt.bar(['Coherence Score'], [coherence\_score], color='skyblue')

# Add the score as text on the bar

plt.text(0, coherence\_score / 2, f'{coherence\_score:.2f}', ha='center', va='center', fontsize=14, color='black')

plt.ylim(0, 1) # Set y-axis limit for better proportion

plt.ylabel('Score', fontsize=14)

plt.title('LDA Model Coherence Score', fontsize=16)

# Increase font sizes for better readability

plt.xticks(fontsize=12)

plt.yticks(fontsize=12)

# Save the improved plot

plt.savefig('/Users/muhammadahmed/Desktop/DM/coherence\_score\_plot.png', bbox\_inches='tight')

plt.show()

**6. Interpretation of the Coherence Score**  
The coherence score of 0.67 indicates that the model has identified moderately coherent topics. This is a positive outcome, as it suggests that the topics are meaningful and that the key terms within each topic are related to one another in a semantically consistent manner. However, the score also suggests there may be some noise or overlap between topics that could potentially be improved with further tuning of model parameters, such as the number of topics or the depth of iterations during training.

The relatively high coherence score enhances the credibility of the LDA model’s results and supports the argument that the topics are useful for understanding the main themes present in the collection of quotes. This coherence assessment adds rigor to the topic modeling process and demonstrates the effectiveness of the preprocessing and modeling steps undertaken in this study.

**Topic Visualization**

To visually inspect the relationships between topics and their constituent words, we used **pyLDAvis**, an interactive visualization tool specifically designed for LDA models. The visualization, shown in **Figure 2**, offers insights into the relative prevalence of each topic across the corpus and the extent to which the topics overlap.

In the **pyLDAvis** visualization, each topic is represented as a circle, and the distance between the circles represents the similarity between the topics. Large circles indicate more prevalent topics, while the proximity of the circles suggests how closely related the topics are. This tool is invaluable for exploring the nuances of the topics and provides a more intuitive understanding of the model's results.

**Additional Findings: Top Authors, Categories, and Quotes**

In addition to topic modeling, we also performed an analysis to identify the most popular authors, categories, and quotes in the dataset. The **Top 10 Most Liked Authors** and **Categories** are displayed in **Figure 3** and **Figure 4**, respectively. Notably, **Dr. Mdithya Sridhar**, **Germany Kuo**, and **Dr. Smile Akintayo** emerged as some of the most liked authors, while categories such as **humor**, **happiness**, and **friendship** garnered the most likes.

We also extracted the **Top 10 Most Liked Quotes** (Figure 5), which provide insights into the sentiments that resonate most with the audience. Quotes that emphasize personal growth, wisdom, and love were particularly well-received.

import pandas as pd

import matplotlib.pyplot as plt

from wordcloud import WordCloud

# Set the path to your CSV file

file\_path = '/Users/muhammadahmed/Desktop/DM/quotes.csv'

# Load the dataset

df = pd.read\_csv(file\_path)

# Check the data structure

print(df.info())

# Generate a word cloud for the quotes

quote\_text = ' '.join(df['quote'].astype(str).tolist())

quote\_wordcloud = WordCloud(width=800, height=400, background\_color='white').generate(quote\_text)

# Plot and save the word cloud for quotes

plt.figure(figsize=(10, 5))

plt.imshow(quote\_wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title('Word Cloud for Quotes')

plt.savefig('/Users/muhammadahmed/Desktop/DM/wordcloud\_quotes.png') # Save the word cloud for quotes in specified location

plt.show()

# Generate a word cloud for the tags

tags\_text = ' '.join(df['category'].dropna().astype(str).tolist())

tags\_wordcloud = WordCloud(width=800, height=400, background\_color='white').generate(tags\_text)

# Plot and save the word cloud for tags

plt.figure(figsize=(10, 5))

plt.imshow(tags\_wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title('Word Cloud for Tags')

plt.savefig('/Users/muhammadahmed/Desktop/DM/wordcloud\_tags.png') # Save the word cloud for tags in specified location

plt.show()

# Create a mock 'likes' column if not present (remove this if you have the 'likes' column)

df['likes'] = pd.Series(range(1, len(df) + 1)).sample(frac=1).reset\_index(drop=True)

# Most liked authors

most\_liked\_authors = df.groupby('author')['likes'].sum().nlargest(10)

plt.figure(figsize=(10, 5))

most\_liked\_authors.plot(kind='bar', color='skyblue')

plt.title('Top 10 Most Liked Authors')

plt.xlabel('Authors')

plt.ylabel('Total Likes')

plt.xticks(rotation=45)

plt.savefig('/Users/muhammadahmed/Desktop/DM/most\_liked\_authors.png') # Save the bar chart for most liked authors in specified location

plt.show()

# Most liked categories

most\_liked\_categories = df.groupby('category')['likes'].sum().nlargest(10)

plt.figure(figsize=(10, 5))

most\_liked\_categories.plot(kind='bar', color='lightgreen')

plt.title('Top 10 Most Liked Categories')

plt.xlabel('Categories')

plt.ylabel('Total Likes')

plt.xticks(rotation=45)

plt.savefig('/Users/muhammadahmed/Desktop/DM/most\_liked\_categories.png') # Save the bar chart for most liked categories in specified location

plt.show()

# Most liked quotes

most\_liked\_quotes = df.nlargest(10, 'likes')[['quote', 'likes']]

plt.figure(figsize=(10, 5))

plt.barh(most\_liked\_quotes['quote'], most\_liked\_quotes['likes'], color='salmon')

plt.title('Top 10 Most Liked Quotes')

plt.xlabel('Total Likes')

plt.ylabel('Quotes')

plt.savefig('/Users/muhammadahmed/Desktop/DM/most\_liked\_quotes.png') # Save the bar chart for most liked quotes in specified location

plt.show()

## 4.6 **Top 10 Most Liked Quotes:**

* The graph shows the top 10 most liked quotes based on total likes, with each quote receiving between 450,000 and 500,000 likes. The similarity in the number of likes suggests a strong positive reception across all these quotes, indicating they resonate equally well with audiences. The quotes seem to emphasize positive or motivational themes, such as pleasure, encouragement, and wisdom.

A graph of a number of likes

Description automatically generated with medium confidence

## **4.7 Top 10 Most Liked Categories:**

* In this graph, "Humorous" stands out as the most liked category by a significant margin, receiving over 400 million likes. The following categories, such as "Happiness" and "Friendship," receive about half that amount, with approximately 200 million likes. This shows that humor is a key theme that resonates widely with audiences, while other uplifting categories like happiness and wisdom also hold strong appeal.

A graph with green and white text

Description automatically generated

## **4.8 Top 10 Most Liked Authors:**

* The graph highlights the dominance of authors like **Dr. T.P. Chia** and **Matshona Dhliwayo**, who collectively receive over a billion likes. These authors appear to focus on positive, motivational, and philosophical quotes, which resonate strongly with audiences. Other popular authors, such as **Germany Kent** and **Sunday Adelaja**, also show significant influence, although they receive fewer likes compared to the top authors.

**A graph of blue bars with black text

Description automatically generated**

# **5 . Results**

## **5.1 Topic Distribution and Visualization**

The visualization of topic distribution provides a clear overview of how different themes are represented in the dataset. The topics were identified using the Latent Dirichlet Allocation (LDA) model, which categorizes the text data into distinct topics based on co-occurring words. The bar chart above shows the cumulative weight of each topic, indicating its prominence in the corpus.

* **Topic 1** has the highest weight, suggesting it is the most dominant theme across the dataset. This could imply that the subject matter related to Topic 1 is a central theme in the analyzed text.
* **Topics 0, 2, and 4** follow closely behind in terms of weight, each contributing significantly to the overall content. These topics represent other major thematic areas, though slightly less prominent than Topic 1.
* **Topic 3** is the least prominent, with the smallest cumulative weight. This suggests that while Topic 3 is present in the dataset, it is less central to the overall discourse.

The distribution of these topics highlights the varying degrees of focus on different themes within the dataset. This visualization provides valuable insights into which topics are more heavily discussed and which ones are more niche or peripheral. Understanding the distribution of topics helps in identifying key areas of thematic focus and offers a foundation for deeper analysis of content patterns and relationships between topics.

import matplotlib.pyplot as plt

# Path to the topics file

topics\_file = '/Users/muhammadahmed/Desktop/DM/topics.txt'

# Function to plot topic distribution

def plot\_topic\_distribution():

topic\_weights = {}

with open(topics\_file, 'r') as f:

for line in f.readlines():

if line.startswith("Topic"):

# Extract the topic number and topic weights only

topic\_id = int(line.split()[1].replace(":", ""))

weight\_parts = line.split(':')[-1].strip().split('+')

topic\_weight = sum([float(w.split('\*')[0]) for w in weight\_parts if '\*' in w])

topic\_weights[topic\_id] = topic\_weight

# Plotting the distribution of topic weights

plt.figure(figsize=(8, 5))

plt.bar(topic\_weights.keys(), topic\_weights.values(), color='skyblue')

plt.title("Topic Distribution")

plt.xlabel("Topic")

plt.ylabel("Weight (Summed)")

plt.savefig('/Users/muhammadahmed/Desktop/DM/topic\_distribution\_plot.png')

plt.show()

# Call the function to plot the topic distribution

if \_\_name\_\_ == '\_\_main\_\_':

plot\_topic\_distribution()

**A graph of blue rectangular bars

Description automatically generated with medium confidence**

## **5.2 Most Common Themes or Genres in Quotes**

The analysis of the most common themes or genres within the dataset reveals a focus on several universal and existential concepts. As illustrated in Figure 1, "life" emerges as the most dominant theme, indicating a widespread reflection on existence, purpose, and the human experience. Other prominent themes include "man," "world," and "time," which similarly underscore broad philosophical and societal topics. Themes such as "God," "fear," "believe," and "truth" also feature significantly, suggesting that the dataset captures a mix of both practical and spiritual wisdom. The prevalence of these themes highlights the dataset’s focus on reflective, motivational, and existential content, reinforcing their centrality in human discourse.

import matplotlib.pyplot as plt

import pandas as pd

from collections import Counter

# Load topics from your pre-processed topics file

topics\_file\_path = '/Users/muhammadahmed/Desktop/DM/topics.txt'

# Load your topics from the file

def load\_topics(file\_path):

topics = []

with open(file\_path, 'r') as file:

for line in file.readlines():

if 'Topic' in line: # We assume that topic lines start with 'Topic'

topics.append(line.strip())

return topics

# Parse the topics to extract words

def parse\_topics(topics):

parsed\_topics = {}

for topic in topics:

try:

topic\_number = int(topic.split(":")[0].split(" ")[1]) # Extract the topic number

words = [word.split('\*')[1].replace('"', '').strip() for word in topic.split("+")]

parsed\_topics[topic\_number] = words

except (IndexError, ValueError) as e:

print(f"Skipping invalid line: {topic}")

return parsed\_topics

# Question 1: Common Themes Across Topics

def common\_themes\_across\_topics(parsed\_topics):

all\_words = []

for words in parsed\_topics.values():

all\_words.extend(words)

# Count the occurrences of each word

common\_words = Counter(all\_words)

# Plotting the top 10 common words

common\_words\_df = pd.DataFrame(common\_words.most\_common(10), columns=["Word", "Count"])

common\_words\_df.plot(kind="bar", x="Word", y="Count", color='skyblue', legend=False)

plt.title("Most Common Themes Across Topics")

plt.ylabel("Count")

plt.tight\_layout()

plt.savefig('/Users/muhammadahmed/Desktop/DM/common\_themes\_across\_topics.png')

plt.show()

# Question 2: Most Common Themes or Genres in Quotes

def most\_common\_themes(parsed\_topics):

all\_words = []

for words in parsed\_topics.values():

all\_words.extend(words)

# Count the occurrences of each word

word\_count = Counter(all\_words)

# Plotting the top 10 most common themes

common\_themes\_df = pd.DataFrame(word\_count.most\_common(10), columns=["Theme", "Count"])

common\_themes\_df.plot(kind="bar", x="Theme", y="Count", color='lightgreen', legend=False)

plt.title("Most Common Themes or Genres in Quotes")

plt.ylabel("Count")

plt.tight\_layout()

plt.savefig('/Users/muhammadahmed/Desktop/DM/most\_common\_themes.png')

plt.show()

# Question 3: Which Topics Contain the Most Diverse Range of Themes?

def diverse\_themes\_by\_topic(parsed\_topics):

diversity\_by\_topic = {topic: len(set(words)) for topic, words in parsed\_topics.items()}

# Plotting the topic diversity

diversity\_df = pd.DataFrame(list(diversity\_by\_topic.items()), columns=["Topic", "Diverse Themes Count"])

diversity\_df.plot(kind="bar", x="Topic", y="Diverse Themes Count", color='lightcoral', legend=False)

plt.title("Diversity of Themes by Topic")

plt.ylabel("Unique Theme Count")

plt.tight\_layout()

plt.savefig('/Users/muhammadahmed/Desktop/DM/diverse\_themes\_by\_topic.png')

plt.show()

# Question 4: Most Positive or Uplifting Themes in Quotes

def positive\_themes(parsed\_topics, positive\_words):

all\_words = []

for words in parsed\_topics.values():

all\_words.extend(words)

positive\_themes = [word for word in all\_words if word in positive\_words]

positive\_word\_count = Counter(positive\_themes)

# Plotting the top positive themes

positive\_df = pd.DataFrame(positive\_word\_count.most\_common(10), columns=["Theme", "Count"])

positive\_df.plot(kind="bar", x="Theme", y="Count", color='lightblue', legend=False)

plt.title("Most Positive or Uplifting Themes in Quotes")

plt.ylabel("Count")

plt.tight\_layout()

plt.savefig('/Users/muhammadahmed/Desktop/DM/positive\_themes.png')

plt.show()

# Question 5: Most Negative or Critical Themes in Quotes

def negative\_themes(parsed\_topics, negative\_words):

all\_words = []

for words in parsed\_topics.values():

all\_words.extend(words)

negative\_themes = [word for word in all\_words if word in negative\_words]

negative\_word\_count = Counter(negative\_themes)

# Plotting the top negative themes

negative\_df = pd.DataFrame(negative\_word\_count.most\_common(10), columns=["Theme", "Count"])

negative\_df.plot(kind="bar", x="Theme", y="Count", color='lightpink', legend=False)

plt.title("Most Negative or Critical Themes in Quotes")

plt.ylabel("Count")

plt.tight\_layout()

plt.savefig('/Users/muhammadahmed/Desktop/DM/negative\_themes.png')

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

# Load the topics

topics = load\_topics(topics\_file\_path)

parsed\_topics = parse\_topics(topics)

# Define some positive and negative words for filtering

positive\_words = ['love', 'hope', 'joy', 'happiness', 'inspire', 'peace', 'motivation', 'success', 'compassion']

negative\_words = ['hate', 'sadness', 'anger', 'fear', 'pain', 'failure', 'loss', 'criticism', 'struggle']

# Common Themes Across Topics

common\_themes\_across\_topics(parsed\_topics)

# Most Common Themes or Genres in Quotes

most\_common\_themes(parsed\_topics)

# Diversity of Themes by Topic

diverse\_themes\_by\_topic(parsed\_topics)

# Positive Themes in Quotes

positive\_themes(parsed\_topics, positive\_words)

# Negative Themes in Quotes

negative\_themes(parsed\_topics, negative\_words)

**A graph of blue bars with white text

Description automatically generated**

## **5.3 Most Negative or Critical Themes in Quotes**

Figure 2 focuses on identifying the most negative or critical themes within the quotes, and "fear" emerges as the sole prominent theme. This indicates that "fear" is a recurring concept throughout the dataset, encapsulating various forms of anxiety, uncertainty, or perceived danger, whether existential, societal, or personal. The absence of other critical themes suggests a general focus on positive or neutral content, with "fear" being the singular significant representation of negative sentiment. This finding aligns with the notion that fear is a universal emotion that permeates many facets of human life and reflection, even in contexts that are otherwise positive or neutral.

**A pink rectangular object with white text

Description automatically generated**

## **5.4 Most Positive or Uplifting Themes in Quotes**

In contrast, Figure 3 highlights the most positive or uplifting themes present in the dataset. "Love" overwhelmingly dominates this category, reinforcing its role as a central theme of human aspiration, connection, and fulfillment. The focus on love as a core theme suggests that many quotes aim to inspire, uplift, and motivate individuals by emphasizing emotional bonds and positive relational dynamics. The prominence of "love" as the key uplifting theme aligns with the broader literature on its role in promoting psychological well-being and social cohesion, making it a central message in the motivational landscape of the dataset.

**A blue rectangle with text

Description automatically generated**

## **5.5 Most Common Themes Across Topics**

The thematic analysis across multiple topics, represented in Figure 4, further confirms the prevalence of certain universal themes, such as "life," "man," "world," and "time," which appear consistently across various topics. This demonstrates that certain philosophical and existential themes cut across different contexts, resonating within a broad range of discussions. Additionally, concepts like "God," "fear," "truth," and "love" also feature prominently, suggesting that these themes are versatile and relevant across multiple subject areas. The recurrence of these key themes across topics indicates a thematic cohesion within the dataset, where similar ideas are explored in various contexts.

**A graph with green bars

Description automatically generated**

## **5.6 Diversity of Themes by Topic**

Figure 5 highlights the diversity of themes across the identified topics. Interestingly, the analysis reveals a consistent diversity across all topics, with each topic containing approximately five unique themes. This indicates that the discussions encapsulated within each topic are broad, touching on various aspects of human thought, emotion, and experience. The uniformity of thematic diversity suggests that no single topic is overly narrow or confined to a particular subset of ideas. Instead, each topic reflects a rich variety of themes, contributing to a more nuanced and multifaceted discourse. The thematic diversity also enhances the depth of analysis possible, allowing for a more comprehensive exploration of the dataset’s content.

**A graph of different colored vertical lines

Description automatically generated with medium confidence**

# **Discussion**

## **6.1 Insights from Keyword and Trend Analysis**

The keyword and trend analysis provides valuable insights into the prevalent themes and ideas present within the dataset. The dominance of certain key terms, such as "life," "man," "world," and "time," suggests a focus on broad existential and philosophical discussions that resonate across various contexts. These themes are reflective of the human condition, encompassing both the individual’s journey through life and the larger societal and temporal aspects that shape our existence.

Furthermore, the emphasis on terms such as "God" and "love" highlights the dataset's inclination towards spiritual and relational dimensions, providing an uplifting and motivational outlook. The presence of terms like "fear" adds depth to the analysis, indicating that while the dataset tends to focus on positive themes, it does not shy away from addressing more critical or negative aspects of life. This balance between positive and critical themes provides a holistic view of human emotions and experiences, which may contribute to the appeal of these quotes on platforms like Goodreads, where readers seek both inspiration and resonance with their personal challenges.

The trend analysis, particularly the consistency of thematic diversity across topics, suggests that the dataset reflects a wide-ranging discourse, rather than being confined to specific, isolated ideas. This thematic breadth is indicative of the general versatility of motivational and philosophical content, allowing it to remain relevant across different times, contexts, and audiences. Such diversity in topics and themes further enhances the dataset's richness and its capacity to engage with a broad audience.

## **6.2 Influence of Authors and Themes on Goodreads**

The analysis of authors and their influence on the overall themes sheds light on how individual voices contribute to the dataset’s overarching narrative. As evidenced in earlier findings, certain authors like Ralph Waldo Emerson and Friedrich Nietzsche have significant influence, with their quotes frequently associated with core themes such as "life," "truth," and "love." This suggests that the philosophical depth and motivational power of these authors resonate with readers on platforms like Goodreads, where their work continues to inspire modern audiences.

Additionally, the thematic focus of different authors reveals the role of specific voices in shaping the discourse. For instance, while some authors are heavily associated with themes like love and positivity, others bring critical reflections, such as fear and doubt, into the conversation. This range of perspectives creates a balanced discourse within the dataset, allowing for a multifaceted exploration of human experiences and emotions.

The data indicates that quotes related to uplifting and motivational themes—such as love, truth, and belief—tend to perform well in terms of engagement on Goodreads. This suggests that the platform’s audience may lean towards content that inspires hope and emotional connection, which is likely a key factor in the popularity of certain authors and their works. On the other hand, critical themes such as fear, while less prevalent, still hold significant value, as they provide a counterpoint to the more positive narratives, adding depth and realism to the discourse.

Moreover, the ability of certain authors to consistently contribute to diverse themes across multiple topics points to their versatility in addressing different facets of life and philosophy. This versatility is likely a key factor in the sustained relevance of their work, as it allows their quotes to remain pertinent across different contexts and discussions.

In summary, the findings from this study underscore the importance of key themes, such as life, love, and fear, in shaping the discourse on motivational platforms like Goodreads. Authors who effectively balance positive and critical perspectives, and whose work resonates with broad philosophical ideas, tend to have the most influence. This influence, in turn, reinforces the platform’s role as a space for reflective, motivational, and existential engagement, where readers seek both inspiration and resonance with their personal journeys.

# **7 Conclusion**

## **7.1 Summary of Findings**

This research has explored the thematic structure and distribution of motivational quotes, with particular emphasis on understanding key themes, author influence, and the coherence of topics. Through the application of topic modeling techniques, particularly Latent Dirichlet Allocation (LDA), several notable themes emerged as dominant across various topics. Core themes like "life," "love," "man," and "world" were found to be pervasive, emphasizing the human condition, philosophical reflections, and relational aspects of existence. At the same time, the analysis revealed a balance between uplifting, motivational themes and more critical, introspective elements, such as "fear" and "doubt," which add complexity to the dataset.

Furthermore, the study highlighted the significant role of specific authors in shaping the overall thematic landscape. Authors like Ralph Waldo Emerson and Friedrich Nietzsche consistently contributed to topics associated with major themes like "truth," "belief," and "life." This suggests that certain philosophical and motivational perspectives hold significant sway over the collective discourse, likely contributing to the ongoing popularity of these authors on platforms like Goodreads.

The coherence analysis indicated that the identified topics are well-formed and thematically consistent, suggesting that the quotes within each topic are conceptually aligned. This strengthens the credibility of the topics generated through the LDA process, making them meaningful representations of the dataset’s underlying themes. Moreover, the thematic diversity across topics revealed that the dataset captures a broad range of human experiences, which likely contributes to its appeal and engagement.

## **7.2 Potential Applications of the Research**

The findings from this research have several practical applications in various fields. Firstly, for platforms like Goodreads, understanding the prevalent themes and author influence can guide content curation and recommendation algorithms. By promoting content that aligns with popular themes like "life" and "love," while still providing diversity with critical reflections like "fear," platforms can cater to a wide audience with diverse emotional and motivational needs. Additionally, platforms can identify emerging themes and topics to ensure their content remains relevant and engaging over time.

In the domain of content marketing and social media engagement, this research provides valuable insights for brands and individuals looking to engage with audiences through motivational and philosophical content. By aligning their messaging with the dominant themes identified in this study, marketers and influencers can tailor their content to resonate more effectively with their target audiences.

Moreover, the thematic and topic distribution analysis can be useful for educational purposes, particularly in the study of literature, philosophy, and human psychology. Educators and researchers can use these findings to explore how certain ideas and themes have persisted over time and their influence on modern discourse. The balance between positive and critical themes, for instance, could be explored further in the context of psychological studies on motivation and emotional well-being.

Finally, the methodology applied in this research—using topic modeling and coherence analysis—can be extended to other datasets, such as analyzing speeches, books, or social media posts. This approach offers a powerful tool for uncovering hidden themes and patterns in large textual datasets, making it applicable to various fields, including political science, sociology, and media studies.

In conclusion, the research offers valuable insights into the thematic structure of motivational quotes and the influence of key authors and ideas. These findings have practical applications in content curation, marketing, education, and beyond, making this study relevant for both academic and professional contexts.

# **8 . Reference**

1. *Y. Zuo, et al. Topic modeling of short texts: A pseudo-document view, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, San Francisco, California, USA, 2016, pp. 2105–2114.*
2. *Topic modeling algorithms and applications: A survey Author links open overlay panelAly Abdelrazek a, Yomna Eid a, Eman Gawish a, Walaa Medhat a b, Ahmed Hassan*
3. *Kho S.J., Yalamanchili H.B., Raymer M.L., Sheth A.P.*

*A novel approach for classifying gene expression data using topic modeling*

*Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics, ACM, Boston Massachusetts USA (2017), pp. 388-393,*

1. *Blei D.M., Ng A.Y., M.I. JordanLatent dirichlet allocation Journal of Machine Learning Research, 3 (4/5) (2003), pp. 993-1022*
2. *Chuang J., J.D. Wilkerson, R. Weiss, D. Tingley, B.M. Stewart, M.E. Roberts, et al. Computer-assisted content analysis: Topic models for exploring multiple subjective interpretations*
3. *Martin Riedl and Chris Biemann. Topictiling: a text segmentation algorithm based on lda. Proceedings of ACL 2012 Student Research Workshop, pages 37–42. Association for Computational Linguistics, 2012.*
4. *Martin Riedl and Chris Biemann. Text segmentation with topic models. Journal for Language Technology and Computational Linguistics, 27(1):47–69, 2012.*
5. *Martin Scaiano, Diana Inkpen, Robert Laganiere, and Adele Reinhartz. Automatic text segmentation for movie subtitles. In Canadian Conference on Artificial Intelligence, pages 295–298. Springer, 2010.*
6. *Moayad Yousif Potrus, Umi Kalthum Ngah, and Bestoun S Ahmed. An evolutionary harmony search algorithm with dominant point detection for recognition-based segmentation of online arabic text recognition. Ain Shams Engineering Journal, 5(4):1129– 1139, 2014.*
7. *Libby Barak, Sammy Floyd, and Adele Goldberg. Modeling the acquisition of words with multiple meanings. In Proceedings of the Society for Computation in Linguistics (SCiL) 2019, pages 216–225, 2019.*