INTERNATIONAL SCIENTIFIC CONFERENCE
20-22 November 2025
GABROVO

# APPLICATION OF MODERN TOOL FOR TEXT ANALYSIS

**Danica B. Milošević[1], Ana Vukic[2], Dušan B. Regodić[3], Borivoje M. Milošević[4*]**

*[1] Academy of Applied Technical and Preschool Studies, Niš, Serbia,*
*[2] University MB, Faculty of Business and Law, Belgrade, Serbia,*
*[3] University MB, Faculty of Business and Law, Belgrade, Serbia,*
*[4] University MB, Faculty of Business and Law, Belgrade, Serbia,*
** Corresponding author: borivojemilosevic@yahoo.com*

**Abstract**
*In this paper we are concerned with the various ways in which computer systems can analyze and interpret texts, and we will assume, for the sake of convenience, that these texts are presented in electronic format. This paper introduces some important concepts, techniques, and terminology that will be used for this purpose. NLP is a field that addresses this very challenge, enabling machines to read, understand, and analyse textual data for all purposes. Some of the material in this paper is somewhat technical, and therefore the paper presents program code for LDA algorihm that can be used for this purpose.*
*In this paper, we will explore the main areas of NLP technology, including tokenization, lemmatization, frequency analysis, sentiment analysis, coherence score and language generation. Each of these technologies has its own specific applications and benefits, which we will also present through text analysis example.*
**Keywords:** NLP, LDA model, Lexical Analysis, Semantic Analysis, Sentiment Analysis, Dirichlet Algorithm..

## INTRODUCTION

Textual analysis [1] of the sum of qualitative data is very challenging. Such analyzes are even more difficult when the topic is controversial, and the results can influence important corporate, marketing, social and even political decisions. The paper explores modern methods of text analysis for qualitative research, using new techniques for language processing ( Natural Language Processing - NLP), and (Latent Dirichlet Allocation - LDA), for the purpose of categorization, organizing and finding hidden topics, as well as summarizing a long context based on a case study and mashine learning, which analyzes certain comments in free text, resulting in all controversial decisions and their impact on the environment.

## TEXT ANALYSIS PROCESS

NLP - Natural Language Processing [2] is all about making computers learn, understand, analyze, manipulate and interpret natural (human) languages and text. The process of text analysis can be divided into several stages:

*Descriptive Stage*: In this phase, the analyst begins with a detailed reading and summarizing of the text. He or she attempts to understand the context, the author's perspective, and the target audience that the text may be of interest to.

*Analytical Stage*: In this phase, analysts draw conclusions and interpretations by examining the emergence of certain concepts, recurring themes, and patterns in the text being analyzed.



*Print ISSN: 1313-230X, Online ISSN: 3033-1404*

*Interpretive Stage*: In this phase, analysts discover an understanding of symbolism and other linguistic nuances inherent in the text, defining the underlying meanings of implicit messages and metaphorical representations.

*Evaluative stage*: This phase answers the question of how interesting or persuasive the text is, how well-defined the arguments are, or how influential the presentation is on the reader. There are predefined standards and criteria based on which the text is processed.

## IMPLEMENTING LDA ALGORITHM FOR TEXT ANALYSIS

Latent Dirichlet Allocation (LDA - Latent Dirichlet Allocation) [3], which in 2003 introduced by David Blei, Andrew Ng, and Michael Jordan in 2010, is a probabilistic generative model for a variety of discrete data, especially textual content. LDA, Table I, assumes that documents are composed of a mixture of topics, and each topic is a word-based distribution. Its goal is to reconstruct this process by reverse engineering in order to reveal the latent topics that lie at the very base of the document.

TABLE I
THE KEY ELEMENTS OF LDA ARE [3-4]:

| | |
|---|---|
| A. Documents: | In the context of LDA, documents are represented as a model of a set of words, where the order of words is ignored, and only their frequency is important. Each element in the vector corresponds to the number of a certain word, where the position of the element represents the index of the word in the dictionary. |
| B. Topic: | Topics are latent (hidden) thematic structures, Fig. 1, that represent sets of words that often appear together in documents. Each document is assumed to be associated with a mixture of topics, and each topic is characterized by a word distribution. |
| C. Words: | Words are individual concepts present in documents. In LDA, a word is associated with a certain topic with a certain probability. The model assumes that each word in the document is generated by an arbitrary choice of a topic from the distribution of topics, and then by an arbitrary choice of words from the distribution of words, we get the selected topic. |

Source: https://medium.com/@pinakdatta/understanding-lda-unveiling-hidden-topics-in-text-data-9bbbd25ae162

LDA works under the assumption that each document can be represented as a mixture of topics, and each word in the document is assigned to one of the document's topics. The goal of LDA is to infer those distributions of topics that best explain the analyzed documents.

By analyzing these proportions, we can understand the underlying themes or subjects in the document [3-4-5].

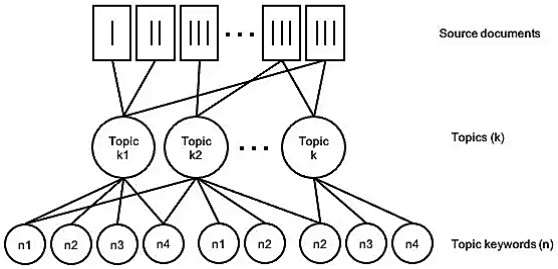
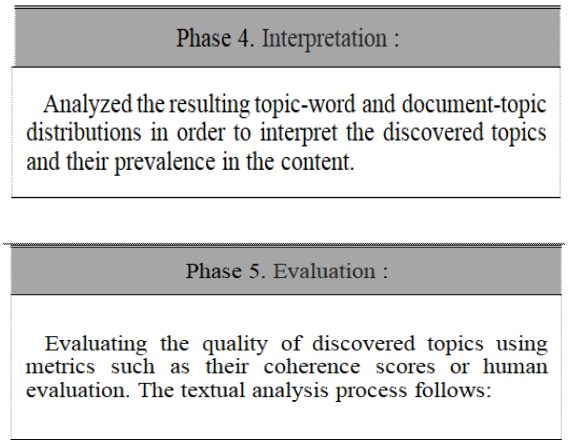



***Fig. 1.*** *The process of organizing the topic structure*

To analyze textual data using LDA, we will follow these algorithm steps, [3-4-5]:

TABLE II
ANALYZE TEXT DATA USING LDA, ALGORITHM STEPS [3-4-5]:

| |
|---|
| Phase 1. Preprocessing: |
| Preparation of text data by removing stop words, punctuation marks and other irrelevant symbols. We also perform stemming or lemmatization to reduce words to their basic form. |
| Phase 2. Construct a Document -Term Matrix (DTM): |
| Converting textual data into a matrix representation, where rows correspond to documents, and columns to unique words throughout the entire content. Each cell in the matrix contains the frequency of word occurrence in the corresponding document. |
| Phase 3. Topic Modeling with LDA : |
| Let's apply the LDA algorithm to the DTM to find out the basic topics in the document. This includes inferring the distribution of topics for each document and the distribution of words for each topic. |




*Source: https://medium.com/@pinakdatta/understanding-lda-unveiling-hidden-topics-in-text-data-9bbbd25ae162*

We start the analysis using the Anaconda3/Spyder application by importing the necessary libraries, including gensim for the implementation of LDA algorithms, and we define the content of the document by topic [3-4-5-6-7].

- Then we tokenize the documents, breaking up a stream of characters into words, punctuation marks, numbers and other discrete items.
- We create a dictionary that maps each word to a unique ID and convert tokenized documents into a representation of a set of words.
- Using the LdaModel class from gensim.models, we train the LDA model [8] on a set of words with a certain number of topics.
- At the end, we write the topics together with the most likely words, of course, strictly related to each topic.
- Now we demonstrate how to perform topic modeling and text analysis of the document bible_en.txt using LDA in Python.
- For consideration, we took from the Bible only the section Genesis 1:1 to Genesis 1:10 and not the entire content of the document, because the reports in that way would be too long [9].

Of course, the text analysis process in this case shows all the described results. The complete program follows:

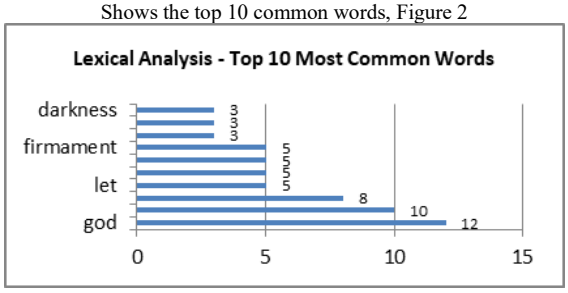
```
# Import necessary libraries
from gensim import corpora
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel
from pprint import pprint
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from collections import Counter
import string
# Download required NLTK data
nltk.download( ['punkt', 'stopwords', 'vader_lexicon'])
# Sample documents
documents = [
"Genesis 1:1 In the beginning God created the heavens and the earth. ",
"Genesis 1:2 Earth was without form, and void; and darkness [was] upon the face of the deep. And the Spirit of God moved upon the face of the waters. ",
"Genesis 1:3 God said, Let there be light: and there was light. ",
"Genesis 1:4 God saw the light, that [it was] good: and God divided the light from the darkness. ",
"Genesis 1:5 God called the light Day, and the darkness he called Night. And the evening and the morning were the first day. ",
"Genesis 1:6 God said, Let there be a firmament in the midst of the waters, and let it divide the waters from the waters.",
```

```
"Genesis 1:7 God made the firmament, and divided the waters which [were] under the firmament from the waters which [were] above the firmament: and it was so. ",
"Genesis 1:8 God called the firmament Heaven. And the evening and the morning were the second day.",
"Genesis 1:9 God said, Let the waters under the heaven be gathered together unto one place, and let the dry [land] appear: and it was so. ",
"Genesis 1:10 God called the dry [land] Earth; and the gathering together of the waters called he Seas: and God saw that [it was] good."]

# ===== Lexical Analysis =====
print( "\n===== Lexical Analysis =====")
# Tokenize and clean words
stop_words = set( stopwords.words('english') + list(string.punctuation))
all_words = []
for doc in documents:
    words = word_tokenize(doc.lower())
    words = [word for word in words if word not in stop_words and word.isalpha()]
all_ words.extend( words)

# Word frequency analysis
word_freq = Counter( all_words )
print( "\nTop 10 Most Common Words:")
for word, freq in word_freq.most_common(10):
    print( f"{word}: {freq}")

# Vocabulary richness
unique_words = set( all_words )
lexical_density = len( unique_words) / len(all_words)
print( f"\nVocabulary Size: {len(unique_words)}")
print( f"Lexical Density: {lexical_density:.2f}")
# ===== Semantic Analysis =====

print( "\n===== Semantic Analysis =====")
# Tokenize the documents
tokenized_docs = [[word for word in doc.lower( ).split() if word not in stop_words] for doc in documents]
# Create a dictionary mapping each word to a unique id
dictionary = corpora.Dictionary(tokenized_docs)
# Convert tokenized documents into bag-of-words representation
corpus = [dictionary.doc2bow(doc) for doc in tokenized_docs]
# Train the LDA model with additional parameters
lda_model = LdaModel( corpus, num_topics=2, id2word=dictionary, random_state=42, passes=10)
# Print the topics
print( "\nDiscovered Topics:")
pprint( lda_model.print_topics())

# Calculate coherence score
coherence_model_lda = CoherenceModel( model=lda_model, texts=tokenized_docs, dictionary=dictionary,
coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence ()
print( f\nCoherence Score: {coherence_lda:.3f}')

# ===== Sentiment Analysis =====
print( "\n===== Sentiment Analysis =====")
sia = SentimentIntensityAnalyzer()
for i, doc in enumerate(documents):
    scores = sia.polarity_scores(doc)
    print( f"\nDocument {i+1}: {doc}")
    print( f"Positive: {scores['pos']:.2f}, Negative: {scores['neg']:.2f}, Neutral: {scores['neu']:.2f}")
    print( f"Compound Sentiment: {scores['compound']:.2f}")
# Overall sentiment
all_text = " " .join(documents)
```

```
overall_sentiment = sia.polarity_ scores( all_text)
print( "\nOverall Sentiment Analysis:")
print(f"Positive:{overall_sentiment['pos']:.2f},Negative: {overall_sentiment['neg']:.2f}, Neutral:
{overall_sentiment['neu']:.2f}")
print( f"Compound Sentiment: {overall_sentiment['compound']:.2f}"
```

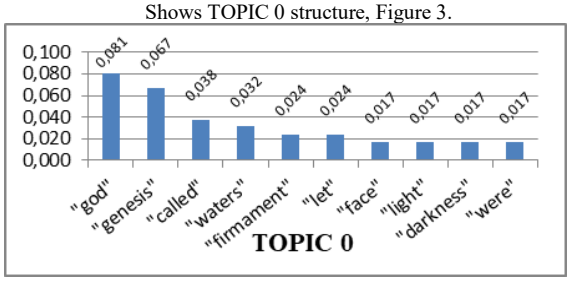
The results of the execution of the program show the following reports:

===== Lexical Analysis =====
Shows the top 10 common words, Figure 2



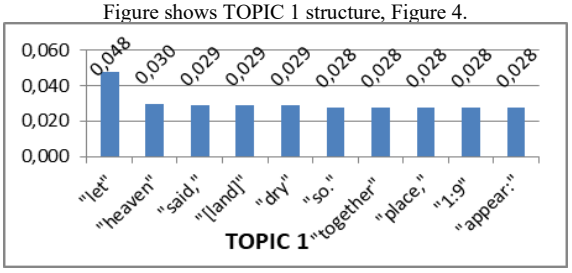

***Fig, 2.*** *Common Words with Vocabulary size: 28 and Lexical Density: 0.64*

===== Semantic Analysis =====
Shows TOPIC 0 structure, Figure 3.




***Fig, 3.*** *TOPIC 0 Distribution*

Figure shows TOPIC 1 structure, Figure 4.




***Fig, 4.*** *TOPIC 1 Distribution*

The output consists of two topics (Topic 0 and Topic 1), each represented by a list of words along with their corresponding weights (importance in the topic). Here's what it means:

**TOPIC 0:** Main Theme: Likely related to "Creation in Genesis" (Biblical context).
Key Words: "god" (highest weight, most important word)

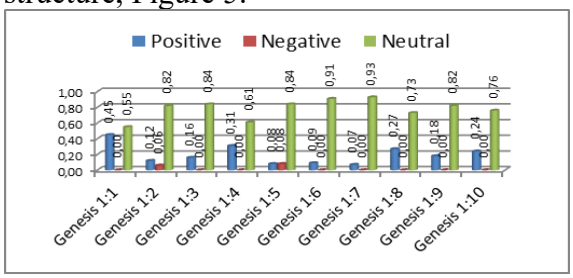
"genesis" (second most important)
"waters," "firmament," "light," "darkness" (related to the Biblical creation story).

**TOPIC 1:** Main Theme: Possibly "Divine Commandments / Structuring the World" (still Biblical).
Key Words:"let," "heaven," "said," "land," "dry" (suggesting God's commands in Genesis).
"1:9" (likely a Bible verse reference).

Each topic is a mixture of words, where higher weights (e.g., 0.081*"god") mean the word is more representative of that topic.
Coherence Score: 0.435

**===== Sentiment Analysis =====**

Figure shows Sentiment Analysis structure, Figure 5.




***Fig, 5.*** *Sentiment Analysis - Positive, Negative and Neutral score*

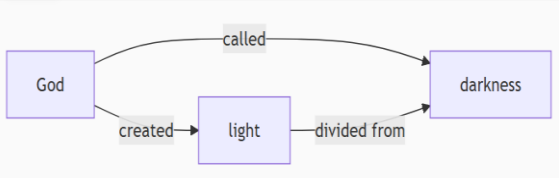
We will consider positive and negative reactions. [10-11]

Overall Sentiment Analysis:
Positive: 0.18, Negative: 0.03, Neutral: 0.79
Compound Sentiment: 0.99

The final results of the execution of the program showed the following reports:

*Lexical*: The text has a moderate lexical density (0.81 ) with the word "god" as the most common word in the document.

*Semantic*: Clear themes emerged (god, light, darkness) and a representation of the relationships between the words in the themes is given in the Figure 6.




***Fig, 6 .*** *Relationships between the words*

*Sentiment analysis*: Mostly neutral with one slightly positive word God.

## CONCLUSION

By applying (NLP) technology, we have demonstrated the possibility of understanding, interpreting and in-depth analysis of text in a meaningful and useful way, which is used in many fields today. The most common applications include main themes, keywords, sentiment analysis and lexical/semantic data analysis, which are important as useful tools for many analysts and scientists of our society.

For this purpose, we used Latent Dirichlet Allocation (LDA) which offers a powerful approach for discovering these hidden thematic structures that provide valuable insights within textual data and their meaning and form.

## REFERENCES

[1.] A. Khan, "Textual Analysis: Definition, Approaches and Examples", *https://www.lettria.com/blogpost/textual-analysis-definition-approaches -and-examples,* Oct 19, 2023

[2.] R. Kibble, "Introduction to natural language processing", *University of London, Department of Computing, Goldsmiths* 2013.

[3.] P. Datta, "Analyzing Text Data with Topic Modeling: Latent Dirichlet Allocation (LDA) Explained ", Medium, Apr 8, 2024

[4.] D. Jurafsky, J.Martin, "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models ", *Stanford University*, 2025.

[5.] J.Golberg, "Neural Network Methods for Natural Language Processing", *Part of the book series: Synthesis Lectures on Human Language Technologies (SLHLT), Synthesis – collection of Technogogies*, 2025, Springer.

[6.] "Text Analytics Toolbox User's Guide", *Guideby The MathWorks, Inc*, COPYRIGHT 2017–2024

[7.] L. Bolelli, S.. Ertekin, and C.Giles, "Topic and Trend Detection in Text Collections Using Latent Dirichlet Allocation ", *European conference on information*, 2009 – Springer

[8.] S. Ke, J. Montiel, J. Nesbit, "Robust machine learning algorithms for text analysis", *Quantitative Economics 15* (2024), 939–970

[9.] https://www.biblegateway.com/passage/?search=Genesis%201&version=KJV

[10.] B.Kumar, P.Kumar, T.Swapna, at all, "Natural Language Processing for Sentiment Analysis ",*International Conference on Cognitive and Intelligent Computing,* ICCIC 2023, 8-9 December, 2025, Hyderabad, India.

[11.] S. Date, K. Sonkamble, S. Deshmukh, "Sentiment Analysis Using Computer-Assisted Text Analysis Tools" , International Conference on Applications of. Machine Intelligence and Data Analytics, ACSR 105, pp. 671–679, 2023.