TSDA: TOPIC STABILITY DRIVEN APPROACH WITH DATA FLOW ANALYSIS AND HYPERPARAMETER OPTIMIZATION FOR ENHANCED TOPIC SENTIMENT SUMMARIZATION

Abstract: Our study introduces a new method called Topic Stability Driven Approach with Data Flow Analysis and Hyperparameter Optimization. This method aims to make the topics we find in documents more stable and reliable. We use two types of Topic models: LDA and LSA. To improve these models, Grid Search and Genetic Algorithm is used to optimize their settings. To ensure the topics are stable, we evaluate them using coherence score metrics for each document. This helps to make sure the topics make sense and stay consistent across different analyses. We also analyze how sentiments change across different topics in the text and we fine-tune our methods using hyperparameter optimization. Our experiments show that, TSDA works well across different datasets and topics. By using this method, we can better understand how people feel about specific subjects. Our method provides a promising way to improve topic sentiment summarization, making it more useful and reliable in various real-world situations.

Keywords: Natural Language Processing, Hyperparameter Optimization, Topic Models, Data Flow Analysis, Grid Search, Genetic Algorithm, Topic Stability, Summarization.

1. INTRODUCTION

In today's modern landscape flooded with textual data, the understanding of sentiments and opinions expressed in written content has become increasingly essential. From social media updates to news articles, texts serve as data collections of valuable insights into people's feelings and attitudes towards various subjects. Sentiment analysis, a process aimed at extracting and understanding emotions and sentiments hidden within large amounts of textual information. One crucial application of sentiment analysis lies in topic sentiment summarization, which involves extracting the sentiments expressed about specific subjects within a corpus of text. This effort provides invaluable insights into public opinion and customer feedback. For instance, when analyzing product reviews, it is important not just to figure out what customers like or dislike about a product but also to understand how people feel about the product overall.

The way people use language online is always changing, which means we need better ways to understand the feelings behind what people write. Traditional methods for analyzing sentiment often struggle to keep up with the different kinds of language used in various situations. This indicates the necessity for evolving approaches to accommodate the evolving patterns of online expression. Our paper presents a novel technique termed as Topic Stability Driven Approach with Data Flow Analysis and Hyperparameter Optimization to help summarize the feelings about different topics in documents. It combines ideas from stable methods, of understanding how data flows, and techniques to find the best settings automatically. This helps us get a deeper understanding of how people feel about the topics they write online. To achieve this, we have used two types of models: LDA and LSA. These models help us identify the main themes in a document. We have also applied techniques like Grid Search and Genetic Algorithm to make these models work even better. These techniques help us find the best settings for our models automatically. One big challenge in understanding text is making sure the topics we identify actually make sense. To address this, we have used coherence score metrics. These metrics help us evaluate how well the topics fit together within each document. Thus, the topics we find are meaningful and consistent.

2. OBJECTIVES

Our main objectives are as follows:

1. Ensure topic coherence within documents for consistent and meaningful topics.
2. Analyze sentiments across topics associated with sentences to understand people's feelings better.
3. Implement the data flow analysis in TSDA to perform topic sentiment summarization accuracy and maintain original feelings.
4. Fine-tune methods with hyperparameter optimization for effective topic sentiment summarization

3. RELATED WORK

3.1 THEME EXTRACTION

Latent Dirichlet Allocation (LDA) [2] is a method for understanding large collections of text, like articles or books. The functioning of LDA involves conceptualizing each as blend of various topics and within each topic, there exists blend of different words. By figuring out these mixes, LDA helps us see what topics are present in a document and how important they are. It is uncovering hidden themes in a big pile of writings, which can be useful for tasks such as organizing articles, understanding what they're about, or even recommending similar ones. Nonnegative Matrix Factorization (NMF) is a method for understanding text without relying heavily on assumptions about the data or models. [10] tells about a novel approach called Deep NMF (DNMF) to make topic modeling easier by using a deep learning method to learn hidden structures in documents and then applying this knowledge to find important words related to specific topics. They show that their method works well compared to other techniques when tested on different types of text.

Topical n-grams [11] is a theme extraction technique that goes beyond the traditional BoW assumption. It focuses on capturing not only individual words but also phrases, considering the importance of word order in understanding text. By sampling topics and their associated word statuses (unigram or bigram) in sequence, the model can identify meaningful phrases within topics, leading to more interpretable topics and improved performance in tasks like information retrieval.

3.2 PARTS OF SPEECH

Sentiment-Analysis (SA) using grammatical category (POS) [3] tagging is a method where text data is analyzed to determine the emotional tone conveyed. In simpler terms, it involves breaking down a sentence into its individual words and identifying their roles, such as nouns, verbs, adjectives, and so on. By understanding the grammatical structure of a sentence, it becomes possible to infer the sentiment expressed within it. Certainly, words such as "happy," "joyful," or "satisfied" tend to evoke positive sentiments, whereas terms like "sad," "angry," or "disappointed" commonly express negative emotions. By analyzing the distribution and usage of such words in a sentence, SA enables informed assessments regarding whether the prevailing sentiment leans towards positivity, negativity, or neutrality. This approach provides a more nuanced understanding of the sentiment behind text data, enabling applications ranging from customer feedback analysis to social media monitoring.

3.3 UNSUPERVISED EXTRACTIVE SUMMARIZATION

Text summarization condenses a long document into a concise summary, capturing its key points. Few deep learning models demand lots of labeled data, which is scarce for lesser-known languages. Moreover, training such models requires significant computational resources due to their complexity. In response, LFIP-SUM, an unsupervised summarization [4] model. It doesn't need labeled data for training. Instead, it relies on pre-trained sentence embeddings and integer programming. Our model automatically selects important sentences using principal component analysis. While it doesn't learn parameters, experiments show it performs comparably well to deep learning models. LFIP-SUM presents a promising solution for summarization challenges, particularly in resource-constrained environments.

3.4 PARAMETER TUNING

Machine learning models play a crucial role in various fields today. To improve their accuracy, different methods of adjusting hyperparameters are explored. So, our work compares Grid-Search and Genetic-Algorithm (GA) [7]. Grid Search tries every combination of hyperparameters, which can take a long time for complex models. Genetic Algorithms mimic nature to improve hyperparameters gradually over time. The effectiveness of these techniques is evaluated in an Arabic sentiment classification task, addressing the challenge of sentiment analysis in a complex language like Arabic. The analysis highlights the strengths and limitations of each tuning technique. This study evaluates Grid Search and Genetic Algorithm for Arabic sentiment analysis, uncovering their strengths and limitations. It provides insights for optimizing machine learning models in practical scenarios.

3.5 EXTRACTIVE SUMMARIZATION

Method for summarizing Arabic text, which aims to condense the important information while reducing the overall text length. It employs a framework centered around graphs, wherein sentences are depicted as nodes. A Modified PageRank algorithm [5] is then utilized to rank sentences, prioritizing them according to the abundance of nouns they incorporate. This approach outperforms other Arabic text summarization methods like Lex-Rank and Text-Rank, especially when using 10,000 iterations of the Modified-Page-Rank algorithm. and applying the Modified-Page-Rank algorithm. [15] the challenge of sifting through vast amounts of unstructured data on the web and social media by introducing methods for text summarization. It critiques the constraints of manual summarization and past machine learning methods, presenting a novel deep learning model surpassing predecessors in accuracy and correctness, assessed via BLEU and ROUGE metrics on extensive data.

3.6 OPINION MINING

SA alternatively termed as Opinion Mining. Its emphasis lies in discerning emotional nuances within text, notably on platforms such as Twitter [12], where opinions are plentiful yet frequently disorganized. This survey explores techniques for analyzing Twitter data to determine whether opinions are positive, negative, or neutral, using methods like ML models and Word-Inventory based approaches, alongside scoring methods and various algorithms such as N-Bayes, and Support Vector Machine.It can indeed be applied in education to evaluate students' learning experiences, but current algorithms may not fully replace human raters. In a study using various machine learning algorithms to analyze students' sentiments [13] about learning experiences, results showed high accuracy in identifying positive and negative sentiments, but struggled with neutral sentiments.

However, an algorithm using word-sentiment associations achieved decent accuracy without needing extensive pretraining, indicating potential for improvement with more educational datasets. It is a field within natural language processing, focuses on understanding human emotions and opinions. This study examines how large language models [14] like ChatGPT perform in various sentiment analysis tasks, finding that while they excel in simpler tasks, they struggle with more complex analyses. Additionally, the study proposes a new benchmark for evaluating sentiment analysis models.

3.7 DATA FLOW ANALYSIS

In modern compilers, the Static Single Assignment (SSA) form helps analyze and optimize programs. Converting programs into SSA involves placing ϕ-functions, which can be complex. Existing methods, like dominance frontiers (DF), assume all variables start at the program's beginning, which isn't always true for local variables. A new ϕ-placement algorithm based on data flow analysis [9] (DFA) is introduced in this paper to address this issue. The correctness and complexity of this approach are demonstrated through theorems and proofs.

The introduction of the new ϕ-placement algorithm based on data flow analysis (DFA) marks a significant advancement in compiler technology. By addressing the limitations of existing methods like dominance frontiers (DF), which assume uniform variable initialization, this algorithm promises greater accuracyin program analysis. Through rigorous testing in the Clang/LLVM compiler framework, the approach showcased notable outcomes, significantly reducing unnecessary ϕ-functions and enhancing precision. Despite the increased computational demands, the method's efficiency in analyzing a substantial proportion of procedures.

One fundamental formula used in data flow analysis is the iterative solution for computing the 'in' and 'out' sets for each program point in a control flow graph. These sets represent the information flowing into and out of each program point, respectively.

The iterative equations for computing the 'in' and 'out' sets are as follows:

1. In sets:

   In[S] = Union of {Out[P] | P Belongs to Pred[S]}

2. Out sets:

   Out[S] = Generated set of S union (Input set of S minus Killed set of S).

Where:

- S is a basic segment in the CFG.
- Pred[S] represents the set of predecessor basic segments of S.
- The generated set(S) indicates the set(definitions) generated by basic segment S.
- Killed set(S) indicates the set(definitions) killed by basic segment S.

4. TSDA FRAMEWORK

Our new method, called TSDA, improves topic models like LDA and LSA by making topics more consistent. We use techniques like Grid Search and Genetic Algorithm to enhance these models. By evaluating topics using coherence scores and analyzing sentiment, we ensure more reliability. Hyperparameter optimization fine- tunes our approach, showing effectiveness across different datasets.

4.1 DATA CLEANING

Data cleaning involves preparing text data for analysis. This includes removing HTML tags, special characters, and digits from the text. Punctuation marks are also eliminated. Then, the text is split into individual words or tokens through tokenization. Lemmatization is applied to standardize words to their base form, ensuring consistency. Contraction expansion is used to convert contracted forms into their full expressions for better accuracy. These steps help maintain the integrity and reliability of analytical outcomes.

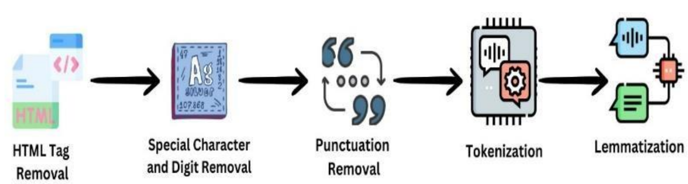

Fig. 1. Steps involved in data cleaning

4.2 DOCUMENT TERM MATRIX

The Document-Term Matrix (DTM) is like a table that shows how many times each word appears in each document of our text collection. We create a list of all the unique words from these text documents as dictionary. Then, for each document corpus is formed using BoW. This DTM helps us analyze text more easily, allowing us to see which words are common in different documents and how they relate to each other.

4.3 TOPIC MODELLING

Topic modeling serves as a method for uncovering the predominant themes or subjects within a compilation of documents. In our study, we use two common techniques for topic extraction: LDA and LSA.

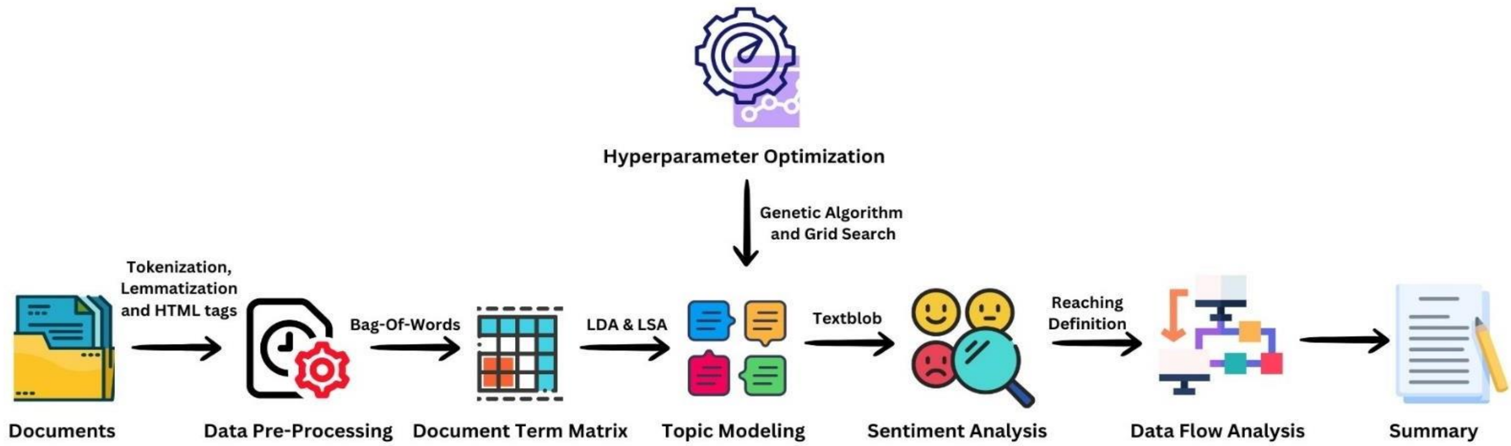

Fig. 2. Overall Workflow of TSDA Framework

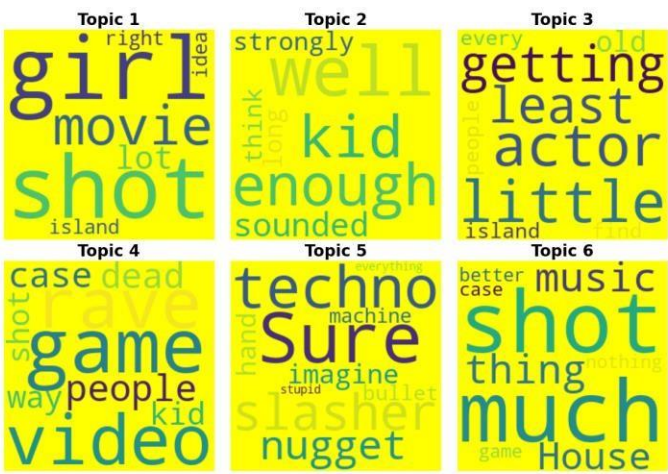

Fig. 3. Word Cloud for Topics and Topic words

1. LATENT DIRICHLET ALLOCATION

LDA serves as statistical framework employed to reveal hidden topics within a corpus of documents. It functions based on the premise that separate documents comprise a blend of dissimilar topics, with every word in document linked to one of these topics. LDA works by iteratively assigning words to topics and adjusting the topic assignments to enhance the potential of observing the actual spread of words in the documents. Through this iterative process, LDA uncovers the underlying topics and their associated word distributions. These topics are represented as probability distributions over words, indicating the likelihood of a word appearing in a document given a particular topic. Overall, LDA provides a powerful framework for automatically discovering the latent themes or topics present in large text corpora.

2. LATENT SEMANTIC ANALYSIS

LSA is a method used for understanding the relationships between words and documents in an extensive corpus of textual data. It works by analyzing the frequency of word occurrences across documents and representing them in a high-dimensional matrix. LSA utilizes Singular-Value-Decomposition (SVD) to condense the feature space of the matrix while preserving vital semantic correlations. This reduction process helps to uncover latent, or hidden, semantic structures within the text data.

By identifying patterns of word co-occurrence, LSA can capture the underlying meaning or context of words. This means that even if words don't appear together frequently, LSA can still recognize similarities in their usage and group them accordingly. In essence, LSA enables us to understand the semantic associations between words and documents.

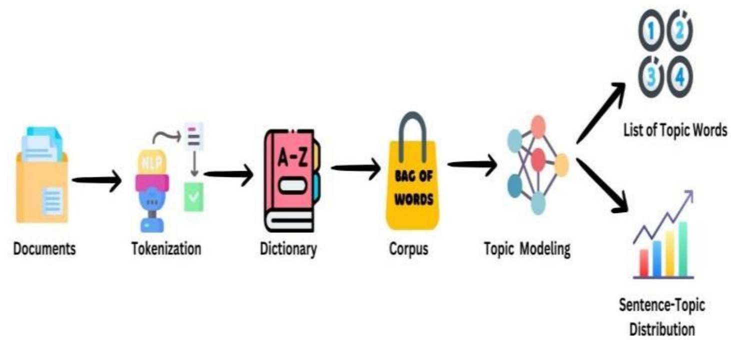

Fig. 4. Topic modeling encompasses both LDA & LSA

4.4 HYPERPARAMETER OPTIMIZATION

Hyperparameter optimization involves adjusting the parameters of a model to enhance its performance to the highest possible level. In our study, we focus on improving the performance of our topic modeling method by adjusting parameters such as alpha, beta, decay, random state, min probability, which we call hyperparameters. Think of hyperparameters as dials on a machine that control how it operates. Two techniques we use for this optimization are Grid Search and Genetic Algorithm.

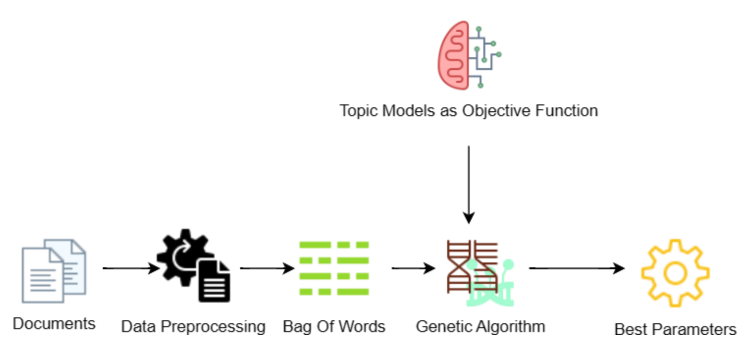

Fig. 5. Finding Best Parameters for LDA Model

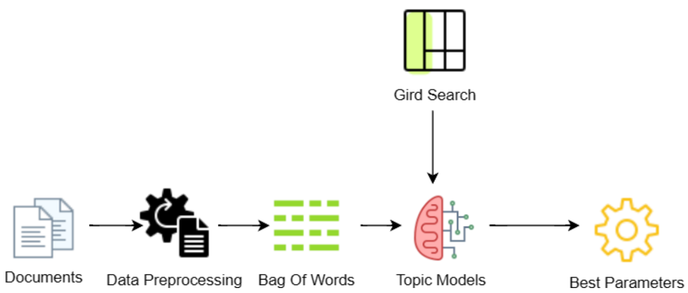

Fig. 6. Finding Best Parameters for LSA Model

Grid Search is a systematic way of trying out different combinations of hyperparameters to see which ones work best. We create a grid of possible hyperparameter values and test each combination to see which gives us better negative log-like-lihood score.

On the other hand, GA mirrors the process of Darwinian selection to find the optimal hyperparameters. It starts with a population of potential solutions (sets of hyperparameters) and evaluates their performance. Then, it selects the best-performing solutions and combines them to create new ones. This process continues over several generations until it finds the best combination of hyperparameters, like evolution refining the traits of a species over time. By using these techniques, we ensure that our topic modeling method works effectively across different types of text and topics. This optimization process is crucial for making our method more accurate and reliable, ultimately leading to better understanding of people's sentiments and more useful insights in real-world applications.

4.5 Topic Stability

Topic coherence checks how words in a topic fit together, making sure they make sense and are consistent. We use coherence scores to see how reliable our topics are. This helps us understand better what the data is about. For example, if we're talking about animals, coherence ensures that words like "dog," "cat," and "pet" are in the same topic, making it clearer and more consistent for analysis. These scores help us understand how well the words within each topic connect with each other.

4.6 Sentiment Calculation

In this process, each sentence in a document is linked with topics generated from topic models based on the sentence's topic distribution. Any topics with a low relevance (less than or equal to 0.1) are discarded as unwanted. Then, the words in the sentences are compared with the words associated with the remaining topics. If there's a match, sentiments are combined for each sentence. Sentiments are determined using a dictionary called Text-Blob, which assigns sentiment scores to words based on their meaning. Essentially, this method helps in understanding the overall sentiment expressed in each sentence by considering the words' sentiments and their association with specific topics.

4.7 Reaching Definition Analysis

The document analysis process involves initially determining the sentiment of each sentence and then executing a reaching definition on the (topic, sentiment) tuple for every sentence to establish the killed and generated sets of definitions. Subsequently, if a specific topic reappears with either the same or a different sentiment, the previous definition is invalidated, and the current one is regarded as generated, facilitating continuous recalculations. This procedure extends throughout the entirety of the document, enabling the computation of both IN and OUT sets. Ultimately, the summary is extracted from the last node of the OUT set, encapsulating the key themes and sentiments of the document.

4.8 Sentiment Classifation

After creating summaries, we check their sentiment using TextBlob, which sorts them into positive, negative, or neutral categories. We repeat this process for the original documents as well. This helps us understand the overall tone of both the summaries and the actual content, whether they express positivity, negativity, or neutrality. This method gives us valuable insights into the sentiment conveyed in the text, aiding in better comprehension and analysis.

Algorithm 1. Sentiment Calculation

Require: sentences S = $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \ldots, \boldsymbol{s_n}\}$; topic words W = $\{\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_n}\}$;

Ensure: sentiment scores for each sentences

```
Initialize an empty list to store sentiment scores
for each sentence s_i in sentences S do
    Initialize sentiment_score = 0
    for each word in word_tokenize(s_i) do
        if word in topic words W then
            Calculate sentiment score of the word (TextBlob)
            Add sentiment score of the word to sentiment_score
    Append sentiment_score to sentiment_scores
end for
return sentiment_scores
```

Alternative dictionaries besides Text-Blob exist for SA, offering diverse approaches and lexicons tailored to specific contexts. One such option is VADER, known for its effectiveness in capturing sentiment nuances, especially in social media texts. Senti-WordNet, based on Word-Net syn-sets, assigns sentiment scores to words based on their syn-set's positivity, negativity, and neutrality. Another popular choice is A-FINN, a simple yet efficient dictionary-based method that assigns polarity scores to words. These alternatives provide flexibility and customization in SA, catering to varied needs and domains beyond what Text-Blob offers.

Algorithm 2. Data Flow Analysis

Require: Topic index of each sentence and corresponding sentiments (index, sentiment)

Ensure: IN and OUT sets

1: Initialize empty lists: definitions_generated = [], definitions_killed = [], IN_sets = [], OUT_sets = []

2: Iterate over each row

3: Initialize empty lists: row_definitions_generated = [], row_definitions_killed = []

4: Process each topic index, and sentiment score in the row:

5: Define the current definition as (topic_index, sentiment_score)

6: Determine if any previous definitions match the current one and mark them as killed

7: Add the current definition to the generated set and merge with previous sets if not killed

8: Append generated and killed sets for the row to respective lists

9: Execute the reaching definitions algorithm to calculate IN and OUT sets

10: Initialize OUT sets for the row

11: Iterate until convergence

12: Update IN and OUT sets for each sentence based on previous sets

13: Repeat until OUT sets no longer change

14: Append IN and OUT sets to their respective lists

15: return IN and OUT sets

5. . EVALUATION

Evaluation of a summary can be approached through diverse methodologies to ascertain its effectiveness. These methods include direct confusion matrix analysis, which assesses the summary's accuracy in capturing important information. Rouge 1, 2, and L Scores assess how closely a summary matches a reference text using n-grams. Fluency & Relevance score, which evaluates the summary's readability and relevance to topics extracted. stratified cross-validation, a technique that ensures the robustness of the evaluation process by validating against various subsets of the data. Each method offers unique insights into different aspects of summary quality, collectively providing a comprehensive evaluation framework.

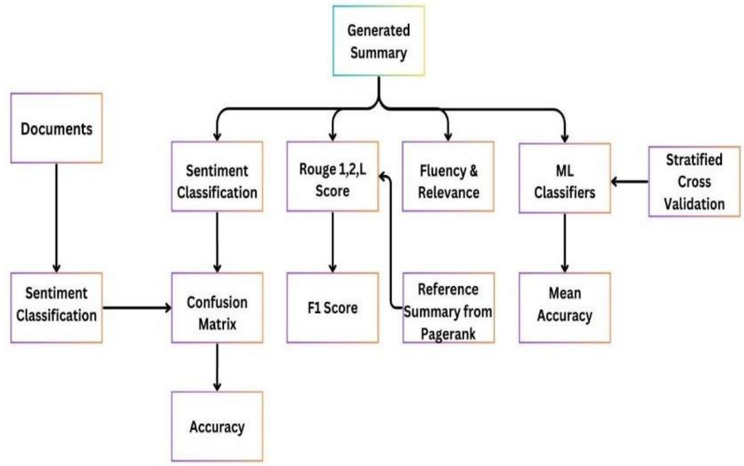

Fig. 7. Methodologies to evaluate Summaries

5.1 DIRECT CONFUSION MATRIX

Direct confusion matrix is often employed in comparing sentiment labels between the summary and the original document. The accuracy from this matrix is important. It shows how many sentiments were classified correctly, which tells us if the summary matches the feelings in the original document. It helps us see if the summary captures the right emotions and overall meaning of the original content. Below is the formula for computing accuracy.

$$\text{Accuracy} = \frac{\boldsymbol{TrPos+TrNeg}}{\boldsymbol{TrPos+TrNeg+FsPos+FsNeg}} \tag{1}$$

5.2 RELEVANCE AND FULENCY

Relevance assesses if the summary contains key words from the top-k words in N topics generated for the document. By comparing the presence of these important topic words in the summary, its effectiveness in capturing the main ideas is determined. A simple formula for this evaluation could be:

$$\text{R} = \frac{\textbf{Number of Top}-\textbf{k Topic Words in Summary}}{\textbf{Number of Top}-\textbf{k Topic Words}} \times \mathbf{100} \tag{2}$$

Let's denote R as the Relevance Score for a Single Document. A higher relevance score indicates that more key words from the document's important topics are present in the summary. To calculate the average relevance score across multiple documents, we use the following formula:

$$\text{Average Relevance} = \frac{\sum_{i=1}^{n} \boldsymbol{R_i}}{\boldsymbol{n}} \tag{3}$$

Here, $\boldsymbol{R_i}$ represents the relevance score for each separate document, and n is the total-number of documents evaluated. This formula allows us to determine the overall effectiveness of the summaries in capturing the main ideas across the entire set of documents.

To assess the readability of a summary, syntactic tagging is to analyze its grammar, determining its fluency score. A simple formula for this evaluation could be:

$$\text{F} = \frac{\textbf{Number of Grammatically Correct Sentences}}{\textbf{Total Number of Sentences}} \times \mathbf{100} \tag{4}$$

Let's denote F as the Fluency Score for a Single Document. A higher fluency scores indicate a higher proportion of grammatically sound sentences in the summary, reflecting its readability and linguistic coherence. This method helps ensure that the summary is easy to understand and effectively conveys the intended message. To calculate the average fluency score across multiple documents, we use the following formula:

$$\text{Average Fluency} = \frac{\sum_{i=1}^{n} \boldsymbol{F_i}}{\boldsymbol{n}} \tag{5}$$

Here, $\boldsymbol{F_i}$ represents the relevance score for each individual document. This formula allows us to determine the overall fluency of the summaries across the entire set of documents.

5.3 ROUGE SCORE

To assess the quality of a summary, Rouge scores (1, 2, L) are utilized by comparing it against a reference summary derived from a PageRank algorithm. These scores quantify the similarity of N-Grams mutual between the generated summary and the reference-text. A higher Rouge score indicates better similarity and coherence with the original document. The average Rouge scores across multiple documents can be calculated using the following formulas:

$$\text{Average Rouge 1-score} = \frac{\sum_{i=1}^{n} \boldsymbol{Rouge-1_i}}{\boldsymbol{n}} \tag{6}$$

$$\text{Average Rouge 2-score} = \frac{\sum_{i=1}^{n} \boldsymbol{Rouge-2_i}}{\boldsymbol{n}} \tag{7}$$

$$\text{Average Rouge L-score} = \frac{\sum_{i=1}^{n} \boldsymbol{Rouge-L_i}}{\boldsymbol{n}} \tag{8}$$

Here $\boldsymbol{Rouge - 1_i}$, $\boldsymbol{Rouge - 2_i}$, $\boldsymbol{Rouge - L_i}$ represent the Rouge scores for separate document, and n is the whole documents evaluated. These averages provide insights into the overall accuracy and coherence of the summaries across the dataset.

5.4 CROSS VALIDATION

The evaluation of summaries is enriched through stratified k-fold cross-validation, employing diverse ML classifiers such as Linear SVC, Random Forest, Multinomial-NB, Logistic Regression and ensemble of combinations of above models. This approach optimizes assessment by leveraging summaries with their sentiment labels. Accuracy, expressed as the proportion of accurate forecasts to all forecasts made., benefits as a pivotal performance metrices. The formula for average accuracy across multiple cross-validation folds is:

$$\text{Average Accuracy} = \frac{\sum_{i=1}^{n} \boldsymbol{Accuracy_i}}{\boldsymbol{n}} \tag{9}$$

Here, $\boldsymbol{Accuracy_i}$ represents the accuracy score for each fold, and k denotes the overall fold count. This method ensures the efficacy of the summarization process by validating the performance of various classifiers on the summary data.

6. EXPERIMENTS AND DISCUSSIONS

This section elucidates the frameworks within TSDA, highlighting their outcomes and the quantity of data samples utilized for each framework, demonstrating their effectiveness in summarizing documents through better evaluation and optimization techniques.

6.1 FRAMEWORKS

TSDA comprises eight frameworks: LDA Nexus, LSA Evolution, TopicBurst LDA, TopicBurst LSA, GenAlgo LDA, GenAlgo LSA, Gridify LDA, and Gridify LSA.

LDA Nexus and LSA Evolution utilize iterative methods, evaluating coherence for each document across 5 to 12 topics in their corresponding models (LDA and LSA). The model achieving the highest coherence score is chosen, and its topics, topic words, and sentence topic distributions are then utilized for subsequent analysis and processing. This iterative approach ensures that the selected model captures the most coherent and relevant topics within the dataset.

TopicBurst LDA and TopicBurst LSA similarly employ iterative strategies, aiming to identify the most suitable number of topics for the entire dataset. They achieve this by computing coherence scores across a range of potential topic numbers and then averaging these scores to determine the optimal configuration. This iterative process ensures that the selected number of topics maximizes coherence across the dataset.

On the other hand, GenAlgo LDA and GenAlgo LSA employ a genetic algorithm methodology. They first preprocess and vectorize all documents, then feed them into LDA or LSA topic models. These models aim to optimize various parameters such as alpha, eta, decay, offset, and minimum probability for LDA, and the number of topics for LSA. Through this approach, the genetic algorithm iteratively refines the parameters to enhance the performance of the topic models, ensuring more accurate and effective topic analysis.

Gridify LDA utilizes a grid search technique to optimize parameters like the number of topics, random state, alpha, and eta. On the other hand, Gridify LSA focuses particularly on determining the ideal number of topics within the scope of 5 to 15. These frameworks operate distinctively to enhance topic sentiment summarization by exploring predefined parameter spaces and adapting to various dataset characteristics, showcasing their versatility and effectiveness in optimizing topic modeling processes.

6.2 DATASETS

The IMDb movie review dataset, an integral part of TSDA, comprises 12,500 reviews expressing both positive and negative sentiments. These reviews serve as crucial data samples for sentiment analysis within the TSDA framework, enabling comprehensive analysis and optimization techniques to enhance topic sentiment summarization in movie reviews.

The table below outlines the number of samples employed for each framework within the TSDA methodology.

Table 1
Data Samples used for each framework.

TSDA Frameworks	Number of Data Samples
LDA Nexus	500
LSA Evolution	500
TopicBurst LDA	7500
TopicBurst LSA	7500
GenAlgo LDA	2500
GenAlgo LSA	7500
Gridify LDA	2500
Gridify LSA	7500

6.3 System Requirements

TSDA requires an operating system compatible with Windows, macOS, and various Linux distributions, alongside a processor like Intel Core i5 or its AMD equivalent. It is recommended to have a minimum of 8 GB RAM and at least 10 GB of available disk space for data storage and processing. Python 3.x must be installed, and users should have access to popular development environments such as PyCharm, Jupyter Notebook, or Visual Studio Code. Essential libraries and packages include NumPy, pandas, scikit-learn, Gensim, NLTK, Matplotlib, Seaborn, tqdm, and GA package.

6.4 Findings And Discussion

Figures 7 and 8 illustrate the topic stability of a document as assessed by the LDA Nexus and LDA Evaluation Framework, respectively. These visual representations depict the consistency and reliability of topics identified within the document, offering insights into the robustness of the respective frameworks' analyses.

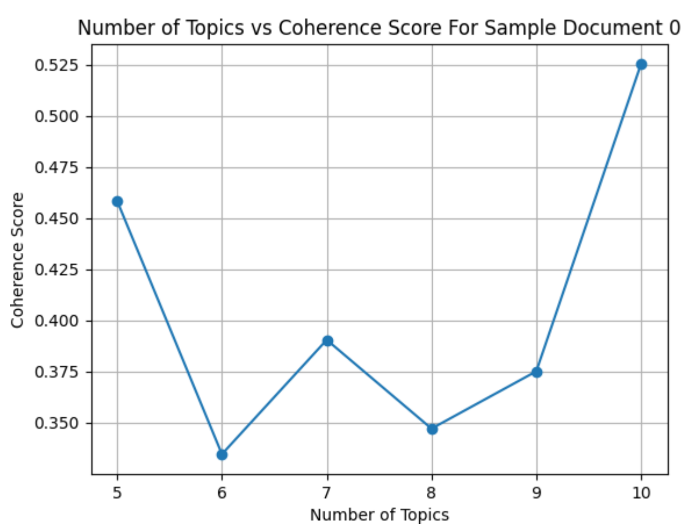

Fig. 8. LDA Nexus – Topic Stability of a document

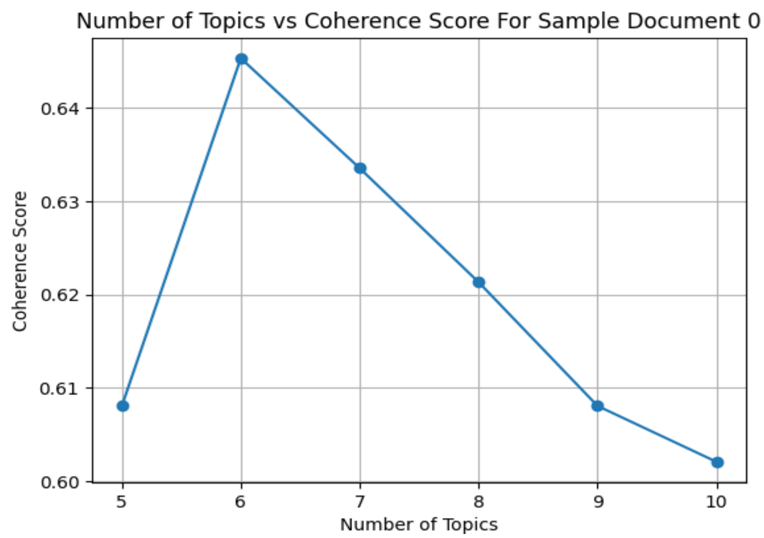

Fig. 9. LSA Evaluation – Topic Stability of a document

The above figures indicate that not all documents exhibit stability with a specific number of topics. Variations in topic stability across documents suggest the need for flexible topic modeling approaches that can adapt to individual document characteristics. After optimizing the hyperparameters of the LDA topic model for topic numbers ranging from 5 to 10, six best settings were identified for parameters like alpha, decay, offset, random state, and minimum probability. These settings are detailed in Table 3 for reference.

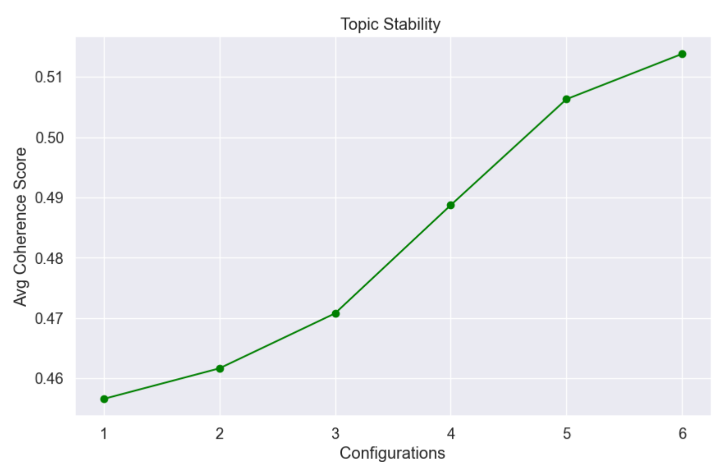

Fig. 10. GenAlgo LDA – Average coherence score vs GA Configurations

Figure 9 displays the average coherence score compared to different configurations in GenAlgo LDA. This graph helps understand how changes in configurations affect the coherence score, aiding in the optimization of Genetic Algorithm parameters for LDA topic modeling.

Table 2
Best Parameters for LDA Model from Gird Search

No of Topics	Alpha	Eta	Decay	Random State
5	0.5	0.1	0.5	42

Table 3
Best Configurations of Parameters for No of Topics from Genetic Algorithm for LDA Model

Configurations	No Of Topics	Alpha	Eta	Decay	Offset	Random State	Min Probability
1	5	0.11333	0.l802	0.6879	2.2924	23	0.0547
2	6	0.10841	0.4517	0.7843	4.8175	49	0.2030
3	7	0.10744	0.7574	0.7312	5.1262	9	0.0536
4	8	0.10874	0.2950	0.6020	9.9071	77	0.0853
5	9	0.11246	0.2876	0.6457	2.3621	86	0.0802
6	10	0.10250	0.9940	0.8194	9.3594	51	0.0102

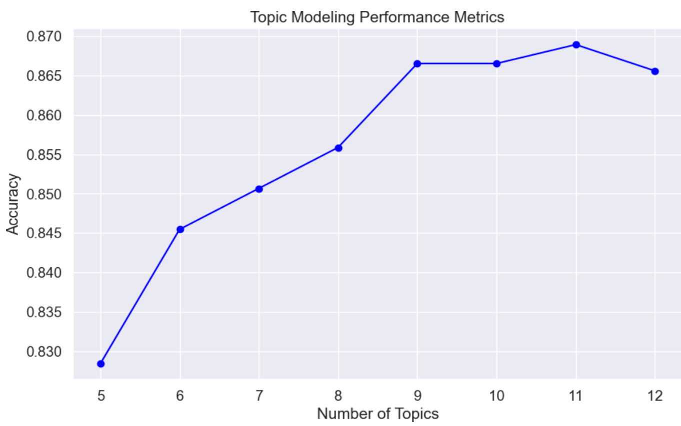

Fig. 11. TopicBurst LDA – LDA Topics vs Accuracy using Direct CM Analysis

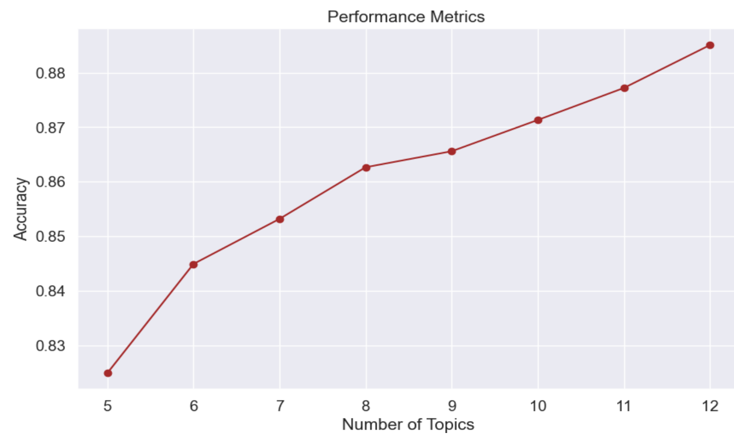

Fig. 12. TopicBurst LSA – LSA Topics vs Accuracy using Direct CM Analysis

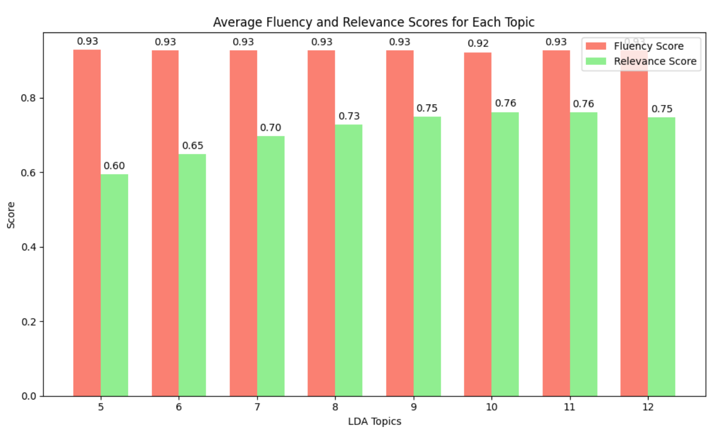

Fig. 13. TopicBurst LDA – LDA Topics vs Fluency and Relevance Score

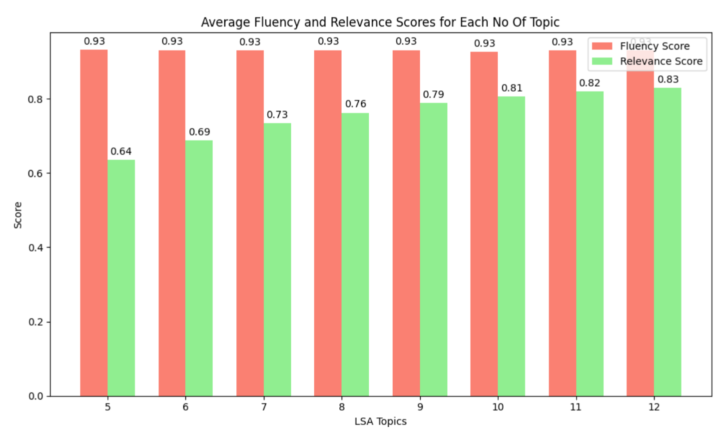

Fig. 14. TopicBurst LSA – LSA Topics vs Fluency and Relevance Score

Fig. 15. GenAlgo LDA –Configurations vs Avg Score

Figures 10 and 11 illustrate accuracy compared to the number of topics generated by LDA and LSA topic models within the TopicBurst LDA and LSA frameworks. Through direct analysis of confusion matrices, these figures reveal that both frameworks demonstrate strong performance when employing 12 topics. This suggests that the accuracy of topic modeling within these framework's peaks at 12 topics, highlighting the effectiveness of this configuration in accurately summarizing topics within the given datasets.

In Figures 13, 14, and 15, we compare Fluency and Relevance scores across different frameworks like TopicBurst LDA & LSA, and GenAlgo LDA. Interestingly, the Fluency scores remain consistent across all these models. However, when it comes to Relevance, Figure 13 stands out, showcasing high scores particularly in TopicBurst LSA Framework. Moreover, in both TopicBurst frameworks, Topic 12 exhibits notably higher Relevance scores. In the case of GenAlgo LDA, Configuration 6 demonstrates superior Relevance scores. These figures provide valuable insights into how each framework performs in terms of maintaining fluency and relevance in the context of topic sentiment summarization.

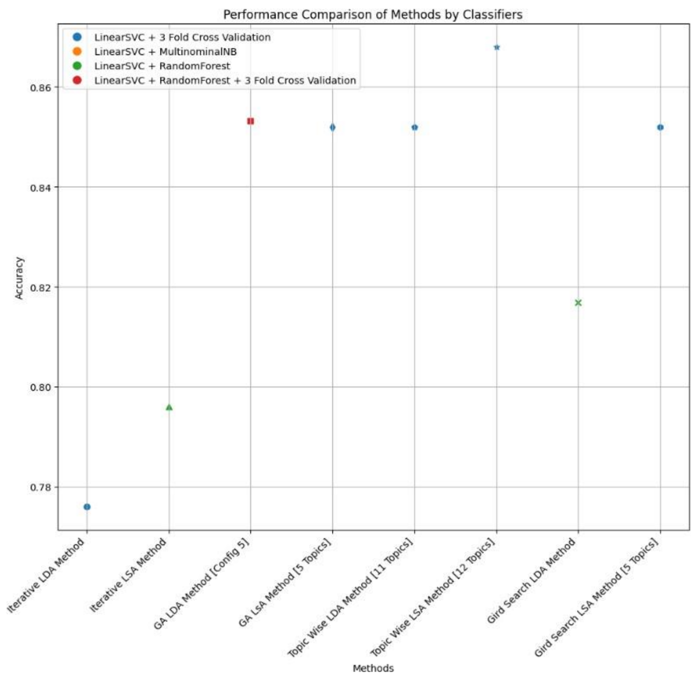

Fig. 16. ML and Cross Validation vs Frameworks with and without Ensemble of models with best settings

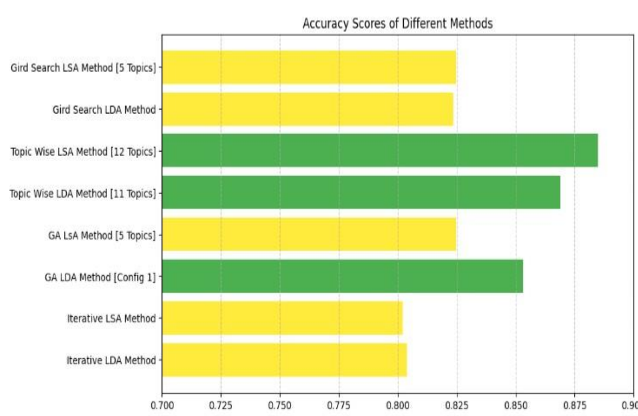

Fig. 17. Direct Confusion Matrix vs Frameworks with best settings

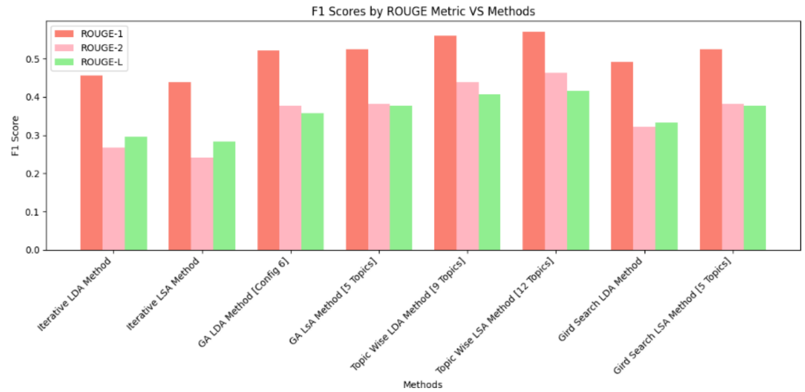

Fig. 18. Rouge 1,2, L Scores vs Frameworks

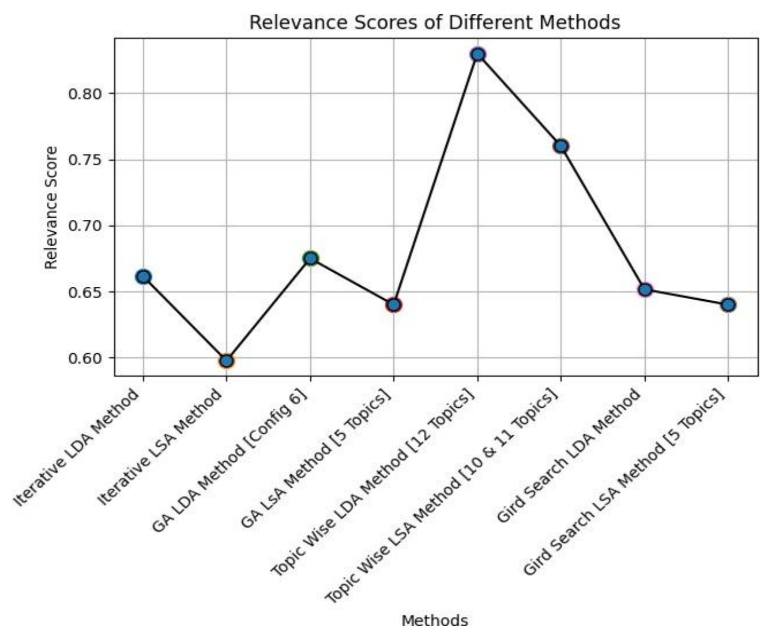

Fig. 19. Relevance Scores vs Frameworks with best settings

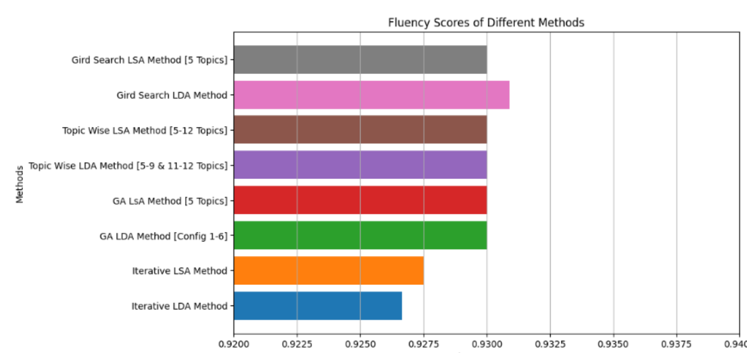

Fig. 20. Fluency Scores vs Frameworks with best settings

In Figure 20, Fluency Scores across various frameworks with their optimal settings are compared, revealing that Gridify LDA consistently outperforms other frameworks in terms of fluency. Similarly, Figure 18 displays Relevance scores among frameworks with their best configurations, with TopicBurst LDA, particularly with 12 topics, showing superior performance.

Figure 17 focuses on the evaluation of Rouge 1, 2, and L scores using different frameworks, comparing TSDA summaries with Page-Rank summaries for the same documents. Here, TopicBurst LSA, employing 12 topics, demonstrates higher performance. Additionally, Figure 16 presents Confusion Matrix results across frameworks with optimal settings, where TopicBurst LSA with 12 topics excels. Furthermore, Figure 15 highlights Machine Learning (ML) and cross-validation outcomes across frameworks, once again revealing TopicBurst LSA with 12 topics as the top performer. These figures collectively provide a comprehensive analysis of various frameworks' performances, indicating that TopicBurst LSA with 12 topics consistently achieves superior results across multiple evaluation metrics in the context of topic sentiment summarization.

7. CONCLUSION

From the comprehensive results, several observations emerge across different evaluation metrics. Firstly, regarding accuracy measured by confusion matrix analysis, both Topic Wise LDA and LSA Methods, especially with 11 and 12 topics respectively, showcase high accuracy levels, outperforming other approaches. Secondly, in terms of fluency, all methods consistently yield high scores, suggesting uniformity in linguistic quality across frameworks. However, focusing on relevance, Topic Burst LDA Method with 12 topics stands out with the highest score, indicating its effectiveness in capturing relevant information. Similarly, by Rouge scores, Topic Wise LSA Method with 12 topics consistently demonstrates superior performance across Rouge 1, 2, and L metrics, suggesting its proficiency in summarization tasks. Notably, in classification tasks, models utilizing Genetic Algorithm (GA) with various configurations exhibit competitive accuracy rates, suggesting the efficacy of optimization techniques in enhancing classification outcomes. Lastly, by relevance assessment, Topic Burst LDA Method again emerges as the top performer, underlining its capability in generating summaries deemed highly relevant. Overall, Topic Burst LDA Method with 12 topics showcases superior performance across multiple evaluation metrics, highlighting its effectiveness in topic sentiment summarization tasks.

8. FUTURE WORK

In the future, TSDA aims to explore additional techniques beyond reaching definition analysis to further enhance document summarization. This task may entail incorporating sophisticated algorithms or methodologies to enhance the efficacy and precision of the summarization procedure. Additionally, there is a focus on optimizing the Genetic Algorithm used within TSDA. This optimization effort includes identifying the best parameters such as population size and crossover types to enhance the algorithm's performance. Furthermore, TSDA seeks to enhance sentiment calculation and classification capabilities. This could involve refining existing sentiment analysis techniques or integrating new approaches to better understand and interpret sentiment within documents. Overall, these future endeavors aim to advance TSDA's capabilities in document summarization, sentiment analysis, and classification tasks, ultimately improving its effectiveness and applicability in various domains.

9. REFERENCES

[1] Li, X., Zou, C., Wu, P., & Li, Q. (2023). TSSRD: A Topic Sentiment Summarization Framework Based on Reaching Definition. "IEEE Transactions on Affective Computing", 14(3)

[2] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. "Journal of Machine Learning Research", 3, 993–1022.

[3] Kalarani, P., & Selva Brunda, S. (2019). Sentiment analysis by POS and joint sentiment topic features using SVM and ANN. "Soft Computing", 23(16), 7067–7079.

[4] M. Jang and P. Kang, "Learning-free unsupervised extractive summarization model" IEEE Access, vol. 9, pp. 14358–14368, 2021.

[5] Elbarougy, R., Behery, G., & El Khatib, A. (2020). Extractive Arabic Text Summarization Using Modified PageRank Algorithm. "Egyptian Informatics Journal", 21(2), 73-81.

[6] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," IEEE Transactions on Neural Networks.

[7] Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis". Informatics, 8(4), 79. https://doi.org/10.3390/informatics8040079.

[8] Gurusamy, V., & Kannan, S. (October 2014). "Preprocessing Techniques for Text Mining." Presented at RTRICS At: Podi. Madurai Kamaraj University.

[9] Masud, A. N., & Ciccozzi, F. (2020). "More precise construction of static single assignment programs using reaching definitions". Journal of Systems and Software, 166, Article no. 110590.

[10] J.-Y. Wang and X.-L. Zhang, "Deep NMF Topic Modeling," Journal of LaTeX Class Files, vol. 14, no. 8, August 2015.

[11] X. Wang, A. McCallum, and X. Wei, "Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval," University of Massachusetts.

[12] Kharde, V. A., & Sonawane, S. S. (2016). "Sentiment Analysis of Twitter Data: A Survey of Techniques". International Journal of Computer Applications, 139(11), 1-6.

[13] Lazrig, I., & Humpherys, S. L. (2022). "Using Machine Learning Sentiment Analysis to Evaluate Learning Impact". Information Systems Education Journal (ISEDJ), 20(1), Article 13. ISSN: 1545-679X. https://isedj.org/.

[14] W. Zhang, Y. Deng, B. Liu, S. J. Pan, and L. Bing, "Sentiment Analysis in the Era of Large Language Models" May24,2023. https://arxiv.org/abs/2305.15005.

[15] Yadav, Arun Kumar, Singh, Amit, Dhiman, Mayank, Vineet, Kaundal, Rishabh, Verma, Ankit, & Yadav, Divakar. "Extractive text summarization using deep learning approach." International Journal of Information Technology, vol. 14, no. (2022): pp. 2407–2415.