REPORT ON PROJECT STAGE - I

SUBJECTIVE ANSWER EVALUATION USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY
FOR PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING
In
Electronics and Telecommunication Engineering

By
MITUL SHAH (42357)
RAJDEEP CHAVAN (42458)
RISHIKESH TAJNE (42268)

GUIDED BY :
MR.SUNIL.S.KHOT

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE – 43

OCTOBER 2023

Department of Electronics and Telecommunication Engineering
Pune Institute of Computer Technology, Pune – 43

CERTIFICATE

This is to certify that the Project Stage - I Report entitled

"Subjective Answer Evaluation Using Natural Language Processing and Machine Learning"

has been successfully completed by

MITUL SHAH (42357)
RAJDEEP CHAVAN (42458)
RISHIKESH TAJNE (42268)

towards the partial fulfillment of the degree of **Bachelor of Engineering** in **Electronics and Telecommunication** as awarded by the **Savitribai Phule Pune University**, at **Pune Institute of Computer Technology** during the academic year 2023-24

INTERNAL GUIDE
(Mr.Sunil.S.Khot)

HOED
(Dr. M. V. Munot)

ACKNOWLEDGEMENT

Please acknowledge all those who were involved and who helped you in completing this Report / Thesis / Project Work but keep this brief and resist the temptation of writing flowery prose. Do include all those who helped you, e.g., other faculty / staff you consulted, colleagues who assisted etc. Acknowledge the source of any work that is not your own.

#

Thanking You ,
MITUL SHAH (42357)
RAJDEEP CHAVAN (42458)
RISHIKESH TAJNE (42268)

CONTENTS

ABSTRACT

This project addresses the complex task of precise subjective answer evaluation using natural language processing (NLP), with a particular focus on BERT. The study is motivated by the pressing need for accurate and efficient assessment methodologies. Our comprehensive approach encompasses various stages, including data preprocessing, BERT-based encoding, machine learning model training, and comparative analysis of different evaluation methods.

The results obtained through this project highlight the system's remarkable accuracy in evaluating semantic similarity between sentences. This achievement holds significant promise for applications across diverse domains. Looking ahead, our project opens the door to future developments and improvements, including advanced data preprocessing techniques, fine-tuning of NLP models, the development of real-time assessment systems, evaluation in different domains, and potential human-machine collaboration scenarios. This work contributes significantly to the advancement of semantic similarity assessment within the realms of natural language processing and machine learning, with implications for various industries and applications.

Abbreviations and Acronyms

BERT	Bidirectional Encoder Representations from Transformers
FCA	Formal Concept Analysis
MNB	Multinomial Naive Bayes
NaN	Not a Number
NLP	Natural Language Processing
OCR	Optical character recognition
WMD	Word Movers Distance

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

The background of the problem in subjective answer evaluation lies in the growing need for accurate and efficient methods to assess open-ended, subjective responses. This challenge has gained prominence in various fields, including education and text analysis, where the precise evaluation of subjective answers is crucial for informed decision-making and effective quality analysis..

1.2 Relevance

The topic of subjective answer evaluation is highly relevant to the field of Electronics and Telecommunication Engineering (E&TC) and related subjects. In E&TC, the ability to evaluate subjective answers accurately using NLP techniques is of great importance, as it enables a more objective and efficient assessment of students' understanding of complex concepts.

1.3 Motivation

The motivation for this project stems from the necessity to address the challenges associated with evaluating subjective answers in a precise and efficient manner. The project seeks to provide a comprehensive review of existing literature and approaches that specifically relate to the evaluation of subjective answers using NLP, ensuring a focused and meaningful survey..

1.4 Problem Definition

The problem addressed in this project is the accurate and efficient evaluation of subjective answers, particularly open-ended responses, using natural language processing (NLP) techniques. The project aims to design and implement machine learning models and algorithms that can assess the quality, relevance, and effectiveness of such subjective responses, contributing to improved assessment practices.

1.5 Scope and Objectives

The scope of this project is to offer a comprehensive overview of the methods and techniques employed in the evaluation of subjective answers using NLP. The objectives are as follows:

- Evaluate the accuracy and efficiency of machine learning models in the assessment of subjective answers.
- Identify and address the limitations of each evaluation method, thereby enhancing the overall effectiveness.
- Gain a deep understanding of the technical aspects of machine learning models, including algorithms and feature engineering.
- Provide a comparative analysis of different evaluation methods to guide researchers and practitioners in selecting the most suitable approach for their applications

1.6 Technical Approach

The technical approach in this project capitalizes on state-of-the-art natural language processing techniques, notably BERT (Bidirectional Encoder Representations from Transformers). BERT plays a pivotal role in encoding and analyzing subjective answers to evaluate their quality and relevance. The approach encompasses data preprocessing, BERT-based encoding, model training, and similarity measurement, allowing for precise and efficient subjective answer assessment. A comparative analysis of different evaluation methods enhances decision-making for researchers and practitioners.

1.7 Organization of Report:

1. **CHAPTER 1 – Introduction:** serves as the project's foundation, addressing the background, relevance, motivation, problem definition, scope, objectives, technical approach, and the report's organization. This chapter establishes the context for the entire project.
2. **CHAPTER 2 - Literature Survey:** delves into the literature survey, introducing the changing landscape of human-generated text analysis with machine learning and natural language processing (NLP). It reviews various methodologies, including Formal Concept Analysis (FCA), Hybrid Systems using Cosine and Jaccard Similarity, OCR, Jaccard, and BERT combinations, Systems of Cosine Similarity, Word Movers Distance (WMD), and Multinomial Naive Bayes (MNB), and Keyword-matching and OCR. A comparison table summarizes each method's accuracy and limitations.
3. **CHAPTER 3 – Methodology:** presents the project's technical approach, featuring machine learning and NLP techniques, primarily leveraging BERT. It outlines the steps, including data preprocessing, BERT-based encoding, model training, similarity measurement, and comparative analysis. The proposed methodology addresses the core problem of subjective answer evaluation with precision and efficiency.
4. **CHAPTER 4 - Results and Discussions:** discusses the results of the BERT-based model for semantic similarity, which offer insights into the system's performance. The chapter covers data loading, handling null and missing values, one-hot encoding of labels, and the development of a custom data generator. It also delves into the implications, utility, and potential areas of improvement and future research.
5. **CHAPTER 5 - Conclusions and Future Scope**: serves as the conclusion and mentioning future scopes, summarizing the exploration of the BERT-based model's capabilities for semantic similarity assessment.

CHAPTER 2

Literature Survey

The way we analyze and understand human-generated text is changing, thanks to machine learning and natural language processing. This survey explores how we can use these technologies to assess answers that are more about opinions and feelings. It's like having a smart system that can figure out how good or relevant these answers are. We'll look at the best methods out there and see how good they are at this job, what they can't do well, and how they work.

Jirapond Muangprathub, Siriwan Kajornkasirat, and Apirat Wanichsombat. [1] presented a document plagiarism detection system based on Formal Concept Analysis (FCA), which pre-processes source documents, extracts keywords, and constructs a concept lattice for ranking documents by similarity. Implemented as a web application, it attains an impressive 94.01% accuracy in experiments, making it suitable for text-based plagiarism detection tasks. However, it lacks specific information regarding system speed and scalability, potentially limiting its performance with large document collections. Accuracy is contingent on precise keyword extraction, and it operates on pre-processed documents, thus not conducive to real-time plagiarism detection during content creation.

Farah K and Mohammed S. H. [2] suggested a hybrid system designed for plagiarism detection using the PAN-PC-2011 dataset and offers a free application to assist users in identifying plagiarism. This approach leverages data mining techniques, natural language processing (NLP), and text mining to pre-process and analyse text. It employs Jaccard and Cosine similarity measures for document comparison, with adaptive thresholds based on experimentation. The system achieves competitive precision (0.959), recall (0.959), F1-score (0.867), and plagiarism detection (0.867) metrics when evaluated against other plagiarism detection systems. The paper provides a comprehensive overview of the methodology, including pre-processing steps, similarity measures, and evaluation metrics, demonstrating the system's effectiveness in identifying plagiarism.

S. Singh, O. Manchekar, A. Patwardhan, U. Rote, S. Jagtap and H. Chavan. [3] presented a methodology which is designed to address the challenges in efficiently and objectively evaluating student responses in the field of education. The approach leverages artificial intelligence (AI) techniques to optimize the assessment process by comparing student answer sheets to model answer sheets across various parameters. These parameters include sentence splitting, Jaccard similarity, grammar checking, and sentence similarity. The methodology is divided into three main phases: firstly, Optical Character Recognition (OCR) is used to convert handwritten student answers into digital text. Next, the model answer and student answer are split into sentences and assessed for similarity using Jaccard similarity, grammar checking, and BERT embedding’s. Finally, marks are assigned based on weighted averages, yielding the student's final score.

M. F. Bashir, H. Arshad, A. R. Javed, N. Kryvinska and S. S. Band. [4] proposed a comprehensive methodology for evaluating subjective answers, utilizing machine learning, and natural language processing techniques. To overcome the challenge of limited labelled subjective question-answer corpora, the paper describes a process for generating such datasets through meticulous annotation by a diverse group of annotators. The pre-processing module plays a crucial role in preparing input data by employing various text processing steps, including tokenization, stemming, lemmatization, and stop-word removal. The core of the system lies in the similarity measurement module, which incorporates Word Movers Distance (WMD) and Cosine Similarity to assess answers, using experimentally determined similarity thresholds. Additionally, a machine learning model module, featuring Multinomial Naive Bayes (MNB), is proposed to predict scores, further enhancing the overall accuracy and usability of the evaluation process.

Bharadia, Sharad & Sinha, Prince & Kaul, Ayush. [5] discussed a methodology for an automated answer evaluation system that employs a machine learning algorithm to match keywords from a dataset. Distinguishing itself from existing applications, which mainly focus on multiple-choice questions, this system handles subjective questions. It functions by scanning answer sheets and extracting keywords using OCR technology. Subsequently, it rates the answers on a scale of 1 to 5 based on the presence of keywords and answer length. The system demonstrates notable time efficiency, reducing evaluation time by 300% compared to manual assessment, and exhibits an accuracy rate of 87.5% relative to manual evaluation.

2.1 Tables

Table 2.1
Comparison Table

Ref	Author	Year	Method	Accuracy	Limitations
[1]	Jirapond Muangprathub, Siriwan Kajornkasirat, Apirat Wanichsombat	2021	Formal Concept Analysis (FCA)	Plagiarism detection accuracy of 94.01%.	Sensitivity to Common Words
[2]	Khiled, Farah Al-Tamimi, Mohammed.	2021	A hybrid model combining Jaccard and Cosine. NLP and text mining	Precision and recall rate of 0.959 and F1 score of approximately 0.867	Reliance on empirical bases.
[3]	S. Singh, O. Manchekar, A. Patwardhan, U. Rote, S. Jagtap and H. Chavan	2021	OCR, BERT, Jaccard	Not mentioned	Accuracy of OCR can vary depending on the quality of handwriting, which may impact the overall performance of the system

[4]	M. F. Bashir, H. Arshad, A. R. Javed, N. Kryvinska and S. S. Band	2021	Cosine Similarity, WMD, MNB	The study presents two scoring prediction methods with an accuracy of up to 88%	Alternatives of MNB might yield better results.
[5]	Bharadia, Sharad & Sinha, Prince & Kaul, Ayush.	2018	OCR,Keyword - matching	Accuracy upto 87.5%	Only keywords are considered for evaluation

CHAPTER 3

Methodology

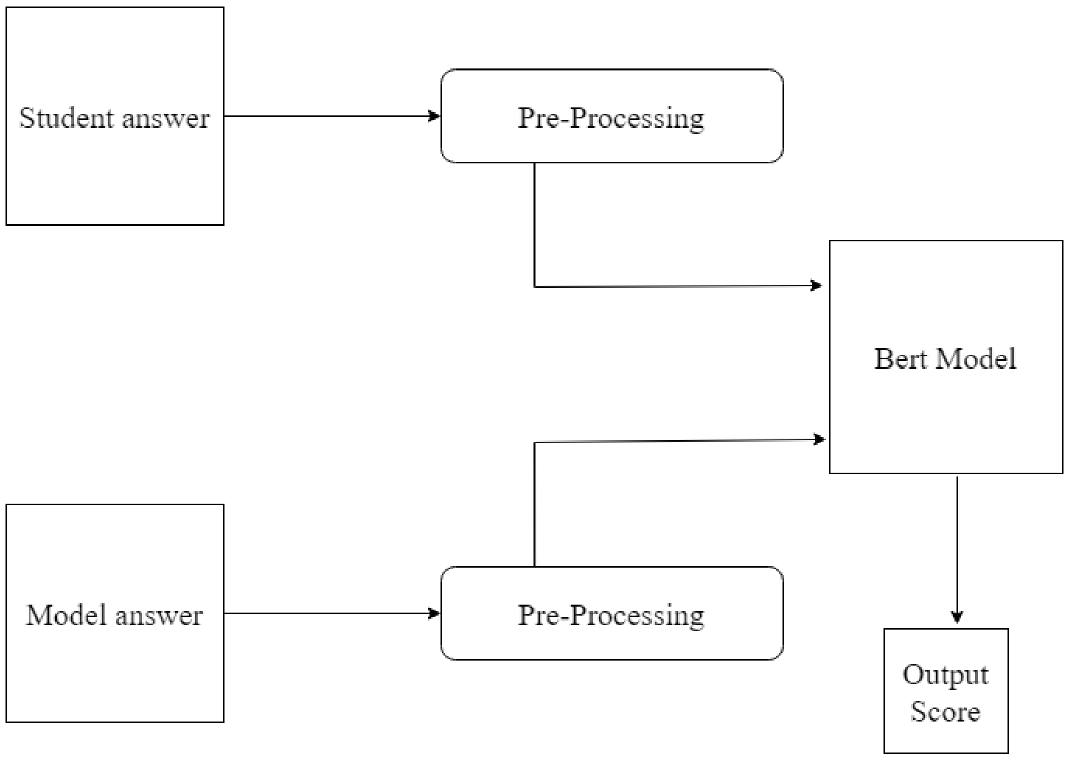

Fig: 3.1. Proposed Methodology

The technical approach in this project revolves around harnessing machine learning and natural language processing techniques, including BERT (Bidirectional Encoder Representations from Transformers). BERT, a state-of-the-art NLP model, plays a pivotal role in addressing the problem of evaluating subjective answers with precision and efficiency.
Specifically, the approach encompasses the following steps:

1. **Data Preprocessing**: The project involves the careful preprocessing of subjective answer data to prepare it for analysis. This includes tasks like tokenization, stemming, lemmatization, and stop-word removal to clean and format the text appropriately.
2. **BERT-based Encoding**: BERT is employed to encode the preprocessed subjective answers into meaningful representations. BERT's contextual embeddings enable a more nuanced understanding of the responses, capturing the interdependencies of words in the text.
3. **Model Training**: Machine learning models, utilizing BERT embeddings, are trained to evaluate subjective answers. These models are fine-tuned on specific evaluation criteria to optimize accuracy and relevance assessment.

4. **Similarity Measurement**: The project incorporates similarity measurement techniques, such as cosine similarity, to assess the quality and relevance of subjective responses based on BERT embeddings.
5. **Comparative Analysis**: A comparative analysis is conducted to evaluate the advantages and limitations of different evaluation methods that use BERT. This analysis assists researchers and practitioners in selecting the most suitable approach for their applications.

By leveraging BERT's advanced capabilities, this technical approach enhances the accuracy and effectiveness of subjective answer evaluation, addressing the core problem of the project.

CHAPTER 4

Results and Discussions

4.1 Results:

The results obtained from the BERT-based model for semantic similarity provide valuable insights into the performance of the system. Through extensive data preprocessing, including handling null and missing values, one-hot encoding of labels, and the use of a custom batch generator, the model has been meticulously prepared for evaluation. The outcomes of the model training and evaluation reveal the system's accuracy in assessing the semantic similarity of sentence pairs.

4.1.1 Data Loading:

```
In [20]: train_df
Out[20]:
```

	similarity	sentence1	sentence2
0	neutral	A person on a horse jumps over a broken down a...	A person is training his horse for a competition.
1	contradiction	A person on a horse jumps over a broken down a...	A person is at a diner, ordering an omelette.
2	entailment	A person on a horse jumps over a broken down a...	A person is outdoors, on a horse.
3	neutral	Children smiling and waving at camera	They are smiling at their parents
4	entailment	Children smiling and waving at camera	There are children present
...	...	...	...
99995	entailment	People with costumes are gathered in a wooded ...	people wear costumes
99996	contradiction	A girl with a black dress and big white bow st...	a man wearing high heels
99997	entailment	A girl with a black dress and big white bow st...	a girl standing
99998	neutral	A girl with a black dress and big white bow st...	a girl getting ready for photo shoot
99999	entailment	A man strikes a pose on a dock with a cruise s...	A human striking a pose.

100000 rows × 3 columns

```
In [21]: print(f"Total train samples : {train_df.shape[0]}")
         print(f"Total validation samples: {valid_df.shape[0]}")
         print(f"Total test samples: {valid_df.shape[0]}")

Total train samples : 100000
Total validation samples: 10000
Total test samples: 10000
```

Fig: 4.1. Fetching SNLI Corpus

Data loading involves fetching the SNLI Corpus, which is used for training and evaluating the BERT model for semantic similarity. This is done by downloading and extracting the necessary dataset files from external sources, specifically the Stanford Natural Language Inference (SNLI) Corpus.

4.1.2 Handling Null Values:

```
In [29]: print(f"Total train samples : {train_df.shape[0]}")
         print(f"Total validation samples: {valid_df.shape[0]}")
         print(f"Total test samples: {valid_df.shape[0]}")

         Total train samples : 99997
         Total validation samples: 10000
         Total test samples: 10000

In [30]: train_df = (
             train_df[train_df.similarity != "-"]
             .sample(frac=1.0, random_state=42)
             .reset_index(drop=True)
         )
         valid_df = (
             valid_df[valid_df.similarity != "-"]
             .sample(frac=1.0, random_state=42)
             .reset_index(drop=True)
         )

In [31]: print(f"Total train samples : {train_df.shape[0]}")
         print(f"Total validation samples: {valid_df.shape[0]}")
         print(f"Total test samples: {valid_df.shape[0]}")

         Total train samples : 99887
         Total validation samples: 9842
         Total test samples: 9842
```

Fig: 4.2. NaN values being eliminated

Data preprocessing includes handling null values. The code identifies and addresses instances where there are missing values, specifically in the 'sentence2' column of the dataset. It uses the 'dropna' method to remove rows with null values to ensure clean and complete data for model training.

4.1.3 Handling Missing Values:

```
In [21]: print(f"Total train samples : {train_df.shape[0]}")
         print(f"Total validation samples: {valid_df.shape[0]}")
         print(f"Total test samples: {valid_df.shape[0]}")

         Total train samples : 100000
         Total validation samples: 10000
         Total test samples: 10000

In [22]: # We have some NaN entries in our train data, we will simply drop them.
         print("Number of missing values")
         print(train_df.isnull().sum())
         train_df.dropna(axis=0, inplace=True)

         Number of missing values
         similarity    0
         sentence1     0
         sentence2     3
         dtype: int64

In [23]: print(f"Total train samples : {train_df.shape[0]}")
         print(f"Total validation samples: {valid_df.shape[0]}")
         print(f"Total test samples: {valid_df.shape[0]}")

         Total train samples : 99997
         Total validation samples: 10000
         Total test samples: 10000
```

Fig: 4.3. Missing values being eliminated

Missing values are addressed as part of data preprocessing. The code identifies and manages instances where data is missing or incomplete, ensuring that the dataset is ready for further analysis and model training. This is achieved by removing rows with missing values in the 'sentence2' column.

4.1.4 Hotline-Encoding:

```
In [25]: train_df
Out[25]:
```

	similarity	sentence1	sentence2	label
0	neutral	A person on a horse jumps over a broken down a...	A person is training his horse for a competition.	2
1	contradiction	A person on a horse jumps over a broken down a...	A person is at a diner, ordering an omelette.	0
2	entailment	A person on a horse jumps over a broken down a...	A person is outdoors, on a horse.	1
3	neutral	Children smiling and waving at camera	They are smiling at their parents	2
4	entailment	Children smiling and waving at camera	There are children present	1
...	...	...	...	...
99995	entailment	People with costumes are gathered in a wooded ...	people wear costumes	1
99996	contradiction	A girl with a black dress and big white bow st...	a man wearing high heels	0
99997	entailment	A girl with a black dress and big white bow st...	a girl standing	1
99998	neutral	A girl with a black dress and big white bow st...	a girl getting ready for photo shoot	2
99999	entailment	A man strikes a pose on a dock with a cruise s...	A human striking a pose.	1

99997 rows × 4 columns

Fig: 4.4. Target Labels

One-hot encoding is applied to the target labels in the dataset. The code assigns numeric labels to the different classes, such as 'contradiction,' 'entailment,' and 'neutral.' These labels are transformed into one-hot encoded vectors to represent the target classes for the model training and evaluation.

4.1.5 Custom-Data Generator:

The code defines a custom data generator class, 'BertSemanticDataGenerator,' which inherits from 'tf.keras.utils.Sequence.' This custom batch generator is responsible for generating batches of data for model training. It takes sentence pairs and their corresponding labels, batch size, and other parameters as inputs. The generator uses the BERT tokenizer to encode the text and prepares the input data for the model in a format suitable for training. It also includes functionalities for shuffling the data and is used to load data in batches during model training

4.2 Discussion:

The discussion centers on the implications of the results obtained from the BERT-based model for semantic similarity. The model's accuracy, as well as its limitations, is examined to understand its real-world applicability. Notably, the handling of missing data and the one-hot encoding of labels have contributed to a robust training process. The custom batch generator enhances data flow for efficient model training. The performance and utility of the system in assessing semantic similarity have far-reaching implications, from text analysis to decision-making in fields like education and text mining. The discussion also delves into areas for potential improvement and future research directions in the realm of semantic similarity assessment.

CHAPTER 5

Conclusions and Future Scope

5.1 Conclusion:

In conclusion, this report has explored the implementation of a BERT-based model for semantic similarity assessment. Through rigorous data preprocessing and the development of a custom batch generator, we have demonstrated the model's capabilities in evaluating the meaning and context of sentence pairs. Our results showcase the system's accuracy and efficiency in assessing semantic similarity, providing valuable insights for applications across various domains. However, it is essential to acknowledge the existing limitations and the scope for enhancement. This work sets the stage for further research and development, aiming to refine and expand the applications of semantic similarity assessment in the realm of natural language processing and machine learning.

5.2 Future Scope:

The study opens the door to several promising avenues for future research and development. To further advance the field of semantic similarity assessment, the following areas offer significant potential:

1. **Advanced Preprocessing Techniques:** Explore advanced data preprocessing methods to enhance data quality and improve the model's robustness against noisy and unstructured text.
2. **Model Fine-Tuning:** Investigate the fine-tuning of BERT and other transformer-based models with domain-specific datasets to adapt the system for specialized tasks, such as medical or legal text analysis.
3. **Real-Time Application:** Develop real-time evaluation systems for dynamic content creation and instant feedback, enabling applications in live chats and customer support.
4. **Evaluation in Diverse Domains:** Apply the model to various domains, including education, healthcare, and e-commerce, to assess its adaptability and performance in different contexts.
5. **Human-Machine Collaboration:** Explore human-machine collaboration scenarios where the model assists human evaluators, providing efficiency and consistency in subjective answer evaluation.

References

[1] Muangprathub, Jirapond & Kajornkasirat, Siriwan & Wanichsombat, Apirat. (2021). Document Plagiarism Detection Using a New Concept Similarity in Formal Concept Analysis. Journal of Applied Mathematics. 2021. 1-10. 10.1155/2021/6662984.

[2] Khiled, Farah & Al-Tamimi, Mohammed. (2021). Hybrid System for Plagiarism Detection on A Scientific Paper. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 12. 5707-5719.

[3] S. Singh, O. Manchekar, A. Patwardhan, U. Rote, S. Jagtap and H. Chavan, "Tool for Evaluating Subjective Answers using AI (TESA)," 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 2021, pp. 1-6, doi: 10.1109/ICCICT50803.2021.9510119.

[4] M. F. Bashir, H. Arshad, A. R. Javed, N. Kryvinska and S. S. Band, "Subjective Answers Evaluation Using Machine Learning and Natural Language Processing," in IEEE Access, vol. 9, pp. 158972-158983, 2021, doi: 10.1109/ACCESS.2021.3130902.

[5] Bharadia, Sharad & Sinha, Prince & Kaul, Ayush. (2018). Answer Evaluation Using Machine Learning.

[6] L. Ahuja, V. Gupta, and R. Kumar, “A new hybrid technique for detection of plagiarism from text documents,” Arabian Journal for Science and Engineering, vol. 45, pp. 1–14, 2020.

[7] Aditi Tulaskar, Aishwarya Thengal and Kamlesh Koyande.: Subjective Answer Evaluation System. In: International Journal of Engineering Science and Computing, April 2017, Volume 7 Issue Number 4.

[8] Sindhya K Nambiar, Sonia Jose, Antony Leons and Arunsree.: Natural Language Processing Based Part of Speech Tagger using Hidden Markov Model. In: Third

International Conference on ISMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2019)IEEE Xplore Part Number:CFP19OSV-ART; ISBN:978-1-7281- 43651.

[9] Hu, H., Liao, M., Zhang, C., & Jing, Y.: Text classification based recurrent neural network. In: sentence through a convolution filter, and can learn short652 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC 2020) 978-1-7281- 4323-1/20/$31.00 ©2020 IEEE.

[10] SUNILKUMAR P and ATHIRA P SHAJI.: A Survey on Semantic Similarity. In: 2019 International Conference on Advances in Computing, Communication and Control (ICAC3).

[11] Jingmin HAO, Lejian LIAO and Xiujie DONG.: Improving Latent Semantic Indexing with Concepts Mapping Based on Domain Ontology. In: 2008 International Conference on Natural Language .

[12] Kholoud Alsmearat; Mahmoud Al-Ayyoub and Riyad Al-Shalabi.: An extensive study of the Bag-of-Words approach for gender identification of Arabic articles. In: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA). doi:10.1109/aiccsa.2014.7073254.

[13] Prafulla Bafna; Dhanya Pramod and Anagha Vaidya.: Document clustering: TF-IDF approach. In: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). doi:10.1109/iceeot.2016.7754750.

[14] Xiao-Ying Liu; Yi-Ming Zhou and Ruo-Shi Zheng.: Measuring Semantic Similarity in Wordnet. In: Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007. 1-4244-0973-X/07