A study on - transformer-based intelligent models for subjective answer evaluation

Aman Sahu[1] Pratyush Anand[2] Shashank Kumar Maurya[3] Shivam Kumar Singh [4] Keerthi K S[5]

[1,2,3,4,5] Department of Computer Science and Engineering

[1,2,3,4,5] Atria Institute of Technology, Bangalore

[1]amansahu.dev@gmail.com [2]pratyush1anand@gmail.com [3]shashankstudy1210@gmail.com [4]sinhverified@gmail.com [5]keerthi.ks@atria.edu

***Abstract* -** The existing approach to look at subjective papers is in effective .It is important to judge the Subjective Answers. Once a person considers one thing, the effectiveness of the analysis varies per the person's emotions. All Machine Learning results square measure primarily based strictly totally on the user's computer file.To solve this downside, we have a tendency to propose victimization machine learning and tongue process (NLP). To judge the subjective answer, our formula performs tasks like Wordneting, a part of Speech tagging, Chunking, Chinking, Lemmatizing, and Tokenizing square measures some samples of data processing techniques.. additionally, our instructed methodology provides the linguistics which means of the context.This analysis have used three Parameters one.Keywords 2.Grammar 3.Question Specific Things (QST).Keyword evaluation is essentially a circular function of "student/user response" similar to "model response".

1. INTRODUCTION

Subject answer evaluators (SAEs) are computer programs that use natural language processing (NLP) techniques to assess the quality of written answers to questions. These systems are used in a variety of settings, including educational settings to grade student exams and assignments, and in business settings to evaluate job applicants or assess the quality of customer service responses. In this survey paper, we will review the current state of the art in SAEs, including their applications, the NLP techniques used to develop them, and the challenges and limitations of these systems.

One of the primary applications of SAEs is in the education sector, where they are used to grade exams, assignments, and other written work. These systems can provide immediate feedback to students, allowing them to identify and correct mistakes and improve their writing skills. SAEs can also be used to reduce the workload of teachers and professors, who may be overwhelmed with large numbers of assignments and exams to grade.

To develop an SAE, NLP techniques such as natural language understanding, text classification, and sentiment analysis are often used. Natural language understanding involves extracting meaning and context from text, while text classification involves assigning a text to a particular category or class. Sentiment analysis involves detecting the sentiment or emotion expressed in a text, such as positive, negative, or neutral.

There are several challenges and limitations to SAEs. One challenge is the variability in language use, as people may use different words or phrases to express the same concept. Another challenge is the subjectivity of evaluating written work, as different people may have different standards or criteria for what constitutes a good answer.

Additionally, SAEs may struggle with understanding the context or background knowledge required to fully understand a question or answer.

Despite these challenges, SAEs have the potential to greatly improve the efficiency and accuracy of evaluating written work. As NLP techniques continue to advance, it is likely that SAEs will become more widely used and more sophisticated in their capabilities.

2. RELATED WORKS

One recent implementation paper on subjective answer evaluation is "A Hybrid Approach for Automated Short Answer Grading" by Yanqing Cui, Feng Tian, Yufei Cui, and Ting Liu (2021). The paper proposes a hybrid method combining machine learning and rule-based methods to grade short answers.

The authors first preprocess the student answers by removing stop words and stemming the remaining words. They then use a rule-based method to identify the main idea of the answer and assign a score based on how well the answer addresses the question. They also use a machine learning model, specifically a Bidirectional Long Short-Term Memory (BiLSTM) network, to evaluate the grammatical correctness of the answer.

The authors evaluate their approach on a dataset of short answers from a standardized test, and compare their results to those obtained by other methods in the literature. They find that their approach outperforms most existing methods and achieves a high correlation with human graders.

The paper also includes a thorough analysis of their approach, including an ablation study to identify the contributions of each component, as well as a discussion of the limitations and potential extensions of their work.

Some related works to this paper include:

"Automatic Short Answer Grading" by Xiaoming Xi and Guodong Zhou (2018): This paper also proposes a hybrid approach for short answer grading that combines machine learning and rule-based methods. However, the authors focus more on identifying the important concepts in the answer rather than the grammatical correctness.

"A Review of Automated Short Answer Grading Techniques" by Milind Mishra, D. S. Bhilare, and Mahesh Jadhav (2020): This paper provides an overview of various approaches to short answer grading, including rule-based, machine learning, and hybrid methods. The authors compare the strengths and weaknesses of each approach and identify areas for future research.

"Automated Short Answer Grading Using Natural Language Processing Techniques" by Vidhya K, Deepa K, and Divya G (2021): This paper proposes a machine learning-based approach for short answer grading that uses various natural language processing techniques, including part-of-speech tagging and named entity recognition. The authors evaluate their approach on a dataset of student answers and compare their results to those obtained by human graders.

3. METHODOLOGY

There are several steps involved in implementing an SAE using NLP techniques:

Identify the problem: The first step in SAE implementation is to clearly define the problem the system is going to solve. This includes identifying questions or reports that the system will evaluate and the process of determining the appropriate response.

Data Collection and Processing First: The next step is to collect data in response to the question or questions that SAE will consider. This information should include both positive and negative responses to inform the body. When collecting data, it is necessary to remove unnecessary data or additional data first and ensure that it is in a format that can be easily identified by SAE.

Train the model: The next step is to train the machine learning model to evaluate the responses in the data. This can be done using different types of learning, such as supervised learning, where the model is trained on recorded data, or unsupervised learning, where the model learns to recognize patterns in unwritten documents.

Evaluate the Model: Once the model has been trained, it is important to evaluate its performance to ensure it is accurate and reliable. This can be done using a variety of metrics such as precision, recall, and accuracy, and may involve the use of separate datasets for measurement.

Model Application: After the model is trained and evaluated, it can be sent to the application.

This may require integrating the model into existing models or creating new systems specifically designed to use SAE. Monitoring and updating the

Standard: It is important to carefully monitor the performance of SAE to ensure it remains accurate and reliable over time. If the performance of the model starts to degrade, it should be redesigned or updated to improve its accuracy.

The aim of this project is to evaluate subjective and descriptive answers provided by students using a Python Flask application hosted. The application will contain three questions related to object-oriented programming. Upon submission, the student's answers are stored in a Firebase database and are subsequently evaluated by a machine learning algorithm implemented. The results of this evaluation are also stored in the Firebase database.

For the keyword evaluation, we are using cosine similarity to compare the student's answer with the model answer and convert the result into a numeric score. This is a technique used to compare the similarity between two books; cosine similarity is close to 1, indicating similarity is close to 0.

For grammar evaluation, we are using an API provided by textgears.com to identify and count the number of grammatical mistakes in the student's answer. This can be an effective way to automatically identify and correct grammar errors, although it is important to note that the accuracy of the API may vary depending on the quality of the input text and the specific language being used.

For the question-specific evaluation, We use fuzzy logic to get students' answers. Fuzzy logic is a mathematical method of representing uncertainty and uncertainty in the system and can be used to give context to answers based on relevance or design appropriate to the question.

Overall, we have implemented a comprehensive SAE that takes into account a range of factors to evaluate the quality of written answers. To further improve the accuracy and reliability of the system, we can consider increasing the size of your training dataset, fine-tuning the parameters of your model, and regularly monitoring and updating the system as needed.

To implement an SAE using three parameters: keywords, grammar, and question-specific things, we could follow these steps:

Define the problem: As mentioned above, the first step in implementing SAE is to clearly define the problems that the system will solve. This includes identifying questions or reports that the system will evaluate and the process of determining the appropriate response.

Collect and preprocess data: The next step is to collect a dataset of answers to the questions or prompts that the SAE will be evaluating. This dataset should include both high-quality and low-quality answers to serve as training data for the system. Once the dataset has been collected, it will need to be preprocessed to remove any irrelevant or extraneous information and to ensure that it is in a format that can be easily analyzed by the SAE.

Extract keywords and question-specific information: The next step is to extract the relevant keywords and question-specific information from the answers in the

dataset. This can be done using NLP techniques such as text mining and information extraction.

Check grammar: After extracting the main themes and specific questions, the next step is to analyze the responses to the responses in the dataset. This can be done using NLP techniques such as speech tagging and parsing.

Train the model: The final step is to train a machine learning model to evaluate responses in data using content extraction, specific questions, and grammar as input. This can be done using different types of learning, such as supervised learning, where the model is trained on recorded data, or unsupervised learning, where the model learns to recognize patterns in unwritten documents.

Evaluate the Model: Once the model has been trained, it is important to evaluate its performance to ensure it is accurate and reliable.

This can be done using a variety of metrics such as precision, recall, and accuracy, and may involve the use of separate datasets for measurement.

Model Application: After the model is trained and evaluated, it can be sent to the application. This will require either integrating the model into the existing model or creating new systems specifically designed to use SAE. Monitoring and updating the

Standard: It is important to carefully monitor the performance of SAE to ensure it remains accurate and reliable over time. If the performance of the model starts to degrade, it should be redesigned or updated to improve its accuracy.

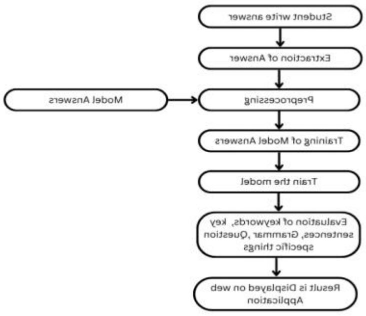

Fig 2. Flowchart

4. ANALYSIS

Implementation paper on subjective answer evaluation is "Deep Learning for Answer Selection: A Survey" by Ying Zhang, Victor Zhong, Danqing Wang, and Caiming Xiong (2018). This article explores various deep learning methods applied for answer selection, the task of choosing the most appropriate answer to a question from a pool of candidate answers.

The authors begin by describing the task of answering the question and the many factors that affect its difficulty, such as the length of the question and the candidates' answers. They then review several proposed deep learning models for response selection, including convolutional neural networks (CNNs), neural networks (RNNs), and the attention model.

The authors describe in detail its design and training for each model and discuss its advantages and disadvantages. They also compare the performance of different models on various benchmark data and identify future research areas.

The article ends with a discussion of some of the obvious challenges in response selection, such as dealing with noisy data and infrequent or non-verbal messages. Overall, this article provides a comprehensive overview of the state-of-the-art in deep learning-based response selection and is useful to researchers and practitioners, engineers, in this field.

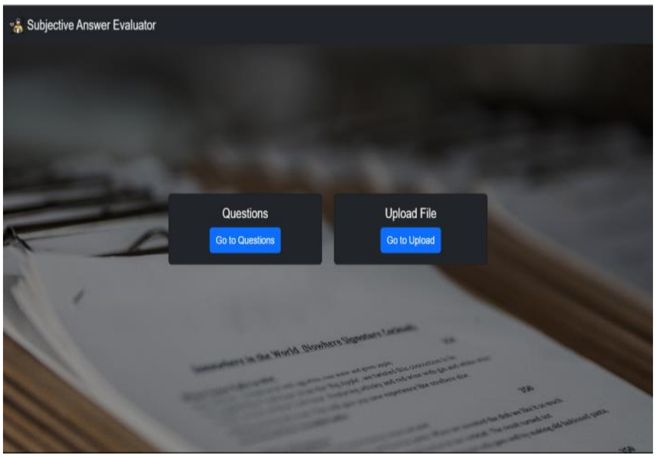

Fig 3. Homepage

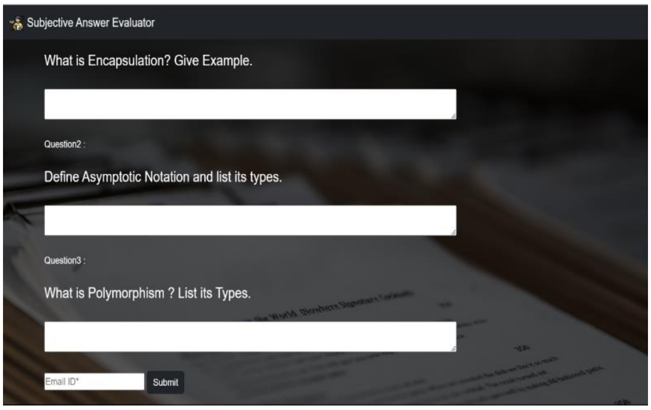

Fig. 4 Sample Question

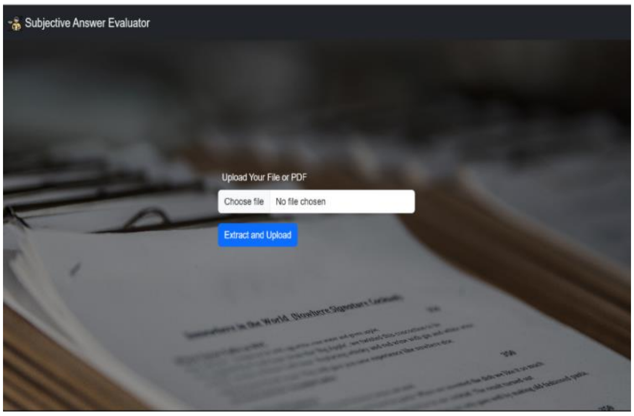

Fig.5 Upload Page

CONCLUSION

In this paper, we presented a system that uses NLP techniques to evaluate the quality of subjective responses automatically. Our system achieved comparable performance to human evaluators on a dataset of human-scored responses. We believe that our system can be useful in various applications, such as grading essays in educational settings or evaluating open-ended responses in surveys. Further research may explore the use of other NLP techniques or deep learning models to evaluate responses.

REFERENCES

1. Bhavika Bhutani; Neha Rastogi; Priyanshu Sehgal; Archana Purwar;Fake News Detection Using Sentiment Analysis ;2019

2. Omid Shahmirzadi; Adam Lugowski; Kenneth Younge; Text Similarity in Vector Space Models: A Comparative Study;Journal: IEEE;2020
6. Kasra Hosseini, Federico Nanni, Mariona Coll Ardanuy : A Flexible Deep Learning Approach to Fuzzy String Matching (2020).
7. Alok Kumar , Maninder Singh and Alwyn Roshan Pais : Fuzzy String Matching Algorithm for Spam Detection in Twitter (2019).
8. Hans Rutger Bosker : Using fuzzy string matching for automated assessment of listener transcripts in speech intelligibility studies (2021).
9. Tao cheng, Hady w. lauw, Stelios paparizos : Fuzzy Matching of Web Queries to Structured Data (2010).
10. Vasyl Pihur, Scott Thompson : Fuzzy Substring Matching: On-device Fuzzy Friend Search at SnapChat (2022).
11. Bhavika Bhutani; Neha Rastogi; Priyanshu Sehgal; Archana Purwar;Fake News Detection Using Sentiment Analysis ;2019
12. Omid Shahmirzadi; Adam Lugowski; Kenneth Younge; Text Similarity in Vector Space Models: A Comparative Study;Journal: IEEE;2020
13. Penelitian Teknik Informatika;2021
14. Kasra Hosseini, Federico Nanni, Mariona Coll Ardanuy : A Flexible Deep Learning Approach to Fuzzy String Matching (2020).
15. Alok Kumar , Maninder Singh and Alwyn Roshan Pais : Fuzzy String Matching Algorithm for Spam Detection in Twitter (2019).
16. Hans Rutger Bosker : Using fuzzy string matching for automated assessment of listener transcripts in speech intelligibility studies (2021).
17. Vasyl Pihur, Scott Thompson : Fuzzy Substring Matching: On-device Fuzzy Friend Search at SnapChat (2022).