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 ineffective .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) .Evaluation of Keywords primarily based circular function Similarity of "student's/user's answer" with "model answer".

Keywords-- Cosine similarity, bag of words (BoW)(frequency),Rule-based algorithm,NLP,Levenshtein distance,Naive bayes

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. LITERATURE SURVEY

[1] Deep learning is a type of machine learning that uses artificial neural networks with many layers of processing units to learn and make decisions based on data. In the field of text classification, deep learning techniques have been shown to be effective in a variety of applications, including sentiment analysis, spam detection, and topic classification. In this paper, the authors review the current state of the art in deep learning-based text classification, including the various types of deep learning models that have been used and the challenges and limitations of these approaches. They also provide an overview of the different evaluation metrics that have been used to assess the performance of deep learning models for text classification and discuss the potential applications of these models in various domains.

The authors conclude that deep learning-based text classification has achieved significant success in a variety of applications and is likely to continue to be a major area of research and development in the field of natural language

processing. However, they also note that there are still challenges and limitations to be addressed, such as the need for large amounts of labeled training data and the potential for overfitting, and that further research is needed to fully realize the potential of deep learning for text classification.
[2] The authors propose a method for detecting fake news using sentiment analysis. The approach involves analyzing the sentiment of the language used in a news article and comparing it to the sentiment of the language used in a fact-checking article about the same news event. The authors trained a machine learning model on a dataset of news articles and fact-checking articles, and used the model to predict the likelihood that a news article was fake. The results showed that the proposed method was effective at detecting fake news, with an accuracy of over 85%. The authors also found that the use of additional features, such as the structure of the article and the credibility of the source, improved the performance of the model. Overall, the paper suggests that sentiment analysis could be a useful tool for detecting fake news and combating its spread.
The authors of the paper conducted experiments to evaluate the performance of their proposed fake news detection method using sentiment analysis. They used a dataset of real and fake news articles and applied machine learning techniques to train a classifier to distinguish between the two. The classifier was trained using a combination of the sentiment of the language used in the articles and the sentiment of the comments about the articles on social media.
The results of the experiments showed that the classifier was able to accurately identify fake news with a high degree of accuracy. The authors also conducted additional experiments to test the robustness of the classifier, including using different machine learning algorithms and varying the size of the training dataset. They found that the classifier performed well across all of the different conditions.
Overall, the authors conclude that sentiment analysis can be a useful tool for detecting fake news and that their proposed method shows promise for identifying fake news in a practical setting. However, they also acknowledge that further research is needed to refine and improve the approach, including exploring other features that may be useful for identifying fake news and examining the performance of the method on different types of news content.
There are many different approaches that have been proposed for detecting fake news, and different methods may be more or less effective depending on the specific characteristics of the news content and the context in which it is being shared. Some common approaches for detecting fake news include:

1. Fact-checking: This involves verifying the accuracy of the information presented in the news article by checking it against reliable sources.
2. Analyzing the source: The credibility and reputation of the source of the news article can be a good indicator of whether it is likely to be fake.
3. Analyzing the language: The use of a certain language or writing style may be indicative of fake news. For example, articles that use sensational or extreme language or that contain numerous typos or errors may be more likely to be fake.
4. Analyzing the content: The content of the article can be analyzed for signs that it is fake. For example, articles that contain conspiracy theories or that present highly unlikely or impossible scenarios may be more likely to be fake.
5. Analyzing the spread of the article: The way in which an article is shared and disseminated can be an indicator of its credibility. For example, articles that are shared widely on social media without being fact-checked may be more likely to be fake.

Overall, it is important to be cautious when evaluating the credibility of news articles and to use a combination of different approaches and techniques to help identify fake news.
[3] Vector space models (VSMs) are a widely used approach for representing and comparing texts in the field of information retrieval. In this paper, the authors perform a comparative study of several different methods for measuring text similarity in VSMs. The methods studied include traditional methods such as cosine similarity and Euclidean distance, as well as more recent methods such as word mover's distance (WMD) and fasttext. The authors evaluate the performance of these methods on a variety of tasks, including document retrieval, paraphrase identification, and plagiarism detection. They also compare the computational efficiency of the different methods.
Overall, the authors find that traditional methods such as cosine similarity and Euclidean distance perform well on most tasks, but newer methods such as WMD and fasttext can offer improved performance in certain cases. However, these newer methods come at the cost of increased computational complexity. The authors also find that the choice of similarity measure can have a significant impact on the performance of a text similarity system.
[4] The use of automated essay scoring (AES) systems in education has increased in recent years, as they can save time and resources in the assessment of student writing. However, accurately evaluating the similarity between essays is an important challenge in AES systems. In this paper, the authors investigate the use of cosine similarity as a measure of essay similarity in an AES system.
Cosine similarity is a measure of similarity between two vectors that takes into account the angle between them rather than their magnitude. It is often used in information retrieval and natural language processing tasks. The authors apply cosine similarity to a dataset of student essays and compare its performance to other similarity measures, including Euclidean distance and Pearson correlation coefficient. They find that cosine similarity performs well on the task of essay similarity assessment, with an average error rate of 8.2%.
[5] Plagiarism, or the act of copying or closely paraphrasing the work of others without proper attribution, is a serious issue in academia. In this paper, the authors investigate the use of cosine similarity, a measure of similarity between vectors, as a method for detecting plagiarism in students' theses.
The authors first preprocess the theses by converting them into vectors using the term frequency-inverse document

frequency (TF-IDF) method. They then calculate the cosine similarity between pairs of vectors, and use this measure to identify pairs of theses that are potentially plagiarized. The authors also compare the performance of cosine similarity to other methods, including Euclidean distance and Pearson correlation coefficient, and find that cosine similarity performs the best in terms of accuracy.

[6] "DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching" is a paper that presents a deep learning approach to fuzzy string matching. Fuzzy string matching is the process of comparing two strings and determining how similar they are, even if they are not exactly the same. This can be useful in situations where responses may not match the expected answer exactly, but still contain relevant information. In this paper, the authors propose a deep learning model called DeezyMatch for fuzzy string matching. The model is based on a transformer architecture, which is a type of neural network that is particularly well-suited for processing sequential data such as text. The authors train the model to predict the similarity between two strings by minimizing the mean squared error between the predicted similarity and the ground truth similarity.

[7] The paper presents a fuzzy string matching algorithm for detecting spam in Twitter messages. The algorithm is based on the Levenshtein distance, which calculates the minimum number of insertions, deletions, or substitutions needed to transform one string into another. To use the algorithm for spam detection, the authors pre-process the Twitter messages by removing punctuation, links, and stopwords, and then calculate the Levenshtein distance between each message and a set of spam keywords. If the distance is below a certain threshold, the message is classified as spam. Overall, the results of this study suggest that the fuzzy string matching algorithm presented in the paper is an effective approach for detecting spam in Twitter. However, it should be noted that the specific performance of the algorithm may vary depending on the specific characteristics of the dataset being used.

[8] "Using fuzzy string matching for automated assessment of listener transcripts in speech intelligibility studies" is a paper that presents a method for using fuzzy string matching to evaluate the quality of listener transcripts in speech intelligibility studies. Speech intelligibility studies are research projects that aim to understand how well listeners can understand spoken language. One way to evaluate the results of these studies is to compare the transcriptions of the spoken language made by listeners to the original spoken language, and assess how accurately the transcriptions reflect the original.

[9] "A Fuzzy Approach to Approximate String Matching for Text Retrieval in NLP" is a paper that presents a method for using fuzzy string matching to improve the performance of text retrieval systems in natural language processing (NLP). Text retrieval systems are software programs that are designed to search through large collections of text documents and retrieve specific documents or passages based on user-provided queries. In this paper, the authors propose a fuzzy approach to approximate string matching for text retrieval in NLP. The authors use the Jaccard similarity, which measures the similarity between two sets by calculating the size of the intersection divided by the size of the union of the sets, as a measure of string similarity.

[10] "Fuzzy Matching of Web Queries to Structured Data" is a paper that presents a method for using fuzzy string matching to improve the performance of systems that match web queries to structured data. Structured data refers to data that is organized in a structured format, such as a database, while web queries are requests made by users to search engines or other online information sources. They apply their method to a dataset of web queries and structured data, and compare the results to those obtained using other methods for matching queries to data.

[11] “The Fuzzy Substring Matching: On-device Fuzzy Friend Search at SnapChat “ is a paper that presents a method for using fuzzy string matching to improve the performance of systems that are targeted at finding the right friend to interact with. The friend list is already available for other purposes, such as showing the chat feed, and the latency savings can be significant by avoiding a server round-trip call to substring matching, ranking prefix matches at the top of the result list.

[7] [11]In this paper, The authors describe their efficient and accurate two-step approach to fuzzy search, characterized by a skip-bigram retrieval layer and a novel local Levenshtein distance computation used for final ranking. The Levenshtein distance, as a metric to compute the distance between the search query and a friend name, is not ideal for our on-device friend search use case.

One potential way to relate the approach presented in this paper to our own project is to consider using fuzzy string matching as a method for evaluating the quality of written responses in a survey. We could potentially use a fuzzy string matching algorithm to compare the responses in our survey to the expected answers and determine how similar they are. we could then use this information to evaluate the quality of the responses and assign a score or grade accordingly.

[12] In “GRAMMAR CHECKERS FOR NATURAL

Grammar checkers are software tools that are designed to help users identify and correct grammar errors in their writing. These errors can include issues with punctuation, verb tense, subject-verb agreement, and more. Grammar checkers can be useful for writers of all levels, from students and professionals to casual writers and bloggers.There are various approaches to developing grammar checkers, and the methodologies used can vary depending on the specific goals and intended audience of the tool. Some grammar checkers are designed to be highly comprehensive, able to detect and correct a wide range of grammar errors, while others may focus on specific types of errors or target a specific language or dialect.One key concept in the development of grammar checkers is the use of linguistic rules and patterns to identify and correct errors. These rules and patterns can be derived from a variety of sources, including language dictionaries, grammar guides, and linguistics research.The internals of grammar checkers can also vary depending on the specific implementation. Some grammar checkers may use machine learning algorithms to identify and correct errors, while others may rely on rule-based approaches or a combination of both.In the evaluation of grammar checkers, it is important to consider a range of factors, including the types of errors that the tool is able to detect and correct, its overall accuracy and effectiveness,

and any weaknesses or limitations it may have. It is also important to consider the user experience when evaluating a grammar checker, as a tool that is difficult to use or understand may not be as useful to users.Overall, the development of grammar checkers for unknown languages presents an interesting and challenging opportunity for researchers and developers. By considering the key concepts and approaches outlined above, it is possible to create effective and useful tools for writers in a variety of languages.Several Indian initiatives for popular languages such as Hindi, Bangla, and Urdu were also observed during our survey..

1. Sentence Tokenization: This includes sentence tokenization as well as word segmentation. The sentence is tokenized into words, which are then broken down into constituent morphs and lexical information about the word is populated from the lexicon.
2. Morphological Analysis: This method returns word stems and associated affixes.
3. POS tagging: Assigning the appropriate POS tag to each word (morpheme)
4. Parsing Stage: Using the chosen approach/methodology, checks the syntactic constraints (agreement constraints) between input words and the formation of Hierarchical phrasal/dependency structure of the input sentence. In the event of a failure to flag grammatical errors, provide an auto correction mechanism or present a list of suggestions to the user. [13] "Neural Network Model In this context, "recall" refers to the ability of the system to correctly identify all of the errors in a text, while "precision" refers to the ability of the system to correctly identify only the errors and not flag any correct sentences as errors. The results of the experiment suggest that the version of the system with a beam search size of 12 had a slightly higher recall (i.e. it identified more errors), but that the precision of both systems was adequate.12The text also mentions that the online grammar checker system is based on the Transformer model and uses spaCy tokenization, a spell checker, and the BPE (Byte Pair Encoding) segmentation algorithm. The Transformer model is a type of deep learning model that is commonly used for natural language processing tasks, and spaCy is a popular open-source library for NLP that includes tools for tokenization, part-of-speech tagging, and more. The checking is the realm of natural language
The scope of Natural language processing (NLP) is reading to BPE algorithm is a technique for compressing text data by replacing frequent pairs of characters with a single, special character [14]" Application for Grammar Checking and Correction" Grammar learn a complete language. A lot of work has been done in the development of a grammar checker, but some effort is put into it. Check the general literature. so we have Comprehensive research and analysis of various grammars Review Approaches, Methods, and Keys. We also provide a concept that considers the accuracy of method.
The approaches are mainly classified
(1) rule-based technology,
(2)Machine learning-based methods and
(3) Hybrid technology. A rule-based approach is best for learning Good, but designing rules can be tedious. This but machine learning reduces fatigue. It depends on the type and size of body used. Or, the simplest of these two techniques is hybrid technology.
Additionally, an error classification scheme is presented. This paper will help identify various kinds of errors. The tasks included there are how This Classification Scheme Helps researchers and developer:
(1) Common mistakes If identified, what kind of Errors should be corrected in a targeted manner.
(2) Identification The degree of error helps determine what is causing I need to check the text length to detect errors.
(3) Identifying the reason for invalid text is very helpful in preparing a valid written response test. The task of grammar checking is simplified by all this. Our observation is Various approaches are:(1) none of the popular approaches is perfectly recognizable for all sorts of things.
Error efficient, (2) most popular tools are not.(3)The experimental data utilized in all approaches is different, so it's difficult because the data used in all approaches is different
Match the results.
(4) Addresses most approaches but so many syntax errors and other errors at word level Efforts are made to detect errors at the statement level, that is, the semantic level.
(5) Detection and Modifications of trailing records are not affected by the research area.
All tools analyzed should also have documentation efficient upload in Word or PDF format Perform grammar check and correction. Above In contrast, the proposed application performs real-time grammar checking and correction by accepting input as:
Following are the common grammatical mistakes that are performed by the users:
A. Punctuational mistakes w.r.t punctuation markers viz. Comma, hyphen, punctuation.
B. Constituents' Agreement mistakes: These errors involve mistakes in the agreement between different parts of a sentence, such as the agreement between a noun and verb (i.e. noun-verb agreement), or between a noun and adjective (noun-adjective agreement).
C. Modifier: A modifier is a word or phrase that describes or qualifies another word or phrase in a sentence. Errors involving modifiers can occur when the modifier is misplaced or when it is not clear which word or phrase the modifier is intended to describe.
D. Vague pronominal reference: Pronouns are words that stand in for a noun or noun phrase. Vague pronominal reference occurs when it is not clear which noun or noun phrase a pronoun is referring to.
E. Inappropriate vocabulary choice (incorrect word sense): This type of error occurs when a word is used with the wrong meaning or context.
F. Lack of parallel structure (diversified structures under the same theme): Parallel structure involves using the same grammatical construction for two or more items in a list or series. Lack of parallel structure can make a sentence difficult to understand or may make it sound awkward or confusing.
G. Sentence sprawl (sentence linking-semantic flow, elaboration vs. summarization): Sentence sprawl refers to the use of unnecessarily long or complex sentences that may be difficult to follow or understand.
H. Tense, Aspect, Modality agreement: Tense refers to the time frame in which a verb's action takes place, while aspect

refers to the duration or completion of the action. Modality refers to the degree of certainty or possibility associated with a verb. Agreement errors in tense, aspect, or modality can occur when the verb does not match the time frame or degree of certainty of the sentence.
[15]. “Quality in an Automated Writing Tutor ”Educators, researchers, and others who create and use adaptive educational technology have some of the most difficult difficulties when it comes to determining the best type of feedback to give students, including its timing, techniques, substance, and impact on performance.
Feedback quality and student feedback acceptance are influenced by a variety of subtle aspects.
This study examined feedback in relation to essay writing using AWE (AWE tools may be useful in understanding how participants use AWE and incorporate feedback). Research has already shown that constructive criticism that focuses on writing methods is more successful than criticism that only corrects grammar and mechanics. But educators, learners, and writers instinctively want comments and automatic revisions on these writing elements. This study provides convincing evidence that instruction in writing mechanics and both forms of feedback help pupils.When analyzing the impact of the grammar and spelling checkers on the first draft and the revision, linear mixed effects (LME) models were used to take into consideration possible effects of the essay question or reading ability.
[16].Natural Language Processing (NLP), whose uses vary from language acquisition to proofreading, includes grammar checking as a significant component. Over the past ten years, a lot of effort has gone into developing grammar checking systems.Less effort is put into surveying the body of literature, though. As a result, we give a thorough analysis of English grammar checking methods, emphasizing both their strengths and weaknesses. Additionally, we methodically chose, investigated, and assessed 12 grammar-checking methods. The twelve methods may be divided into three groups: (1) Rule-based techniques, (2) Machine learning-based techniques, and (3) Hybrid techniques.Each technique has advantages and disadvantages. Rule-based strategies are most suited for language learning, but developing rules is a time-consuming effort. We also developed an error categorization method in this study that distinguishes five categories of errors: sentence structure errors, punctuation errors, spelling errors, syntax problems, and semantic errors. These mistakes are further classified. Researchers and developers would benefit from this categorization approach in the following ways: (1) identifying the most common errors would indicate what types of errors should be targeted for correction, (2) identifying the level of the error would indicate what length of text should be examined to detect any error, and (3) identifying the cause of invalid text would aid in the discovery of a solution to write a valid text. This makes the process of grammar checking easier.
Our observations are as follows, based on a thorough examination of several approaches:
(1) No existing technique can detect all sorts of mistakes efficiently;
(2) most tools are not available for study or general usage; and (3) all approaches utilize different experimental data, making comparison difficult.
(4) The majority of techniques have focused on syntax faults and their subtypes, with very few attempts made to detect errors at the sentence and semantic levels.
(5) Detection and correction of run-on sentences is still another unexplored research topic,
[17] The text discusses methods for enhancing sentiment analysis through the use of NLP methods. These strategies include removing stop words and punctuation, weighing the relevance of words in a dataset using term frequency-inverse document frequency (TF-IDF), creating a bag of words, employing bigrams in addition to unigrams, stemming, lemmatization, and part-of-speech (POS) tagging. Lemmatization uses dictionaries to take into consideration the true meanings of words and their morphological links, whereas stemming entails breaking words down to their simplest form. Words in a phrase are given the appropriate parts of speech by POS tagging. These methods can be used to enhance the precision of a sentiment analysis classifier for a dataset.
[18]The term frequency-inverse document frequency (TF-IDF) approach and the multinomial Naive Bayes (MNB) classification algorithm are used to describe a framework for sentiment analysis in this article. By upgrading the classification principles to manage word occurrence dependency and standardize categorization, the framework is intended to enhance the performance of the MNB algorithm in text categorization. A variety of text classification possibilities are offered by the suggested methodology and algorithm, which may be changed in the future to incorporate artificial intelligence for increased accuracy. The MNB algorithm performs well in classifying a lot of text documents since it is quick, straightforward, and simple to use.
[19] This article explains how to identify false news using passive aggressive and term frequency-inverse document frequency (TF-IDF) vectorizer algorithms. The project's goal is to review and contrast current studies on the identification of fake news. These algorithms can be used to determine whether a user-submitted article is genuine or false. However, the research also implies that utilizing an automatic fact-checking model that makes use of a knowledge base may be a more effective tactic and that relying exclusively on trained models of text for fake news identification may not be sufficient in all circumstances. The disadvantage of this strategy is that manual updates are needed to maintain the knowledge base up to date.
[20] a number of solutions to the text data use issues of the Naive Bayes classifier. Several transforms from the information retrieval field have improved the efficiency of Naive Bayes text categorization. For example, the adjustment makes text that closely approaches the power law appear more multinomial.
Training with the complement class solves the issue of uneven training data. By normalizing the classification weights, the Naive Bayes method is better able to handle word occurrence dependencies. We have empirically shown that these modifications significantly improve Naive Bayes' performance on a variety of data types and more closely match the characteristics of bag-of-words textual data. The modified Naive Bayes approach for classifying texts is rapid, easy, and almost cutting edge.
[21] This article explains how to classify text documents

using different distributions using a transfer-learning technique called NBTC (Naive Bayes Transfer Classification). Naive Bayes classifiers provide the foundation of NBTC, which uses the EM method to change an NB model for use with fresh data. The model is initially calibrated using the distribution of the training data, and it is then changed using the distribution of the test data using KL-divergence measurements to reflect the separation between the two distributions. The learning algorithms performed on diverse datasets have revealed that NBTC offers the best results and has high convergence features. The NBTC technique could, however, be improved in a number of ways, including by extending its use beyond text categorization and by developing theoretical measures to increase the precision of the correlation between KL values and trade-off parameters. The trade-off parameters may also be determined by measures other than KL-divergence.

3. CONCLUSION

Following an examination of several research papers on subjective answer evaluator, it is possible to conclude that this technology has great potential for automating the evaluation of subjective answers. These research papers have shed light on various approaches and techniques that can be used to create an effective and reliable subjective answer evaluator.

One of these papers' key findings is that machine learning algorithms, particularly deep learning models, can be trained to evaluate subjective answers accurately. These models can analyse the content of the answer and assign a score based on various criteria such as relevance, coherence, and clarity by leveraging techniques such as natural language processing and sentiment analysis.

Furthermore, these papers show that by using human feedback to refine the model, the performance of the subjective answer evaluator can be improved even further. The model can learn to identify subtle nuances in subjective answers that are difficult to capture using automated techniques alone by incorporating feedback from human evaluators.

Finally, subjective answer evaluator has the potential to transform the way we evaluate subjective answers. While there are still challenges to overcome, the findings of these research papers indicate that this technology is rapidly evolving and could soon become an integral part of the educational and assessment processes.

REFERENCES :

1. Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao;Deep Learning–based Text Classification: A Comprehensive Review; Journal: ACM Journals ;2021
2. Bhavika Bhutani; Neha Rastogi; Priyanshu Sehgal; Archana Purwar;Fake News Detection Using Sentiment Analysis ;2019
3. Omid Shahmirzadi; Adam Lugowski; Kenneth Younge; Text Similarity in Vector Space Models: A Comparative Study;Journal: IEEE;2020
4. Alfirna Rizqi Lahitani; Adhistya Erna Permanasari; Noor Akhmad Setiawan;Cosine similarity to determine similarity measure: Study case in online essay assessment;Journal: IEEE;2016
5. Oppi Anda Resta, Addin Aditya, Febry Eka Purwiantono;Plagiarism Detection in Students' Thesis Using The Cosine Similarity Method;Journal:Jurnal Dan Penelitian Teknik Informatika;2021
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. Krishna Prakash Kalyanathaya, Dr.D. Akila, Dr.G. Suseendren: A Fuzzy Approach to Approximate String Matching for Text Retrieval in NLP (2019).
10. Tao cheng, Hady w. lauw, Stelios paparizos : Fuzzy Matching of Web Queries to Structured Data (2010).
11. Vasyl Pihur, Scott Thompson : Fuzzy Substring Matching: On-device Fuzzy Friend Search at SnapChat (2022).
12. Nivedita Bhirud, Rakesh Bhavsar, Bhausaheb Vyankatrao Pawar a survey of grammar checkers for natural languages Journal: researchgate.net
13. Gang Hao , Senyue Hao A Research on Online Grammar Checker System Based on Neural Network Model Journal: Journal of Physics Conference Series
14. Yash Thakare, Tejas Sridhar, Navanit Srisangkar, Pankaj Vanwari Application for Grammar Checking and Correction Journal : International Research Journal of Engineering and Technology
15. Kathryn S. McCarthy , Rod D. Roscoe , Aaron D. Likens, and Danielle S. McNamara spelling and grammar checkers improve essay quality in a awt Journal : Springer Nature Switzerland AG.
16. Jitendra Singh Thakur, Madhvi Soni A Systematic Review of Automated Grammar Checking in English Language Journal: Jabalpur Engineering College, India
17. Improved Multinomial Naïve Bayes Approach for Sentiment Analysis on Social Media,Journal: SSRN JournalsPublished: April 2019
18. Title: Multinomial Naive Bayes Classification Model for Sentiment;Journal: IJCSNS;Published: March 2019
19. Title: Fake News Detection and Classification using Natural Language Processing
    Journal: IJRASET;Published: June 2022
20. Title: Tackling the Poor Assumptions of Naive Bayes Text Classifiers;Journal: Massachusetts Institute of Technology;
    Published: 21 August 2003

21. Title: Transferring Naive Bayes Classifiers for Text Classification;Journal: AAAI;Published: 2007