A NOVEL APPROACH FOR AUTOMATED ANSWER SCORING USING SEMANTIC ANALYSIS

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Applications

By

Deepender

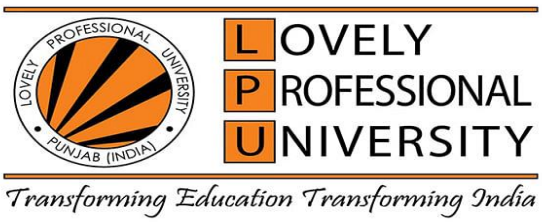
Registration Number: 12021175

Supervised By

Dr. Tarandeep Singh Walia (25153)

Department of Computer Applications (Associate Professor)

Lovely Professional University



LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2026

DECLARATION

I hereby declare that the thesis entitled "A Novel Approach for Automated Answer Scoring Using Semantic Analysis" submitted by me for the Degree of Doctor of Philosophy in Computer Applications is the result of my original and independent research work carried out under the guidance of my Supervisor Dr. Tarandeep Singh Walia, Associate Professor, School of Computer Applications, Lovely Professional University. This work has not been submitted for the award of any degree or fellowship of any other University or Institution.

Deepender
12021175
School of Computer Applications
Lovely Professional University
Phagwara, Punjab-14441, India
Date: 29-09-2025

CERTIFICATE

This is to certify that the thesis entitled "A Novel Approach for Automated Answer Scoring Using Semantic Analysis" submitted by Deepender (12021175) for the award of the degree of Doctor of Philosophy in Computer Applications, Lovely Professional University, is entirely based on the work carried out by him under my supervision and guidance. The work reported, embodies the original work of the candidate and has not been submitted to any other University or Institution for the award of any degree or fellowship, according to the best of my knowledge.

Dr. Tarandeep Singh Walia
Associate Professor
School of Computer Applications
Lovely Professional University
Phagwara, Punjab-144411, India
Date: 29-09-2025

ABSTRACT

AAS is important in educational technology due to its reliable and efficient approach to assessing what students have learned. Accurately evaluating the most detailed answers in languages like Hindi has remained a big challenge for present AAS systems. Because Hindi has complex grammar and meaning, traditional scorers often give lower scores to Hindi answers. In this work, a blended method for computer-aided answer scores is studied, working by incorporating syntax and meaning in state-of-the-art technologies.

The approach suggested in this research is developed to assist with Hindi long text answers. It joins features from traditional machine learning with additional support from advanced neural networks for semantic analysis. XGBoost is used to analyze word length, sentence length, punctuation counts, POS tagging, dependency parsing and grammar rules found in the text. Conversely, semantic features are obtained by using Word2Vec embeddings, scoring coherence, detecting polysemy and Latent Semantic Analysis (LSA). The student's answers are further improved by using the multilingual BERT (mBERT) model which gives contextual embeddings of the meanings that polysemous and ambiguous words might have. Moreover, an RNN-LSTM model with an attention mechanism is used to handle long-term aspects in answers which is why the model is suitable for studying long sequences of words.

This thesis introduces a hybrid approach meant for processing long answers written in Hindi. It makes use of traditional machine learning to collect syntax-related features while employing neural networks for semantic meaning. Some syntactic features such as the number of words, the length of sentences, punctuation marks, part-of-speech tags, dependency parsers and grammar errors, are analyzed and modeled using XGBoost. However, features related to meaning are found using tools such as Word2Vec, coherence analysis, identifying polysemy and Latent Semantic Analysis. Multilingual BERT (mBERT) is applied to improve the word meaning of the student responses by providing meaningful contextual information about polysemous and vague terms. Moreover, a Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM) and attention is used to notice any long-term connections within the answers, so it is well suited for analyzing complex text.

It is possible to combine syntactic and semantic analysis by using a model that combines XGBoost, RNN-LSTM and mBERT. The structural correctness of the student's written work is determined by using XGBoost to analyze its syntax. An attention layer added

to the RNN-LSTM network helps the model pay most attention to the semantic parts of the student response as they are processed by mBERT. The model is also trained on a set of Hindi question-answer pairs that have been given labels, as well as scores for each pair. Thanks to its attention mechanism, the LSTM can give greater importance to meaningful elements in the response that relate well to the reference answer which improves its accuracy in finding errors.

Accuracy, precision, recall and F1-score are used as standard methods to evaluate the proposed hybrid model. By comparing the model to traditional and machine learning-only techniques, it is clear that the model provides better scores for long-text Hindi answers. Because both syntactic and semantic features are blended in the framework, the proposed system is more effective in scoring automated answers. It handles difficulties caused by multiple meanings and unclear words, as well as pays proper attention to the word order in the answers.

Based on the findings, the model does a much better job scoring answers with complex grammar and multiple meanings, especially in Hindi. Having polysemy detection in the model allows it to recognize words with several meanings and interpret them accurately. Besides, the model works well with answers that differ in length and structure which is often seen as a challenge in grading long-text questions.

All in all, this thesis introduces an automated system that scores using features from both syntax and semantics in combination with XGBoost, RNN-LSTM and mBERT. The proposed model is a major advancement in automated scoring for Hindi and other similar languages that have complicated forms and sentence structures. The hybrid model helps overcome the drawbacks of previous AAS systems by providing a reliable approach to scoring responses to open-ended questions. This method is useful for more than just technology in schools and can help analyze the detailed structures and meanings in languages to solve problems in text and speech. We aim to use the model in other languages and carry out more study on how to detect polysemy and model coherence.

ACKNOWLEDGMENT

First and foremost, my special gratitude to almighty God through him, I always get inner strength and positive vibes.

This thesis would not have been possible without the enabled guidance and keen interest of my worthy supervisor Dr. Tarandeep Singh Walia, Associate Professor, Lovely Professional University, Punjab, India. The extensive discussion with him has always made me stay on the right track. He has always been patience and cooperative whenever his guidance and expertise was needed. He helped me not only by sharing his valuable time, but also analytically reviewing publications, reports, and presentations from time to time.

A special thanks to the management of Lovely Professional University for supporting me in the best possible manner and facilitating me in balancing my research work. I am thankful to doctoral committee of LPU for their constructive suggestions and for ensuring the progress of my research work at the correct pace.

This thesis would never have been conceived or borne fruit without the unconditional support of my parents Sh. Daljeet Singh, and Mrs. Rajbala and all other family members. My sincere and special thanks to my wife Dr. Sanju whose unconditional support during all the time is so appreciated. She inspired me in all dimensions of life and instilled confidence in me to complete this journey successfully. I am proud to humbly dedicate this research work to my family, friends and my good wishers.

Words cannot truly express my feelings and appreciations to my friends and colleagues to make my journey comfortable during this work and making it memorable and pleasant. Finally, I thank all the persons who have extended direct and indirect support from time to time and contributed in any of the ways during the execution of this research work.

Deepender

Content

List of Table

List of Figure

LIST OF ABBREVIATIONS

AAS	Automated Answer Scoring
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
mBERT	Multilingual Bidirectional Encoder Representations from Transformers
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
ML	Machine Learning
DL	Deep Learning
POS	Part-of-Speech
XGBoost	eXtreme Gradient Boosting
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
RMSE	Root Mean Square Error
MSE	Mean Squared Error
MAE	Mean Absolute Error
WER	Word Error Rate
API	Application Programming Interface
DF	Data Frame
QA	Question-Answer
SQL	Structured Query Language
N-gram	Contiguous sequence of N items from a text or speech
BOW	Bag of Words
CSV	Comma-Separated Values

Chapter-I

Introduction

In today's world, education is becoming more digital and technologically advanced. The method used to verify students' responses is a significant aspect of this modification. Manual verification of responses takes too much time as well as consistent discrepancies from person to person. Large amount of student answer make it difficult to perform fair and fast verification. An automatic scoring system needs to be developed because students need their answers evaluated both quickly and correctly.

1.1 Manual Scoring

Human evaluators perform manual scoring to assess student responses by utilizing predefined assessment standards or their personal interpretation about the content. The evaluation process requires thorough reading and understanding of individual responses while being detailed in nature. The assessment process requires extensive manual labor because it takes too much time to evaluate responses from many students. Evaluation based on human scoring involves potential inconsistencies because scorers may create different results based on their individual biases. Scoring decisions become less objective when factors like scorer mood and experience and training along with external distractions affect the scoring process. [1] These limitations make it difficult to maintain uniform standard of assessment and poses significant challenges in terms of efficiency, consistency, and scalability.

1.2 Automated Answer Scoring (AAS)

Modern educational scoring technology known as Automated Answer Scoring (AAS) provides effective solutions which reduce human scoring workload and enhance evaluation consistency and reliability. AAS stands for a standardized evaluation process, which automatically assign numerical value to individual responses while removing human involvement [2]. Educational systems along with standardized assessments and multiple industries consider AAS to be a critical element. AAS systems experienced significant improvement in both accuracy and effectiveness throughout past years. The technological method solves manual scoring problems by using pre-scored answers as a foundation. The manually scored responses teach the system about answer and score relationships to create an

automated assessment and scoring process for new responses bound to known patterns. Automated scoring systems eliminate multiple issues that arise when grading is completed manually. The programming of computers allows them to avoid fatigue and remain free from emotional biases while evaluating questions in any response sequence [3]. As a result, computer-assisted scoring can substantially reduce evaluation time while enhancing the reliability and consistency of the scoring process. Moreover, automated scoring can help mitigate issues of subjectivity and inconsistency that often arise in manual evaluation. Reviewing responses in manual scoring and having a personal tie between teacher and learner can change the way scoring happens.

1.3 Natural Language Processing in Automated Scoring

Artificial Intelligence led a transformative impact throughout all industries including education with NLP operating as a fundamental component of these modifications. The ability of machines to process human language naturally through NLP creates an effective system for automatic assessment processing NLP techniques help automated answer scoring systems to evaluate student responses for their grammar, vocabulary, writing style as well as their content alignment [4]. NLP systems have the capability to correctly evaluate both subjective and objective types of questions. The automated answering tools perform answer comparisons between student responses and model answers to verify correctness. The capabilities of NLP enable generated feedback to analyze students' writing for improvement areas based on their performance. The incorporation of NLP in digital assessment creates faster, more accurate and fairer evaluation processes. Overall, student performance evaluation in educational institutions experiences a transformation through the use of NLP technology.

1.4 Automated Answer Scoring Methods

This section explains the proposed method for automated answer scoring that combines rule-based techniques with statistical methods to produce accurate consistent assessment of student responses. Rule-based approaches use a set of predefined expert rules, such as Concept Mapping and Information Extraction, to analyze student answers with the help of linguistic and conceptual features, and statistical methods use data-driven approaches, such as Corpus-Based and Machine Learning-Based, to measure the semantic similarity by converting textual responses into numerical forms[5]. The fundamental sequence of the proposed method appears in Figure 1.1.

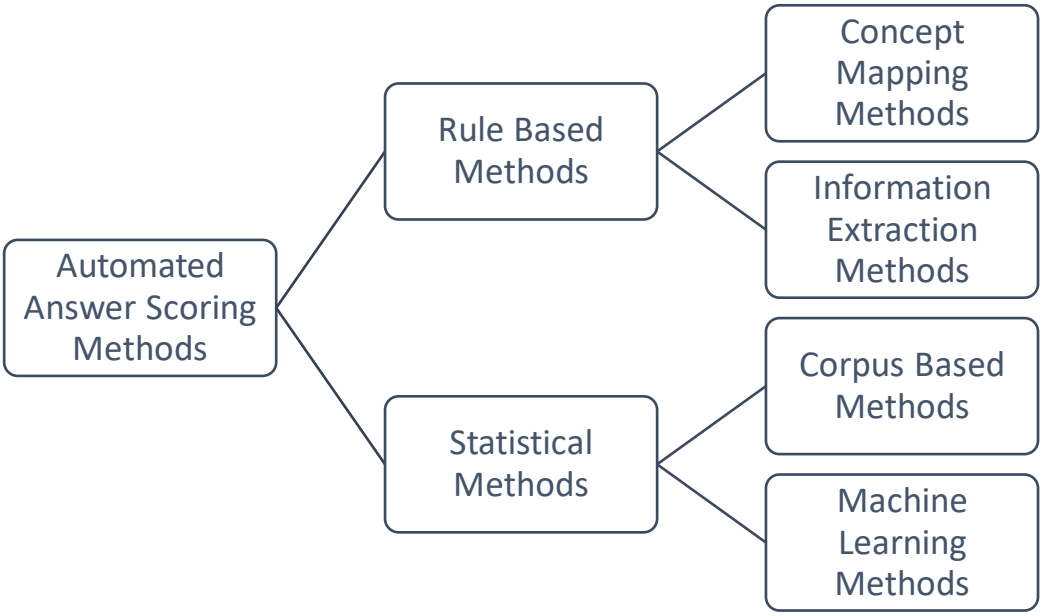


Figure 1.1: Automated Answer Scoring Methods

1.4.1 Rule-based Methods

Rule-based systems offer a fundamental method to implement expert knowledge by using predefined rules which makes text data interpretation and evaluation possible. The rules within answer scoring systems operate to detect vital language elements which include structural elements along with grammatical patterns and word matches. These criteria allow the rule-based technique to generate precise understandable scores for the assessment process [6]. In this context, concept mapping techniques are aimed at leading to defining and correlating main concepts and their relations in student responses with those in reference responses so that the conceptual knowledge is tested not only with the words in the surface. Also, the methods of information extraction are used to get certain entities, relationships or factual elements of responses and conduct a structured comparison with the expected elements of answers. Grammatical structure and student response are measured by using part-of-speech tagging, dependency parsing and by comparing their syntactic trees. The manually created rules enable human examiners to view transparent reasoning behind scoring decisions because they provide clear interpretability. Moreover, this component serves as the foundation for identifying exact or near-exact textual matches, which are crucial for answer evaluation.

1.4.2 Statistical Method

While rule-based methods offer interpretability and control, they may fall short in capturing deeper semantic relationships. Statistical analysis tools measure semantic similarity between student and reference answers to address this problem [7]. The corpus-based techniques make use of large text corpora to obtain the statistical expression of language, including term frequency and inverse document frequency, to allow the comprehension of word significance and contextual relevance in the answers. By comparison, machine learning based approaches automatically learn scoring patterns on labeled data by modeling semantic relationships by a method like: a vector space model and word embedding. These techniques include the use of vector-based models such as TF-IDF, word embedding (e.g., Word2Vec or FastText), and similarity measures like cosine similarity and Jaccard index. The methods transform textual data into numerical representations for systems which enables them to decode contextual meanings and different lexical meanings and associated concepts beyond literal word match patterns. The statistical features together with rule-based methods produce a better solution for answer scoring because they provide stronger data-based evaluation. Together, the integration of rule-based and statistical techniques forms a hybrid model that improves scoring performance by combining the strengths of both interpretability and semantic depth.

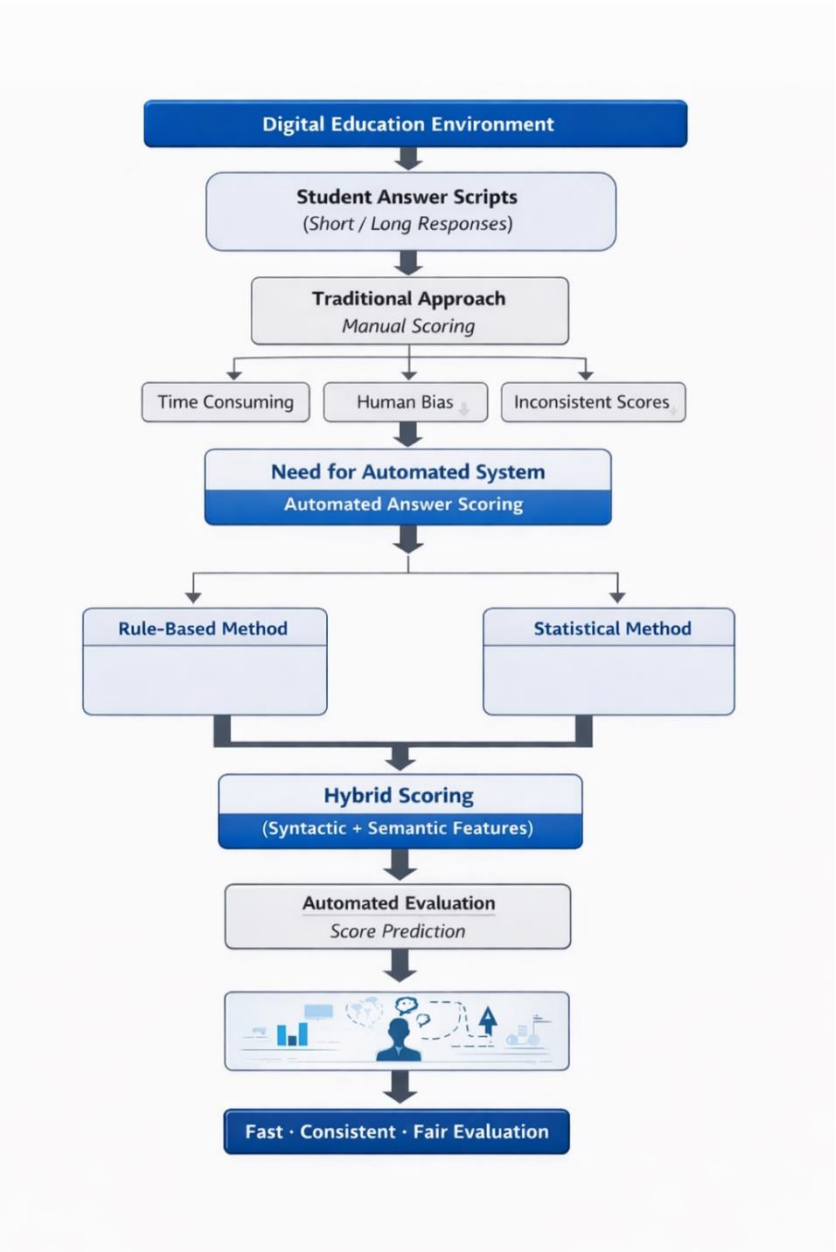


Figure 1.2: Overview of Automated Answer Scoring System

1.5 Background and Research Issues

From a long time, educational assessments uses manual scoring techniques as their traditional evaluation approach. The assessment procedure requires significant manual effort while taking long durations to complete and depends on human judgment thus limiting its potential for widespread usage. The need for better scoring systems tends to the development of Automated Answer Scoring (AAS) systems designed to fix the issues encountered in manual grading processes. The assessment system uses natural

language processing together with machine learning techniques to conduct automatic student response evaluation through predefined standards such as semantic similarity measures and syntactic structure analysis along with answer relevancy check. The automated scoring systems demonstrate effectiveness in English assessments yet struggle with the assessment of non-English languages. Hindi exists as a complex language because it combines rich morphology with flexible syntax and uses Devanagari script and free word ordering which creates barriers for processing. The direct transfer of English model adaptations leads to inadequate results for processing Hindi text.

This investigation aims to resolve important difficulties in automated scoring of Hindi answers. It focuses on the effective preprocessing and representation of Hindi student responses and reference answers to capture both syntactic and semantic features. A key objective is to identify the most relevant and reliable linguistic features and similarity measure for accurately evaluating student answers. This research looks at existing features and models to determine how they can be applied to solve the morpho-syntactic complexity of the Hindi language. Additionally, issues like handling semantic variation, context-dependent interpretations, and maintaining consistency in scoring across diverse question types remain significant challenges. The research objective of this investigation focuses on designing an automated Hindi score evaluation system that incorporates both syntactic and semantic features to assess student answers effectively. The present AAS systems mainly analyze syntactic features independently or semantic features independently thus reducing their overall nuance understanding capabilities in processing student answers. The problem is further aggravated when evaluations contain both short and extensive responses because detail complexity between response lengths is substantial. The research develops a new methodology by merging syntactic and semantic feature extraction strategies to boost the scoring precision for various kinds of responses.

1.6 Objectives of the Proposed Work

The primary objectives of this research are:

1. To study and analyze the resources, like collection of questions and their reference answers, corpus, lexicons etc.
2. To integrate the modules for extracting syntactic and semantic features
3. To propose a hybrid approach for Automated Scoring
4. To validate the performance of Automated Answer Scoring

1.7 Purpose of Summarization

The purpose of summarization is to provide a concise and focused overview of its key aspects while ensuring clarity and accessibility. In the context of AAS for Hindi question-answer sets, summarization achieves several objectives, including presenting the main challenges, proposed solutions, and relevance of the research. By structuring information in a clear and systematic manner, summarization ensures that readers can quickly grasp the scope and importance of the research. It is structured into the following types:

1.7.1 Generic Summarization

Manual scoring is often slow, inconsistent, and subjective. This research introduces an automated scoring system to solve these problems, making the evaluation process faster, more accurate, and fair. By combining syntactic and semantic features, the system handles different types of questions and answers, from short, factual responses to long, descriptive ones. This approach ensures the system is practical for real-world use in education [8].

1.7.2 Domain Specific Summarization

Hindi, being a complex language with free word order and rich morphology, creates unique challenges for automated scoring. To address this, the system uses advanced techniques like POS-tagging and dependency parsing for syntactic analysis, along with semantic tools such as word embedding and similarity measures. These features make the system capable of accurately scoring answers written in Hindi, reflecting the language's diversity and structure.

1.7.3 Query Focused Summarization

The research answers critical questions about the functionality and adaptability of the scoring system. For example, the system efficiently handles mixed-length responses by combining syntactic features like dependency parsing with semantic measures such as TF-IDF. The dataset, specifically designed for Hindi, reflects the linguistic diversity of the language by including questions and answers with varying structures and complexities [9]. By addressing these key aspects, the research ensures the system's relevance and applicability across a wide range of real-world scenarios.

1.8 Motivation for the Research Work

Educational assessments have expanded to such an extent that educators require efficient evaluation systems. The implementation of Automated Answer Scoring (AAS) systems presents an effective strategy to solve current assessment evaluation difficulties through expedited and standardized cost-efficient scoring processes. The majority of automatic AAS systems were designed for English while lacking ability to adapt successfully to Hindi and other morphologically rich languages. The native language of over 40% of India's population speaks Hindi but educational technology research shows little attention towards this language. The complex grammatical structure together with unpredictable word placements and variable morphology in Hindi produces significant barriers for machine systems. The lack of high-quality specific datasets in Hindi hinders the development of performance-based score models. This research is motivated by the need to fill this gap by developing an AAS system tailored for Hindi responses. It combines linguistic feature extraction, semantic analysis, and machine learning to evaluate student responses of varying lengths and complexities, ultimately aiming to improve the fairness and efficiency of assessments in Hindi-medium education.

The main purpose of Hindi dataset collection derives from the need to create a strong assessment-specific corpus which accurately demonstrates diverse question formats along with response length variations in actual testing scenarios. Hindi, as a morphologically rich and syntactically diverse language, presents unique challenges for automated scoring. Existing datasets often fail to capture the linguistic intricacies of Hindi, such as compound words, flexible word order, and context-dependent meanings. By collecting a dataset in Hindi, this research aims to create a resource that represents these nuances, thereby enabling the development of scoring systems better tailored to the language's complexity.

1.9 Challenges of the Study

Despite the advantages of AAS, developing effective scoring algorithms remains a challenging task. Existing systems often struggle with accurately understanding and evaluating complex, nuanced responses, particularly in languages with rich morphological structures like Hindi [9]. Traditional text analysis methods, which rely heavily on keyword matching and syntactic patterns, may not fully capture the meaning and context of responses. Semantic analysis, a branch of natural language processing (NLP), offers a promising approach to address these limitations. By focusing on the meaning and context of words

within a response, semantic analysis can provide a more accurate and nuanced evaluation of textual data. However, integrating semantic analysis into AAS systems requires careful consideration of various factors, including data preprocessing, feature extraction, and algorithm design. Hindi, being morphologically rich, has multiple forms for verbs, nouns, and adjectives. Lemmatization becomes crucial to handle these variations. Many words in Hindi have multiple meanings based on context (e.g., "कल" means both "yesterday" and "tomorrow"). Variations in Devanagari script usage (such as typing errors or informal use of symbols) require careful normalization.

1.10 Structure of the Thesis

The thesis is divided into six main chapters, each designed to build upon the previous one and provide a comprehensive understanding of the research conducted. Chapter 1 introduces the study, outlining its background, significance, objectives, motivation for the research work and Challenges of the study, which set the stage for the exploration of automated answer scoring. Chapter 2 presents a detailed literature review, examining existing methodologies, frameworks, and technologies relevant to automated scoring, highlighting gaps that this research aims to address. Chapter 2 presents a comprehensive review of the existing literature on Automated Answer Scoring (AAS), focusing on the methodologies, frameworks, and technologies that have been proposed and developed over the past few decades. It outlines the evolution of AAS systems, highlights recent advancements, and identifies key research gaps that this study aims to address. Chapter 3 discusses the feature extraction techniques employed in this study, focusing on both syntactic and semantic aspects of student responses. Effective feature extraction plays a crucial role in the performance of automated answer scoring systems, as it enables the model to analyze the structural and contextual elements of textual data. These extracted features serve as the foundation for building effective automated scoring models in subsequent chapters. Chapter 4 outlines the methodology adopted in this research, detailing the data collection process, preprocessing steps, and the application of machine learning and deep learning models for automated answer scoring. It explains how the data is preprocessed followed by the extraction of syntactic features and semantic features. The chapter presents a hybrid approach that combines machine and deep learning techniques to improve the evaluation of student responses in Hindi. Chapter 5 focuses on the results of the study, presenting data analyses and findings that validate the effectiveness of the hybrid scoring approach. The proposed approach demonstrates improved accuracy and reliability in scoring Hindi answers compared

to traditional methods. The integration of semantic analysis significantly enhances the system's ability to evaluate nuanced and contextually rich responses. Experiments on the dataset reveal that the hybrid model outperforms baseline models in capturing both coherence and content relevance. Finally, Chapter 6 concludes the thesis by discussing the implications of the findings, suggesting areas for future research, and summarizing the contributions made to the field of automated assessment. This structured approach ensures a logical flow of information and a thorough examination of the research objectives. In summary, this thesis addresses critical gaps in Automated Answer Scoring by introducing a comprehensive framework that combines syntactic and semantic analysis. By leveraging advanced natural language processing techniques, the study not only improves scoring accuracy for Hindi but also sets a foundation for developing robust AAS systems in other morphologically rich languages.

1.11 Significance of the Study

The implementation of an effective Automated Analysis System for Hindi text leads to transformative effects in educational assessments as well as other domains. In educational settings, such a system can provide timely and consistent feedback to students, enhancing the learning experience. It can also reduce the burden on educators, allowing them to focus on more interactive and personalized teaching methods[10] The integration of semantic analysis into automated answer scoring represents a significant advancement over traditional methods. By focusing on the meaning and context of responses, semantic analysis can provide a more accurate and reliable evaluation, leading to better decision-making and outcomes.

The implementation of an effective Automated Answer Scoring System for Hindi text has the potential to bring transformative changes across educational and other domains. In educational settings, such a system can provide timely, consistent, and objective feedback to students, significantly enhancing the learning experience. It minimizes human bias, ensures fairness in evaluation, and enables large-scale assessments, especially important in diverse and populous regions like India. Moreover, it reduces the workload of educators, allowing them to focus on more interactive, creative, and personalized teaching approaches. Beyond education, automated scoring systems can be valuable in government exams, recruitment processes, and language proficiency testing, where scalability and efficiency are critical. The integration of semantic analysis into such systems marks a substantial improvement over traditional keyword- or pattern-based approaches. By understanding the meaning and context

of responses, semantic techniques provide more nuanced and accurate evaluations, leading to better academic and administrative decision-making. Furthermore, for a linguistically rich and complex language like Hindi, the development of such systems also contributes to the advancement of natural language processing tools and resources in regional languages, promoting digital inclusivity and linguistic diversity

1.12 Summary

The rising significance of Automated Answer Scoring (AAS) is explained in this chapter for contemporary educational systems. It clarifies various weaknesses of manual scoring that include long evaluation duration alongside imperfect consistency as well as evaluation limitations. This section analyzes existing rule-based and statistical techniques before discussing problems related to free-text response assessment. The research calls for improved scoring efficiency in particular for the morphologically complex Hindi language. It investigates efficient preprocessing along with combining syntactic attributes with semantic characteristics for the analysis. It aims to identify reliable similarity measures and linguistic cues tailored to Hindi. The chapter concludes this article by discussing both the motivational and objective reasons that led to the creation of a hybrid scoring model for standardized assessment.

Chapter-II

Review of Literature

2.1 Introduction

The literature review serves as a foundational framework for understanding the advancements and challenges in automated answer scoring, situating this research within the broader context of educational assessment technologies. Looking at various studies closely shows what works well and does not work as well about the ways different scoring systems measure student responses.

This chapter aims to identify gaps in current research and underscore the need for a hybrid approach that integrates multiple evaluation criteria, ultimately laying the groundwork for the subsequent development and validation of a more effective automated scoring system. It examines several approaches, methods and technologies introduced over the years such as older rules and the latest trends in machine learning and deep learning. Particular attention is given to syntactic and semantic analysis techniques and also highlights the limitations of existing systems, especially in evaluating long and context-rich responses in morphologically complex languages like Hindi.

The aim of this literature review is to understand the current state of research in the field of automated answer scoring (AAS), with a focus on identifying the strengths and limitations of existing methodologies. By examining previous studies, this review aims to uncover gaps in current approaches, particularly in handling long, descriptive answers and in applying AAS techniques to morphologically cash rich languages such as Hindi. The review also helps to establish a theoretical foundation for the proposed research by analyzing developments in syntactic and semantic features extraction, and the applications of machine learning and deep learning models. Ultimately, this review informs the design and accomplishment of the proposed scoring system, guiding the selection of appropriate methods and tools. By critically analyzing prior work, this chapter identifies key research gaps and establishes the foundation upon which the current study is built. To organize the information clearly, I will present each study and its findings in this chapter, making it easier to compare and understand the key insights and gaps in the existing research.

2.2 Automated Answer Scoring System

Education systems aim to foster learning by teaching academic subjects and evaluating student understanding. It is crucial to measure students' abilities to tell how good the school and the students are. [11]. These competencies are evaluated by analyzing learning outcomes, where assessments and tests play a central role in identifying students' knowledge and proficiency across subjects [12]–[113]. It is helpful to note that a dependable evaluation system stresses the importance of weak performances as well as reveals strengths that may help a student perform well in future exercises [14].

Teachers can analyze student errors through assessments and use this information to guide classroom instruction and help students learn from their mistakes [15]. Despite being essential for assessing student understanding, manual grading of student responses remains a major challenge in educational systems. Human graders are often required to interpret and infer meaning from diverse student answers, which can lead to inconsistencies and inaccuracies in evaluation [16]. Fatigue, bias, and ordering effects frequently affect grading outcomes, especially when evaluators review a large number of responses over time [17]-[18]. Additionally, the inherently subjective nature of manual grading makes it highly dependent on the individual grader's perceptions, mood, and level of concentration [19]. The problem is further exacerbated by the increasing number of students and the significant amount of time required for accurate assessment [20]–[22]. Studies show that nearly forty percent of a teacher's time is spent on evaluating student work, which reduces the time available for instruction and classroom engagement [23]. Implementing artificial intelligence (AI) offers a valuable solution to the limitations of manual grading.

Artificial intelligence ability to generate innovative and accurate outcomes has made it a prominent tool across various sectors, particularly in education [24]. Among AI technologies, Natural Language Processing (NLP) stands out for its capacity to analyze and understand human language, making it well-suited for addressing challenges in evaluating written responses. One of the key applications of NLP in education is automated answer scoring, where student responses are assessed without human intervention [25], [26]. This approach has gained popularity due to its efficiency, consistency, and objectivity. Automated answer scoring systems are designed to assign relevant scores to student responses based on their content, structure, and meaning [27].

As natural language processing (NLP) techniques evolved, machine learning (ML) and deep learning (DL) models began to upgrade the ability of scoring systems to interpret both the syntactic structure and semantic content of responses. Improved deep networks such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks, make it possible for Automated scoring systems to perform better[28]-[35].Machine learning is a topic that can be applied to automate scoring. Term frequency inverse document frequency (TF-IDF) [36], [37], long short-term memory (LSTM) [38, 39], support vector machines (SVMs) [14], [40], [41], latent semantic analysis [42], k-Nearest Neighbors (KNN) [14], finite state machine [43], and the bagging and boosting [14] have all been employed. Despite these advancements, significant challenges remain—particularly in scoring long, descriptive answers and in extending these techniques to morphologically rich languages like Hindi.

Most existing studies have focused on English datasets, with limited work addressing multilingual AAS. Additionally, many current systems focus primarily on either syntactic or semantic features, but not both, which affects scoring accuracy and fairness. This review highlights the need for hybrid models that combine syntactic and semantic analysis for more accurate assessment. It also underscores the importance of developing AAS systems that can adapt to diverse linguistic structures and educational settings. Although Automated Answer Scoring is a common assessment task across various languages, research in this area has predominantly centered on English, with significantly less attention given to other languages, especially those in the Indic family. Hindi, being the most widely spoken language in India with approximately 528 million native speakers (INDIA 2011), represents a major gap in this research landscape. As millions of students and job applicants write responses in Hindi for academic and professional evaluations, there is a growing urgency to develop automated scoring systems for this language.

The rise of Hindi-speaking telecallers, especially in startups like Apna that serve South Asian markets, further amplifies the need for automation in screening and evaluation processes. However, Hindi NLP remains in its early stages, largely due to a lack of accessible and annotated datasets. Indian datasets available for automated answer scoring include ScAA (Science Answer Assessment) dataset for Hindi and Marathi Language, L3Cube-IndicQuest covering 19 Indic languages and Punjabi ASAG dataset. To address this, some researchers have employed translated English datasets to train large language models for Hindi [44]-[46], enabling progress in neural approaches that bypass traditional feature engineering by learning

hierarchical linguistic representations directly from data. To further support the literature review, Table 1 provides a summary of selected works related to automated answer scoring.

Table 2.1: Overview of Key Research Studies on Automated Answer Scoring

Ref. No.	**Author Name**	**Title**	**Journal/Conference /Year**	**Main findings and Conclusion**
[47]	Rudner & Liang	Automated Essay Scoring Using Bayes Theorem	National Council on Measurement in Education (NCME) (2002)	Demonstrated that Bayesian statistical methods can effectively automate essay scoring with results comparable to human raters.
[48]	Attali & Burstein	Automated Essay Scoring with e-rater V. 2	The Journal of Technology, Learning and Assessment (2006)	Introduced the e-rater V.2 system which uses linguistic features to evaluate essays automatically with reliable scoring accuracy.
[49]	Shermis & Burstein	Handbook of Automated Essay Evaluation	Routledge (2006)	Provided a comprehensive overview of automated essay evaluation systems and highlighted their potential in large-scale assessments.
[50]	Evanini & Wang	Automated Speech Scoring for Non-Native Middle School Students	INTERSPEECH Conference (2013)	Developed an automated speech scoring model capable of evaluating pronunciation and fluency of non-native students.

[51]	Higgins & Heilman	Susceptibility of Automated Scoring to Gaming Behavior	Educational Measurement: Issues and Practice. (2014)	Identified that automated scoring systems can be manipulated by gaming strategies, emphasizing the need for robust evaluation mechanisms.
[52]	Bejar et al.	Examining the Vulnerability of Automated Scoring Systems	Assessing Writing (2014)	Showed that automated scoring systems are vulnerable to construct-irrelevant strategies that may distort evaluation results.
[53]	Burrows et al.	Automatic Short Answer Grading: Trends and Future Directions	International Journal of Artificial Intelligence in Education (2015)	Reviewed existing short answer grading techniques and highlighted emerging NLP-based methods and future research directions.
[54]	Cummins et al.	Using Multi-Task Learning for Automated Essay Scoring	Association for Computational Linguistics (2016)	Demonstrated that multi-task learning improves automated essay scoring performance by learning multiple linguistic features simultaneously.
[55]	Alikanioti s et al.	Automatic Text Scoring Using Neural Networks	arXiv preprint arXiv:1606.04289 (2016)	Proposed neural network-based models that significantly improved automatic text scoring accuracy.

[56]	Amorim & Veloso	Automatic Essay Scoring in Brazilian Portuguese: Challenges and Opportunities	Proceedings of the Student Research Workshop (2017)	Investigated automated essay scoring for Brazilian Portuguese and highlighted challenges in low-resource languages.
[57]	Ajetunmobi et al.	Ontology-Based Approach for Automated Essay Evaluation	2017 International Conference on Computing Networking and Informatics (ICCNI)	Proposed an ontology-based approach to capture semantic relationships for improving essay evaluation accuracy.
[58]	Shi et al.	Exploring the Limits of Automated Scoring Systems in Education	Computational Linguistics Conference (2018)	Examined limitations of automated scoring systems and emphasized the need for improved linguistic and semantic modeling.
[59]	Shadiev et al.	Automated Evaluation for Spoken English Proficiency Tests	Computer Assisted Language Learning (2018)	Developed automated assessment techniques for spoken English proficiency using speech analysis technologies.
[60]	Mohanty et al.	Effectiveness of Automated Essay Scoring on Students' Writing Skills	Educational Data Mining Conference (2019)	Found that automated essay scoring can positively influence students' writing skills through rapid feedback.
[61]	Hussein et al.	A Comprehensive Review of Automated Essay	PeerJ Computer Science(2019)	Presented a comprehensive survey of automated essay scoring systems,

		Scoring Systems		highlighting NLP and machine learning approaches.
[62]	Hargreaves et al.	Text-Based Automated Scoring Systems: Insights and Challenges	British Journal of Educational Technology(2022)	Discussed key challenges in text-based automated scoring including fairness, reliability, and interpretability.
[63]	Choshen et al.	Bias Detection in Automated Essay Scoring Systems	Educational Data Mining Conference(2023)	Identified potential biases in automated essay scoring systems and proposed methods for bias detection.
[64]	Xu et al.	Automated Content Scoring Using Topic Models	ACM SIGKDD Conference on Knowledge Discovery and Data Mining(2023)	Applied topic modeling techniques to improve content-based essay scoring performance.
[65]	Basu et al.	Towards Explainable Automated Essay Scoring Using Attention Mechanisms	Educational Data Mining Conference(2023)	Proposed attention-based neural models to enhance explainability in automated essay scoring systems.
[66]	Kumar et al.	Exploring Bias in Automated Essay Scoring Models	Educational Data Mining Conference (2024)	Analyzed bias issues in modern automated essay scoring models and emphasized fairness-aware evaluation.
[67]	Martin et	A Survey of Automated Writing	AI and Education	Provided a recent survey of automated writing

	al.	Evaluation and Its Challenges	Conference(2024)	evaluation highlighting technological advancements and research challenges.

2.3 Syntactic and Semantic Analysis

AAS systems heavily rely on syntactic and semantic analysis to effectively understand and evaluate student responses. These components allow the system to assess not only the surface-level writing but also the deeper intent and meaning behind it. Syntactic features provide insight into sentence formation, grammatical correctness, and structural organization. Studies have shown that such features significantly influence scoring accuracy, particularly in responses where structure and formality matter. On the other hand, semantic features help capture the underlying meaning and contextual relevance of an answer, going beyond grammar to assess how well a student has understood the content. In this context, the work of Pantulkar Sravanthi and B. Srinivasu (2017) [68] is notable, where they explore semantic similarity between sentences. Their study evaluates three approaches—cosine similarity, path-based similarity, and feature-based methods—and proposes an unsupervised technique to compute sentence-level similarity based on word relationships and contextual meaning. Ajay et al. [69] character count, word count and sentence length are some things the project essay grading tool examines to give grades.

Darwish et al. [70] proposed essay scores that depend on how the sentences and meanings are organized. The features of syntax were extracted from the results of lexical analysis and parsing and the semantic features were predicted using the Tf-IDF vector and gave the final score for the essay. Several approaches to figure out semantic similarity are knowledge-based, corpus-based, and word-embedding-based measures (Gomaa and Fahmy, 2013; Sahu and Bhowmick, 2020). [71]-[72] Corpus-based similarity measures finds how many words are alike according to information retrieve from large corpora (Gomaa and Fahmy, 2013). [71] Latent semantic analysis (LSA) is the most desired corpus-based similarity technique. LSA assumes that words having very close meanings will appear in similar segments of texts. it makes use of a “bag of words” image, wherein the order words are placed is of no importance (Cutrone et al., 2011; Ratna et al., 2013). [73]-[74].Similarity between words using information from semantic networks is measured by knowledge-based approaches (Gomaa and Fahmy, 2013). [71] Mikolov and his colleagues built a word-

embedding model that has been proven efficient in representing the meanings of words in a vector space Mikolov et al., 2013 [75] see also, Bengio et al., 2003; Levy and Goldberg 2014). [76]-[77] Word representation in a vector space reflects the semantics of the words. In continuation of the literature review on syntactic and semantic analysis, a summary of additional relevant studies is presented in the following table for a comparative overview.

Table 2.2: Review of Studies Utilizing Syntactic and Semantic Features in Automated Answer Scoring

Ref. No.	**Author Name**	**Title**	**Journal/Conference /Year**	**Main findings and Conclusion**
[78]	Smith & Anderson	Improving Automated Essay Scoring Through Contextual Embeddings	IEEE Transactions on Learning Technologies(2024)	Demonstrated that contextual embeddings (e.g., transformer-based models) significantly improve essay scoring accuracy by capturing deeper semantic relationships.
[79]	Dong & Zhang	Automatic Feature Extraction for Essay Scoring	EMNLP Conference(2016)	Proposed automatic feature extraction techniques that reduce manual feature engineering while improving essay scoring performance.

[80]	Dasgupta et al.	Enhancing Feature Engineering for Automatic Essay Scoring	NLP Techniques for Educational Applications (2018)	Showed that deep learning–based feature engineering enhances semantic and syntactic representation for more accurate essay scoring.
[81]	Dasgupta et al.	Enhancing Feature Engineering for Automatic Essay Scoring	NLP Techniques for Educational Applications (2018)	Confirmed that advanced feature engineering with deep learning improves automated essay scoring performance and robustness.
[82]	Tan et al.	Improving ESL Automated Essay Scoring with Language-Specific Features	International Conference on Educational Data Mining(2024)	Introduced language-specific linguistic features to improve automated essay scoring for ESL learners.
[83]	Bejar et al.	Does Response Length Affect Scoring in Automated Systems?	ETS Research Report Series (2013)	Found that response length significantly influences

				automated scoring results, highlighting potential bias in scoring models.
[84]	Zhang & Xie	Automated Essay Evaluation with Content, Grammar, and Fluency Focus	Educational Data Mining(2024)	Developed a comprehensive scoring framework evaluating essays based on content relevance, grammar accuracy, and fluency.
[85]	Smith & Anderson	Improving Automated Essay Scoring Through Contextual Embeddings	IEEE Transactions on Learning Technologies(2024)	Showed that contextual embedding techniques enhance semantic understanding and improve scoring reliability.
[86]	Cozma et al.	Essay Scoring Using String Kernels and Word Embeddings	arXiv preprint arXiv:1804.07954 (2018)	Combined string kernels with word embeddings to improve semantic similarity detection in

				essay scoring.
[87]	Deepender &Walia	Semantic Analysis in Automated Essay Evaluation	2019 International Conference on Advances in Computing and Communication	Proposed a semantic analysis approach for automated answer scoring to enhance evaluation accuracy and understanding of student responses.
[88]	Contreras et al.	Automated Essay Scoring Using Ontology and Text Mining Techniques	ICSCEE Conference (2018)	Utilized ontology and text mining techniques to capture semantic relationships for more accurate automated essay evaluation.

2.4 Machine Learning Approach

Machine Learning techniques have significantly advanced the progress of automated scoring systems. The use of machine learning has helped in scoring numerous types of tasks automatically. Many studies on regression, SVMs, decision trees, XGBoost and random forests have been carried out. Nehm, Ha, and Mayfield (2012) [89] studied the possibility of using machine learning to assess how well students explain the idea of evolutionary change. Their scoring program was found to be a powerful and cost-effective tool for assessing. It was found that the scoring program was useful and cost-saving for assessing student knowledge in

a difficult science area. In 2012, ETS made an automated way to score written short responses (Heilman & Madnani, 2013). [90] A machine learning algorithm produces scoring models that maps each test takers' answers to scores. Yannakoudakis, Briscoe and Medlock (2011) [91] used machine learning algorithms to score English as a second or other language examination scripts. Some other investigations (M. Chen & Zechner, 2011; Zechner & Bejar, 2006) [92]-[93] looked into whether it is possible to use machine learning to automatically assess spoken English of people who are not native speakers. Various experiments have been carried out to see which machine learning methods work best in automatic scoring.

In 2012, H. Chen, He, Luo and Li [94] showed that using the ranking SVM approach to scoring essays gets better results than k-NN and MLR. The main idea behind machine learning is to calculate the value of an unknown with the help of the data already available (known as training data). Support vector machines (SVM) Random Forest (RF), k-nearest neighbor regression (KNN) and MLR provide many ways to relate the dependent variables to the independent variables and are among the best choices for identifying how people's scores depend on essay characteristics. Jing Chen Test paper for review in download [95] shows how SVM, RF and k-NN methods run against MLR to see if they can better predict human scores. According to Santos, Verspoor and Nerbonne (2012) [96], the best results in classifying English proficiency level from essays were achieved by logistic model trees (LMT), using logistic regression, among 11 machine learning algorithms that were tested.

Zechner and Bejar's (2006) [93] they have applied both SVM and classification and regression trees (CART; Breiman, Friedman, Olshen, & Stone, 1984)[102] to score spoken responses. They used SVM models for score prediction and CART models for uncovering the role of different features and feature classes in classifying spoken responses. The results suggested that scoring based on SVM yields machine–human agreement that approaches human–human agreement in some cases. M. Chen and Zechner (2011) [92] experimented with two algorithms, ML Rand classification tree, to build scoring models for automatic scoring of speech responses. It was shown that machine learning models were better than decision tree models. The preceding paragraphs looked at how different machine learning techniques performed in analyzing various tasks automatically. The results of machine learning models are not the same for all problems, ways to evaluate them or datasets. [97] In this study [98], the main aim was to create a tool that would assess digital English essays automatically by using the XGBoost classifier. Twelve different scoring areas were created to make sure every student essay was analyzed thoroughly. For this research, the data used came

from argumentative and narrative essays written by students in junior high school which is typical of school assignments. Researchers ensured the effectiveness of the model by performing cross-validation in 5 groups which increases the dependability and usability of the results. Its accuracy of 66.87% shows that the automated grading system is suitable for schools. In research study [99], the main concern was to make essay scoring more effective with advanced techniques for selecting features. The researchers chose Mutual Information Regression to analyze which features are connected to measuring the quality of essays. After that, the research compared how four machine learning models performed when using linguistic data as input. Using this neural network improved the performance and pointed out that the success of this type of scoring depends on well-chosen features. Studies [100] and [101] also looked into using machine and deep learning to study essays written for Hanyu Shuiping Kaoshi (HSK). According to these studies, three different models were created using techniques like Word2Vec and TF-IDF. Most of the models in the study were based on both XGBoost and Deep Neural Networks (DNN). When used with TF-IDF features, the XGBoost model recorded the least MAE at 6.7%.The following table presents an overview of key research studies that have investigate the application of machine learning techniques in automated answer scoring (AAS). Each study focuses on different machine learning models, methodologies, and findings, providing valuable insights into their effectiveness and performance in AAS tasks. The following table offers additional key studies in the domain of machine learning technique, highlighting their main findings.

Table 2.3: Summary of Key Studies Applying Machine Learning Techniques for Answer Evaluation

Ref. No.	**Author Name**	**Title**	**Journal/Conference/ Year**	**Main findings and Conclusion**
[102]	Fonseca et al.	Automatic Grading of Brazilian Essays Using Machine Learning	PROPOR 2018 Conference	Proposed a machine learning–based system for automatic grading of Brazilian essays, demonstrating reliable scoring comparable to human evaluation.
[103]	Devlin	BERT: Pre-training of Deep Bidirectional	arXiv preprint arXiv:1810.04805	Introduced BERT, a deep bidirectional

		Transformers for Language Understanding	(2019)	transformer model that significantly improved contextual language understanding in NLP tasks.
[104]	Jin et al.	BERT-Based Models for Grammatical Error Correction and Essay Scoring	Educational Data Mining (2022)	Applied BERT-based models for grammatical error correction and essay scoring, achieving improved accuracy through contextual language representation.
[105]	Zhang et al.	Improving Automated Scoring Through Machine Learning	International Conference on Artificial Intelligence (2023)	Demonstrated that machine learning techniques enhance automated scoring systems by improving feature learning and evaluation accuracy.
[106]	Chauhan et al.	BERT-Based Automated Answer Grading	International Conference on Artificial Intelligence (2023)	Implemented BERT-based models for automated answer grading, showing improved semantic understanding and scoring reliability.
[107]	Sharma et al.	Multi-Modal Approaches for Improving Automated Essay	AAAI Conference on Artificial Intelligence(2023)	Proposed multi-modal approaches integrating textual and auxiliary features to enhance

		Scoring		automated essay scoring performance.
[108]	Xie et. al	Automated Essay Evaluation with BERT and Transformer Models"	Educational Data Mining(2023)	Evaluated transformer-based models, including BERT, for automated essay evaluation and reported significant improvements in semantic scoring.
[109]	Patel et al.	BERT for Essay Scoring: An Evaluation of the Model's Strengths and Weaknesses	Educational Data Mining(2024)	Analyzed the strengths and limitations of BERT for essay scoring, highlighting its effectiveness in semantic understanding but challenges in bias and interpretability.

Although these models perform well in many scenarios, their reliance on handcrafted features and limited generalization across datasets are notable drawbacks. These issues have paved the way for deep learning methods.

2.5 Deep Learning Approach

Deep learning has brought transformative changes to AAS by enabling automated feature learning and context modeling. Techniques like CNNs, RNNs, LSTMs, and Transformers have been applied to sequence-based answer evaluation. A sequence-to-sequence model for assisting in automated essay scoring was introduced by Dasgupta et al. [110]. At the beginning, they extracted feature vectors from text using GloVe which allows the model to understand the meanings of related words. The vectors were then sent to a CNN layer to allow the network to notice smaller aspects such as phrase patterns and how sentences

are put together. After getting the local features from the CNN, the RNN processed the seq2seq training by spotting how the elements in the essay relate to each other. In the end, an activation layer was added to estimate the score of the essay using the processed sequence of features. They [111] created a bidirectional Long Short-Term Memory (bi-LSTM) model. Words in the input essays were turned into vector form by the Word2Vec library which uses different words' contexts to make these representations. Even though Word2Vec usually uses dense vectors, the model began with one-hot encoding to ensure words were distinct. Process the essay using both forward and backward directions improved accuracy of the model, as it is now able to catch the reasons behind changes in the scoring. In [112], Uto introduced a deep neural network made up of CNN and RNN architectures. First, a lookup table layer turned words into vectors for the starting point of the model. 0-padding was applied to a CNN which enabled it to capture n-gram level features while preserving the edge information. After that, the features were fed into an LSTM model which learned how the essay was structured one sentence after another. Lastly, the sigmoid activation function was used to produce the essay score. As a result of following this architecture, the system could recognize common patterns and also analyze the entire context which made its scoring strong.

K. Surya et al. [113] evaluated several deep learning models for short answer scoring, including character-level CNN, word-level CNN, word-level Bi-LSTM, and BERT. Their findings showed that BERT significantly outperformed the other models, followed by Bi-LSTM, which outperformed both CNN models. They further tested the robustness of these models by introducing controlled perturbations. The Bi-LSTM model showed better tolerance to spelling errors and synonym replacements, while BERT excelled in handling paraphrased responses. Pranjal Patil *et al.* [114] introduced a hybrid model combining sentence modeling and semantic similarity using a Siamese neural network. Each sentence was processed through Bi-LSTM networks enhanced with attention layers, and semantic similarity was computed using a fully connected layer with logistic regression. Their model was tested on a dataset comprising 135 questions in physical sciences with corresponding student responses, demonstrating effective short answer grading. Tuanji Gong et al. [115] proposed an attention-based model integrating pretrained word embedding and a bidirectional RNN with an attention mechanism. Their approach emphasized capturing semantic nuances for improved short answer scoring and was validated through two comparative experiments. The following table summarizes the key contributions of various studies in the application of deep learning techniques for Automated Answer Scoring (AAS).

Table 2.4: Summary of Key Studies Applying Deep Learning Techniques for Answer Evaluation

Ref. No.	Author Name	Title	Journal/Conference/Year	Main findings and Conclusion
[116]	Ding et al.	Neural Automated Essay Scoring: A Survey	Educational Data Mining (2020)	Provided a comprehensive survey of neural network–based automated essay scoring methods and highlighted advancements in deep learning techniques.
[117]	Misgna et al. (2025).	A survey on deep learning-based automated essay scoring and feedback generation	Artificial Intelligence Review (2025)	Reviewed deep learning–based AES systems and concluded that transformer and neural models significantly improve scoring accuracy and feedback generation.
[118]	Cao et al.	Domain-Adaptive Neural Automated Essay Scoring	ACM SIGIR Conference (2019)	Proposed a domain-adaptive neural AES model capable of adapting to

				different essay topics and improving scoring robustness.
[119]	Dong et al.	Attention-Based Recurrent Neural Network for Essay Scoring	CoNLL 2017 Conference	Introduced an attention-based RNN model that captures important textual features to enhance automated essay scoring performance.
[120]	Chen & Li	Relevance-Based Scoring Model Using Hierarchical RNN	IALP 2018 Conference (2018)	Developed a hierarchical RNN model focusing on content relevance to improve automated scoring accuracy.
[121]	Cai	RNN-Based Scoring System for Automatic Essay Grading	International Conference on High Performance Compilation (2019)	Applied recurrent neural networks to automate essay grading, demonstrating improved semantic

				understanding of student responses.
[122]	Chen & Zhou	CNN-Based Essay Scoring and Optimization Research	ICAIBD Conference(2019)	Proposed a CNN-based essay scoring approach that extracts textual features to enhance scoring efficiency.
[123]	Xie& Su	Multi-Faceted Approach to Automated Scoring Using Deep Learning	IEEE Transactions on Learning Technologies (2020)	Developed a multi-faceted deep learning framework combining multiple linguistic features for improved automated scoring.
[124]	Liu & Zhang	Exploring Neural Approaches to Content-Specific Automated Scoring	Neural Information Processing Systems (NIPS)(2022)	Investigated neural approaches for content-specific scoring, improving evaluation of topic relevance in essays.
[125]	Sharma &	Enhanced Feature Selection	International	Enhanced

	Joshi	for Automated Essay Scoring with LSTM	Conference on Advanced Computing and Communication Technologies(2023)	automated essay scoring by integrating feature selection techniques with LSTM neural networks.
[126]	Chen & Liu	Integrating Deep Learning with Automated Essay Scoring for Higher Accuracy	International Conference on Educational Data Science(2024)	Demonstrated that integrating deep learning models significantly improves automated essay scoring accuracy and reliability.
[127]	Wang &Huang	Automated Essay Scoring for Multilingual Essays Using Neural Networks	Association for Computational Linguistics(2024)	Proposed a neural network–based system for multilingual essay scoring, improving evaluation across multiple languages.
[128]	Ahmed & Singh	Analysis of Neural Networks for Scoring College-Level Essays	International Conference on Computational Linguistics(2024)	Analyzed neural network models for scoring college-level essays and confirmed their effectiveness in capturing

				semantic features.
[129]	Kim & Kim	Adversarial Attacks on Neural Automated Essay Scoring Systems	IEEE Transactions on Neural Networks and Learning Systems(2023)	Investigated adversarial attacks on neural AES models and highlighted vulnerabilities in automated scoring systems.
[130]	Jeon et al.	Evaluation of Neural Essay Grading Models	Educational Data Mining Conference (2021)	Evaluated various neural essay grading models and compared their performance in automated assessment tasks.
[131]	Choshen et al.	On the Weaknesses of Neural Automated Essay Scoring Models	Educational Data Mining(2021)	Identified limitations of neural AES models, including bias and lack of robustness in certain scoring scenarios.

2.6 AAS in Low-Resource and Indian Languages

Despite global advancements, AAS research in low-resource languages like Hindi remains limited. Morphological richness, code-mixing, and data scarcity pose significant challenges. Language processing for Indic Languages is happening gradually but is still making progress. A few notable developments include Bhattacharyya (2010) [132], Arora (2020) [133], Kakwani et al. (2020) [134], Ramesh et al. (2021) [135], and more. Desai and Dabhi (2021) [136] present a comprehensive report on the advancements in Hindi NLP while Harish and Rangan (2020) [137] offers an in-depth survey on regional Indic language process ing. Developments in large pre-trained multilingual models like mBERT (Devlin et al. 2018), [138] XLM-RoBERTa (Conneau et al. 2020), [139] Distil mBERT (Sanh et al. 2019), [140] IndicBERT (Kakwani et al. 2020) [134] etc. have featured Hindi plus other regional Indic languages as well. Yet, as was described in the earlier section, finding annotated data remains a problem that many in the Hindi NLP field want to solve.

Table 2.5: Overview of Key Research Studies on Automated Answer Scoring for Indian Language

Ref. No.	Author Name	Title	Journal/Conference/Year	Main findings and Conclusion
[141]	Agarwa let al.	ScAA: A dataset for automated short answer grading of children's free-text answers in Hindi and Marathi	Proceedings of the 17th International Conference on Natural Language 2020	Even BERT-based ASAG models make errors on ScAA, showing the need for further research.
[142]	Sun, J., Song, T., Peng, W., & Song, J.	A survey of automated essay scoring: Challenges, advances, and future.	*Neurocomputing*, *650*, 130916. Elsevier (2025)	The study reviews the evolution of Automated Essay Scoring (AES), highlighting the transition from feature-based statistical methods to neural network and

				pre-trained language model approaches, and emphasizes future research directions such as trait-based scoring and cross-domain evaluation.
[143]	Sanuvala, G. et al.	Automatic short answer scoring on an Indian dataset using transformer-based language models	International Conference on Computer & Communication Technologies, 2023	Transformer-based embeddings enable effective short answer grading on Indian datasets.

2.7 Research Gap and Motivation

Although significant advancements have been made in Automated Answer Scoring (AAS), a number of challenges still persist, especially when it comes to evaluating responses in low-resource languages like Hindi. Most existing systems focus on high-resource languages, primarily English, with models and datasets specifically tailored to their linguistic structure. This creates a major shortcoming in the generalizability and applicability of these systems to other languages that exhibit different morphological and syntactic patterns.

Key Challenges in Automated Answer Scoring (AAS)

- Small research in Hindi: The majority of the AAS systems are based on English and other high-resource languages.
- Language adaptability problems: English based models are not easily generalized to morphologically rich languages such as Hindi.
- Poor semantic modeling advanced methods such as LSA and word embeddings are not often optimized over Hindi.
- Morphological complexity: The current systems have been unable to accommodate Hindi inflections and syntactical variations.
- Absence of hybrid solutions: Not many systems combine lexical, syntactic, semantic and contextual characteristics.
- Few AI-based solutions: There are not many AI-based solutions to Hindi AAS (machine learning and deep learning).

Contribution of Proposed Research

- Develop a Hindi-specific Automated Answer Scoring framework**.**
- Integrate syntactic and semantic feature extraction**.**
- Apply machine learning/deep learning models for improved scoring accuracy.
- Ensure fair and reliable evaluation of Hindi descriptive answers**.**

In terms of feature representation, many earlier approaches rely heavily on basic lexical features such as word counts or TF-IDF vectors (Ajay et al., [69]; Darwish et al., [70]), which are often insufficient for capturing deeper semantic understanding. Even though techniques like Latent Semantic Analysis (Cutrone et al., 2011; Ratna et al., 2013) and word embedding (Mikolov et al.) offer better semantic modeling, they are seldom trained or adapted for Hindi, making them less effective in capturing the nuances of morphologically rich languages.

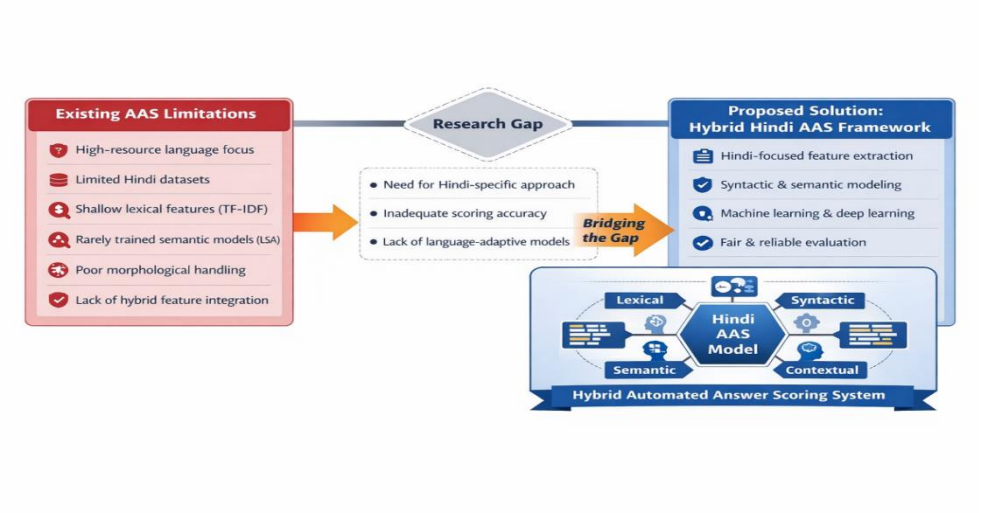


Figure 2.1: Conceptual Framework for Hindi Automated Answer Scoring System

Furthermore, despite the exploration of various semantic similarity measures—such as cosine similarity, path-based, and knowledge-based approaches (Pantulkar & Srinivasu, 2017; Gomaa & Fahmy, 2013) there is still a lack of integration of these methods in systems designed specifically for Indian languages. Most of the existing systems do not leverage hybrid feature sets that combine syntactic, semantic, and contextual insights in a meaningful way. These limitations motivate the need for a more comprehensive and language-adaptive approach to AAS. The proposed research seeks to bridge these gaps by developing a robust

feature extraction and modeling pipeline tailored to the characteristics of the Hindi language. It aims to incorporate both syntactic and semantic features, supported by machine learning or deep learning methods, to enhance scoring accuracy and fairness in Hindi answer evaluation.

2.8 Summary

The literature review provides an overview of advancements and challenges in automated answer scoring (AAS), emphasizing the need for a hybrid approach that integrates multiple evaluation criteria. It reviews various methodologies, frameworks, and technologies, ranging from traditional rule-based approaches to modern machine learning and deep learning models. Key areas of focus include syntactic and semantic analysis techniques and the limitations of current systems, especially when dealing with complex languages like Hindi. It highlights the need for further research, particularly in the context of morphologically rich languages and the evaluation of long, descriptive answers. It discusses the evolution of AAS, from traditional techniques to machine learning models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Support Vector Machines (SVMs). Despite significant progress, challenges remain, particularly in handling multilingual datasets, with most research focusing on English.

Additionally, the review emphasizes the importance of syntactic features (sentence structure and grammar) and semantic features (meaning and context) in improving scoring accuracy. Various methods for semantic similarity, such as cosine similarity, path-based similarity, and word-embedding models, are discussed, along with their role in understanding the underlying meaning of responses. Machine learning techniques, including regression, decision trees, and random forests, are explored for automating the scoring process, with several studies comparing their performance. The review also addresses the gap in AAS systems for Hindi, a language with a large speaker base but limited research in this area. It stresses the urgency of developing scoring systems for Hindi, especially as it becomes more prominent in educational and professional contexts. Overall, the review sets the stage for the proposed research, identifying key gaps and informing the design of an improved automated answer scoring system.

Chapter-III

Linguistic Feature Extraction for Automated Answer Scoring

3.1 Introduction

The success of automated answer scoring (AAS) depends on diverse features because these elements help instruction of neural networks and supervised machine learning models to correctly analyze student responses. The model requires these features to deliver accurate scoring decisions because they detect important elements of student answer starting from linguistic correctness through content relevance to structural coherence. Feature extraction stands as a vital step during this process to find meaningful information in text content. Under this step unstructured student responses are converted into structured representations for successful machine learning analysis or rule-based systems interpretation [135]. The process of feature extraction produces simplified data sets that contain features which directly link to the scoring task.

In AAS, feature extraction refers to the process of identifying relevant characteristics from a text response that can be used to assess its quality. These include linguistic features and content-related aspects that determine how well the student's answer line up with the given question. The scoring model relies on features derived from text processing which deliver deep answer analysis for its foundation.

In particular, linguistic features play a key role, encompassing both syntactic and semantic dimensions of language. Syntactic features relate to the structural and grammatical elements of a response, while semantic features capture the meaning, context, and similarity to the reference answer. Combining these features enable automated systems to evaluate not just how an answer is written, but also what it communicates, ensuring a comprehensive approach to scoring.

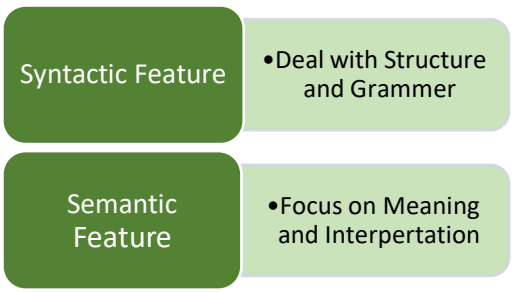


Figure 3.1: Linguistic Feature

AAS features fall into three common categories based on their syntactic along with semantic characteristics. Statistical-based features, often used in regression models, style-based (syntactic) features, which focus on the writing style and structure, and content-based features, which emphasize semantic relevance and meaning While statistical features are frequently used in traditional machine learning approaches like regression, neural network models tend to incorporate both style-based and content-based features due to their ability to catch complex syntactic patterns and deep semantic meaning.

This chapter will now explore syntactic and semantic features in detail, as they form the backbone of linguistic and meaning-based evaluation in automated answer scoring. Syntactic and semantic features useful for AAS are outlined in Figure 3.2, and a detailed description of these features is provided in the subsequent sections.

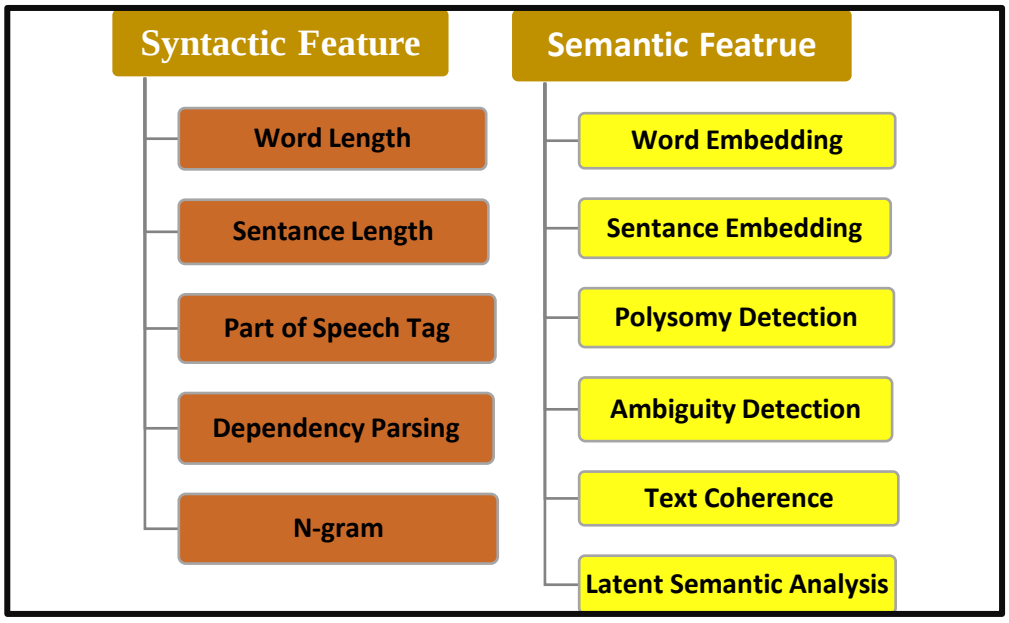


Figure: 3.2 Syntactic and Semantic Features for Automated Answer Scoring

3.2 Syntactic Feature for Automated Answer Scoring

In automated answer scoring, syntactic features play a critical role in evaluating the linguistic complexity and structural coherence of student responses. These features focus on the structure and pattern of words within a sentence, assessing adherence to grammatical rules. Common syntactic indicators include word length, sentence length, part-of-speech (POS) tagging, dependency parsing, and n-gram patterns. These elements help gauge the completeness, writing proficiency, readability, and overall depth of a response. Automated

scoring systems typically analyze text using a range of linguistic features to determine quality. Among these, text length features are widely used due to their simplicity and effectiveness in general text analysis [148]. Such features could consist of the average number of words in a sentence or the average number of sentences. By comparing the length of student responses to reference answers, these features contribute to a more objective assessment. Text length metrics are often used alongside other syntactic properties such as POS tags and dependency parsing. The subsequent sections explain the syntactic features that are used in this study.

3.2.1 Word Length

Word length is typically measured in terms of the average number of characters per word giving insight into the complexity of vocabulary used by students. Longer words often indicate the use of more sophisticated vocabulary, which can be associated with higher proficiency levels. Conversely, shorter words may suggest simpler language usage. The formula for computing the average word length is:

$$Average\ Word\ Length = \frac{\sum characters\ in\ words}{total\ no\ of\ words}$$

Longer word length might indicate a well-articulated response, while excessively short words could imply a lack of depth. The word length is computed using Algorithm, which calculates the average number of characters per word, is given below:

1. **Input**

Student answer S

2. **Tokenization**

Split the student answer S into words:

$$W=\{w_1,w_2,\ldots,w_n\}$$

3. **Word Length Calculation**

For each word $w_i \in W$, compute its length:

$$l_i = |w_i|$$

where W set of words in answer

4. **Aggregation**

Compute the average word length:

$$L_{avg} = \frac{\sum_{i=1}^{n} l_i}{n}$$

Where l_i is the length (number of characters) of word W_i and n is the total number of words.

5. **Output**

Average word length L_{avg}

3.2.2 Sentence Length

Sentence length refers to the average number of words per sentence, providing a straightforward yet powerful indicator of the level of detail and syntactic complexity in the answer. This feature allows for a preliminary comparison between student and reference answers, helping to identify responses that may be overly simplistic or lacking necessary information. Longer sentences can reflect complex sentence structures and a higher level of syntactic maturity, while shorter sentences may indicate simplicity or lack of elaboration. The formula for average sentence length is:

$$Average\ Sentence\ Length = \frac{Total\ Words}{Total\ Sentence}$$

This feature is especially relevant for evaluating descriptive or explanatory answers, where length can reflect the depth of the student's understanding and the richness of their response. Higher-scoring responses in automated scoring systems often exhibit moderate to long sentence lengths, as they tend to contain more explanation, reasoning, and supporting details. However, excessively long sentences might introduce readability issues, making clarity an important factor. The word count and number of sentences in the student's response should, at first, indicate whether it is detailed in the same way as the reference answer. The step-by-step method for sentence length extraction is outlined in Algorithm.

1. **Input**

Student answer S

2. **Sentence Segmentation**

Split the student answer S into ordered sentences:

$$S = \{s_1, s_2, \ldots, s_m\}$$

3. **Sentence Word Count**

For each sentence $s_j \in S$, count the number of words:

$$Len\ (s_j) = \text{Number of words in } s_j$$

4. Aggregation

Compute the average sentence length:

$$L_{avg} = \frac{\sum_{j=1}^{m} len(s_j)}{m}$$

If m = 1 (only one sentence), $L_{avg} = len(s_1)$

5. Output

Average sentence length L_{avg}

If the reference answer is long and detailed, and the student's answer is significantly shorter, it might indicate a lack of completeness. The shorter length of the student's answer might suggest that it lacks important details, even though it is grammatically correct. By correlating word and sentence length with other syntactic features like POS tagging and dependency parsing, we can gain deeper insights into the structural properties of the answers, aiding in more accurate automated scoring.

3.2.3 Part-of-Speech (POS) Tagging

Before anything else in NLP, POS tagging helps correct grammar and words, making it possible for answer scoring software to evaluate correctly [149]. It categorizes words based on their part of speech, using the way they are used and what they mean in a sentence. Tagging is an automatic system that assigns descriptions to each token.

A POS tagger assigns grammatical categories such as noun, verb, or adjective to each word or token in a text, helping to analyze the syntactic structure and grammatical correctness of language. POS tagging becomes particularly significant in the context of Hindi, a morphologically rich language with fewer standardized linguistic rules and limited annotated corpora, making the development of accurate POS taggers more challenging [150]-[151].

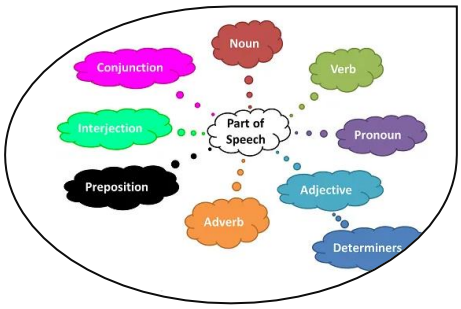


Figure 3.3: POS Tagging

Rule-based, statistical and hybrid are the major ways to do POS tagging [149]. Rule-based tagging uses hand-written linguistic rules along with contextual information, but it struggles with unknown or unseen text due to its dependence on predefined rules. Statistical approaches rely on the frequency and probability of word-tag pairs observed in annotated corpora. While effective, they may sometimes generate grammatically incorrect sequences. Hybrid approaches combine both methods—applying probabilistic models first, followed by language-specific rules—and often yields better performance than either approach alone.

In this study, POS tagging is applied to lemmatized tokens using tools such as the pos-tag function of the Natural Language Toolkit (NLTK) and the Stanza toolkit, which is trained specifically on Hindi corpora. The POS-labeled data was stored in separate columns of the dataset, enabling comparative analysis of various pre-processing techniques. The algorithm used for calculating POS tag is given below

1. **Input**

Student answer S

2. **Tokenization**

Split the student answer S into words:

$$W = \{w_1, w_2, \ldots, w_n\}$$

3. **POS Tagging**

For each word $w_i \in W$, assign a Part-of-Speech (POS) tag t_i using a POS tagger:

$$T = \{t_1, t_2, \ldots, t_n\}$$

4. **POS Count Vector**

Count the frequency of each POS category (e.g., NOUN, VERB, ADJ, etc.) in T:

$$V_{POS} = [\text{count(NOUN)}, \text{count(VERB)}, \text{count(ADJ)}, \ldots]$$

5. **Output :** POS vector V_{POS}

This tagging facilitated the identification of syntactic patterns in both questions and student responses, providing valuable insight into grammatical consistency and coherence. Furthermore, POS tagging formed the foundation for more advanced syntactic tasks such as dependency parsing, thereby put up to the overall effectiveness and accuracy of the scoring system [152]. To incorporate syntactic information into machine learning models, the POS tags were converted into vector representations. This transformation allows the syntactic structure of student responses to be quantitatively evaluated, enhancing the system's ability to assess language use and grammatical patterns in automated answer scoring.

3.2.4 Dependency Parsing

The aim of dependency parsing is to add meaning to words in a sentence by connecting them into structural relationships [153]. Here, Natural Language Processing (NLP) depends on analyzing word links both in the construction of sentences and in their meanings. In a dependency structure, each word (except the first one) depends on another nearby word and is labeled with a grammatical token such as subject, object or modifier. How components of a sentence are connected in the deep structure plays an essential role in tasks such as information extraction, semantic role labeling, machine translation and answering questions with higher accuracy. Lately, dependency-based parsing stands out in syntax due to its better and more direct ways to model how sentences are built than regular constituency parsing [154]. Languages that have free word order are best handled by dependency parsing which handles word relationships more directly than strict phrase structures. Progress in dependency parsing such as new algorithmic techniques and neural models, has become noticeable in recent times. Tratz et al. [155] used advanced parsing methods to help NLP tasks operate more effectively, proving that modern parsing is becoming more effective. In the view of Kübler et al. [156], the main ways of doing dependency parsing are through grammar and data-driven methods. To parse a sentence, grammar-based parsing relies on manually created rules and gives the resulting relationships in a tree diagram. Unlike the previous approach, data-driven dependency parsing depends on annotated data and machine learning to discover parsing rules for themselves. At the moment, data-driven methods are preferred because of their versatility and much better results compared to others in real situations.

Dependency parsing separates the sentence into main words and their modifiers so it can be understood better. The central word in the sentence is the head which determines the meaning of the other words. It shows the relation between the speech and its related words. It uses another (the head) as a reference to fully make its point and to build meaning for the listener or reader. A sentence is turned into a dependency tree during the process of dependency parsing. Roots, nodes and edges are part of what makes up a Graph. A dependency tree has nodes and each node corresponds to a word in the sentence. These nodes keep details about the word including its lemma, PoS tag and extra information. Edges show the grammatical connections between the individual words in the sentence. Every edge is marked with a description of the way the two words are related. As in a tree, the topmost

word in the sentence is the root and this word directs the rest of the sentence. The detailed procedure for c the Dependency Parsing is presented in given Algorithm.

1. **Input**

Student answer S

2. **Sentence Segmentation**

Split the student answer S into ordered sentences:

$$S = \{s_1, s_2, \ldots, s_m\}$$

3. **Tokenization**

For each sentence s_j, tokenize into words:

$$W_j = \{w_1, w_2, \ldots, w_{n_j}\}$$

4. **Dependency Parsing**

For each sentence s_j, apply a dependency parser to extract dependency triplets:

$$D_j = \{(\text{head, relation, dependent})\}$$

head = governing word

relation = grammatical relation (nsubj, obj, amod, etc.)

dependent = dependent word

5. **Aggregation**

Combine all dependency triplets from all sentences:

$$D = \bigcup_{j=1}^{m} D_j$$

6. **Dependency Vectorization**

Convert the dependency set D into a vector representation for scoring:

$$V_{dep} = [\text{count}(\text{relation}_1), \text{count}(\text{relation}_2), \ldots, \text{count}(\text{relation}_k)]$$

Each dimension represents the frequency of a dependency relation in the answer.

7. **Output:** Dependency vector V_{dep}

Recognizing these dependencies makes it clear which words in a sentence are the subject, verb or other parts and which are related in such ways as subject and verb or adjective and noun. The subject in a sentence usually takes on the role of the agent. They supply extra information on the word they follow. Object-Verb (O-V): The object receives the action of the verb, usually in transitive verb structures. Preposition-Object (P-O): In a prepositional phrase, the preposition governs the object it introduces. Auxiliary-Verb

(Auxiliary-Head): An auxiliary verb allows us to make tenses, moods or voices and it relies on the main verb.

This detailed syntactic analysis is crucial for evaluating the structural correctness of student responses, allowing the automated scoring system to assess not just the presence of correct words, but also their proper usage and arrangement. The dependency parsing results are integrated into scoring model, contributing to a more comprehensive evaluation of the syntactic quality of the answers. To facilitate the integration of dependency parsing results with other quantitative features, the parsed structures are converted into vector form. This conversion allows the syntactic relationships to be effectively utilized in machine learning models, thereby enhancing the automated scoring system's ability to evaluate the structural quality and grammatical accuracy of the responses.

3.2.5 n-gram Feature

An n-gram feature captures local syntactic patterns within the text. It represents a contiguous sequence of *n* items, typically words or characters, within a given text. The process involved segmenting each sentence into n-grams and then computing their occurrence across the dataset. This method allows for capturing short-range dependencies and common structural patterns, which are indicative of specific syntactic constructions in Hindi. By focusing on these n-grams, the model can better assess the grammatical consistency and coherence of student responses relative to the reference answers. For syntactic feature extraction, both bigram (n=2), and trigram (n=3) models are useful to analyze the frequency and distribution of these sequences in the student and reference answers. For clarity and reproducibility, the procedure is expressed as Algorithm.

1. **Input:**

Student answer S, n-gram size n (e.g., 2 for bigrams, 3 for trigrams)

2. **Tokenization**

Split the student answer S into words:

$$W = \{w_1, w_2, \ldots, w_m\}$$

3. **N-Gram Generation**

For each sequence of n consecutive words, generate n-grams:

$$NG=\{(w_1, w_2, \ldots, w_n), (w_2, w_3, \ldots, w_{n+1}), \ldots, (w_{m-n+1}, \ldots, w_m)\}$$

4. **N-Gram Vocabulary Creation**

Create a vocabulary of all unique n-grams in the answer (or in the dataset if vectorizing globally):

$$Vocab_{ngram} = \{ng_1, ng_2, \ldots, ng_k\}$$

5. **Vectorization**

Convert the n-grams of the answer into a **frequency vector** based on the vocabulary: $V_{ngram} = [count(ng_1), count(ng_2), \ldots, count(ng_k)]$

6. **Output:** n-gram vector V_{ngram}

Bigrams help in understanding how pairs of words co-occur in the context, providing insights into common phrase structures, while trigrams capture slightly more extended syntactic relationships. The resulting n-gram frequency vectors are then used as features in the automated scoring model, contributing to the overall syntactic evaluation. Combine n-gram vectors with other syntactic and semantic features to form a comprehensive feature set, which work as input into the scoring model to predict the final scores.

3.3 Semantic Feature in Automated Answer Scoring

Semantic Features focus on the meaning and content of the words and sentences. It analyzes the meaning conveyed by the student's answer, comparing it with the reference answer to assess accuracy, relevance, and coherence. Semantic features include word and sentence embedding, polysemy detection, ambiguity detection, text coherence and latent semantic analysis.

3.3.1 Word and Sentence Embedding

Word embedding makes it possible to use words by turning them into numbers that include the sense and context in few and thin dimensions, so they can be mixed into different models [158]. At present, Word embedding help solve different problems in sentiment analysis and other NLP activities. In A vector in a high-dimensional space is what defines the meaning of every word. Synonyms or terms that share meanings have vectors that are grouped by this system. The actual values of these vectors are learned from large corpora. They serve as the input layers in deep learning methods [159]. The embedding of semantically similar words tends to be close together in the vector space. An example to illustrate how word embedding transforms words from text to vector space is given in figure 3.4. Semantically similar words cluster together in this space.

The most popular types of word embedding are Word2Vec, Glove and fastText. After that, various methods of neural network embedding have been applied to documents, sentences and paragraphs. Sentence embedding is obtained by aggregating the word embedding of individual words in a sentence using methods such as averaging or weighted sum. Word embedding can be projected to a lower dimension space using (PCA) for visualization purpose. This can help in understanding the relationships between words and identifying clusters of similar words.

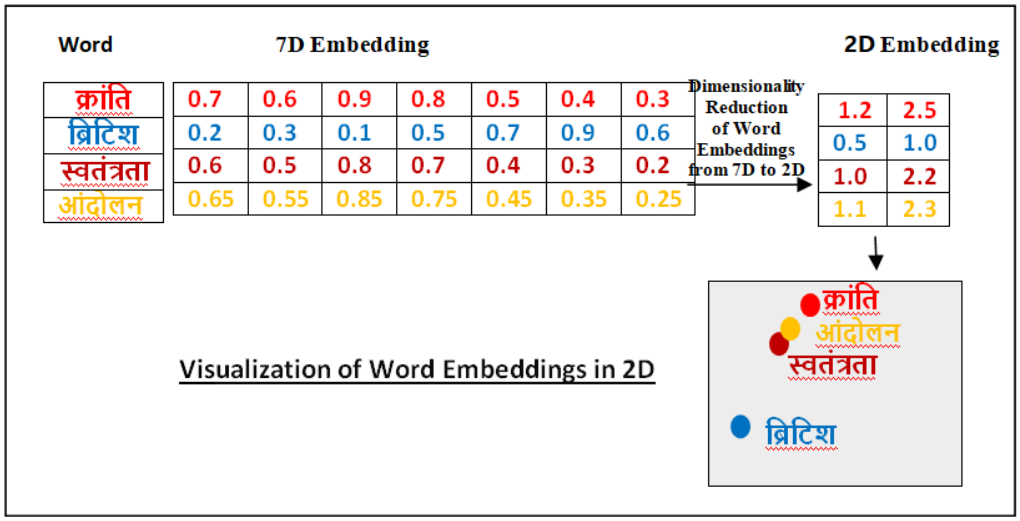


Figure 3.4: Visualization of Word Embeddings

To formalize the process, the algorithm for word and sentence embedding is described below

1. **Input:**

Student answer S

Pre-trained embedding model M

2. **Tokenization**

Split the student answer S into ordered words:

$$W = \{w_1, w_2, \ldots, w_m\}$$

3. **Word Embedding Extraction**

For each word $w_i \in W$

$$e_i = M(w_i) \in R^d$$

where, d = embedding dimension.

Collect all embeddings into a matrix:

$$E_W=[e_1,e_2,\ldots,e_m]^T$$

Handle out-of-vocabulary (OOV) words by assigning:

Random vector, Zero vector, Subword embedding.

4. Sentence Embedding Construction

Aggregate word embeddings to form a sentence embedding:

Average Pooling:

$$E_S = \frac{1}{m}\sum_{i=1}^{m} e_i$$

Weighted Pooling (TF-IDF):

$$E_S = \frac{\sum_{i=1}^{m} tfidf\ (w_i).e_i}{\sum_{i=1}^{m} tfidf\ (w_i)}$$

Contextual Models (e.g., BERT/SBERT):

Use the [CLS] token or pooled sentence representation directly:

$$E_S = M(S)$$

5. Output

Word embedding matrix E_W and Sentence embedding vector E_S

Doc2Vec, Google Sentence Encoder (GSE) or InferSent as the usual architectures for these part of NLP [162]. Mikolov et al. worked for Google and released Word2Vec [163] which is widely known as the leading word embedding method. Using Word2Vec effectively decreased the number of features in text classification and raised the accuracy. It is manufactured as an open-source project at Stanford [164]. According to [165], the authors proposed that FastText is a word embedding technique that describes words by their n-gram characters. Word2vec, GloVe and FastText learn about language using a window which means they cannot pick up information from sentences spread across long periods. It is now clear that BERT's latest architecture involves using the transformer network. It is capable of overcoming the problem when the word is not in the known vocabulary. Also, it can embed words and sentences together. Google designed Bidirectional Encoder Representations from Transformers (BERT) as a way to contextualize words [164]. It depends on a transformer-encoder that includes multiple layers and attends to parts of the input in both directions.

BERT makes it possible for a machine to grasp the meaning of words in a sentence with proper context. Bi-directional Encoder Representations and Transformers (BERT) is a

method which can generate vector representation from long sentences. Compared with traditional word embedding, BERT can essentially avoid the problem of word segmentation [167].To capture the deep semantic representation of the student and reference answers, sentence embedding is required. BERT is a pre-trained transformer model that processes text bi-directionally, providing context-aware embedding for sentences. The lemmatized sentences from both student answers and reference answers were fed into the pre-trained BERT-Base, Multilingual Cased model. This version of BERT supports the Hindi language, ensuring the semantic nuances of the text are well represented.

For each sentence, the BERT model generates a 768-dimensional embedding. BERT is applied as an entire model in the process of automated scoring. Some researchers used BERT with extra manually-designed features and this approach helped improve performance. It was found in [168] that the automated model boosts performance compared to other approaches [168]. The high-dimensional sentence embedding produced by BERT is reduced using Principal Component Analysis (PCA) to lower the computational complexity of the subsequent models. After experimentation, the optimal number of principal components is determined, balancing accuracy and computational efficiency. The reduced-dimensional embedding was then used as input to the machine learning and deep learning models.

3.3.2 Polysemy Detection

Polysemy refers to words that have multiple meanings depending on context. For example, in Hindi, environmental discourse is the word "संरक्षण" (*Sanrakshan*), which can have different meanings based on context:"पर्यावरण संरक्षण अधिनियम 1986 लागू किया गया।" Here, "संरक्षण" refers to protection or preservation of the environment. "वन संरक्षण के लिए सरकार ने कई कदम उठाए हैं– In this case, "संरक्षण" refers to conservation of forests. Using BERT embeddings, we can identify and disambiguate polysemous words to ensure that the correct meaning is understood in the given context. Generate word embeddings using BERT. Calculate cosine similarity between different occurrences of polysemous words in varying contexts. Detect if multiple meanings are present and determine the appropriate sense in the student's answer. The algorithmic representation for polysemy detection is given below.

1. **Input:**

Student answer S and Lexical resource

2. **Tokenization**

Split the student answer S into words:

$$W = \{w_1, w_2, \ldots, w_m\}$$

3. Sense Inventory Lookup

For each word $w_i \in W_i$

Retrieve its possible senses from a lexical database (e.g., WordNet):

$$\text{Senses}(w_i) = \{s_1, s_2, \ldots, s_k\}$$

If using contextual embeddings (e.g., BERT), identify multiple **sense clusters** of the word across contexts.

4. Polysemy Identification

For each word w_i

If $|\text{Senses}(w_i)| > 1$, then w_i is **polysemous**.

Assign polysemy weight:

$$p(w_i) = |\text{Senses}(w_i)|$$

5. Sentence-Level Aggregation

Compute the polysemy score for the whole answer:

$$P = \frac{1}{m}\sum_{i=1}^{m} w_i$$

This gives the **average number of senses per word** in the answer.

6. Output

Polysemy score P for the student answer.

3.3.3Ambiguity Detection

Sentences or phrases that have several meanings because of their context are called ambiguous. The existence of words, phrases, or sentences with multiple interpretations—is a complex and intriguing aspect of human language, crucial to how we communicate and understand meaning. Ambiguity is a persistent challenge in NLP because computer requires organized data but human speech is unstructured and frequently confusing in nature. How a single word or phrase is meant can change from one situation to another. For instance, consider the sentence**:** "महात्मा गांधी भारतीय स्वतंत्रता संग्राम के एक मुख्य नेता थे।" This sentence can be interpreted in two different ways: महात्मा गांधी स्वतंत्रता संग्राम के सबसे प्रमुख नेता थे। and महात्मा गांधी मुख्य नेताओं में से एक थे, लेकिन अकेले नहीं।

Words in some languages may mean many things and the way sentences are structured may permit several meanings. Fromkin et al. [169] state that ambiguity can be either lexical or structural and structural is sometimes referred to as grammatical. Using the same word in two different ways causes trouble when people communicate, whether by writing or talking. When words have more than one meaning, it appears in writing. In daily speech, it happens because many forms of a word sound the same [170]. She pointed out [171] that certain sentences or phrases appear unclear since their grammatical structure allows them to be understood in more than one sense. When a sentence becomes unclear due to its structure alone and not the meanings of the parts, it is called structural ambiguity. The cause of this kind of ambiguity is the way words or phrases in a sentence are set up.

Such difficulties create major issues for NLP since the main purpose is to ensure computers can read and interpret human speech accurately. Still, unclear statements in natural language may cause problems with the scoring of the responses. Many times, these misunderstandings happen because of a shortage of knowledge, different ways of saying things and hidden assumptions. Hence, it is necessary to remove any ambiguity in the requirements to make them better and more accurate.

The step of ambiguity detection is given in figure 3.5. To handle ambiguities in natural language requirements, the process begins with identifying unclear or vague statements using NLP techniques or expert evaluation.

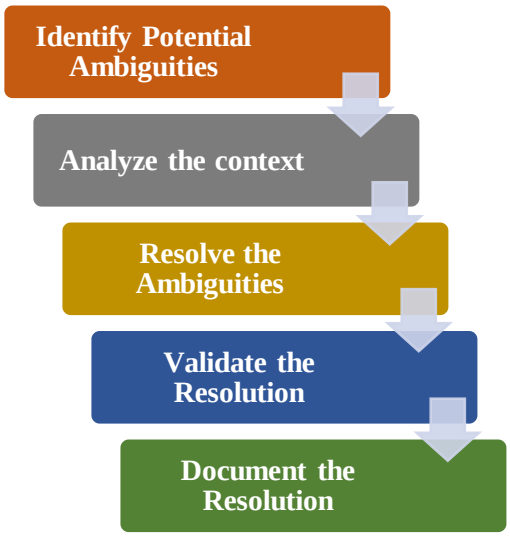


Figure 3.5: Ambiguities Detection Steps

These ambiguities are then analyzed in context, considering domain-specific knowledge and the intended interpretation of the requirements. Understanding the background helps in resolving the ambiguities through rephrasing, adding missing information, or eliminating implicit assumptions. The clarified requirements should be validated by stakeholders or domain experts through reviews or testing.

1. **Input:**

Text T (student answer)

2. **Preprocessing**

Normalize text (remove punctuation, lowercasing). Tokenize T into words w_1, w_2,..., w_n Perform POS tagging on each token.

3. **Lexical Ambiguity Detection**

For each token w_i, Query a lexical database. Get the number of senses S (w_i) If S (w_i) > 1, mark token as lexically ambiguous**.**

4. **Lexical Ambiguity Score:**

$$A_{lex} = \frac{\sum_{i=1}^{n} 1[S(w_i) > 1]}{n}$$

Where, 1 is an indicator function.

5. **Syntactic Ambiguity Detection**

Parse sentence using a dependency or constituency parser. Generate all possible parses P(T). Count number of valid parses |P(T)|. If |P(T)|>1, mark sentence as syntactically ambiguous**.** Syntactic Ambiguity Score:

$$A_{syn} = \frac{|P(T) - 1|}{|P(T)|}$$

6. **Combined Ambiguity Score**

Combine lexical and syntactic ambiguity into a single metric:

$$A_{total} = \alpha . A_{lex} + (1 - \alpha) A_{syn}$$

where $\alpha \in [0,1]$ is a weight depending on importance.

7. **Decision**

If $A_{total} \geq \theta$ (threshold), classify the text as ambiguous**.** Else, classify as unambiguous.

This ensures that the revised statements meet the intended goals. After validation, it's crucial to document each ambiguity, the chosen resolution, the reasoning behind it, and the contributors involved in the process. This systematic approach improves the clarity, consistency, and reliability of software requirements.

3.3.4 Text Coherence

Text quality analysis depends greatly on the coherence level. It reviews how linked the sentences are and checks if the document is properly structured. A document written with coherence discusses the changes in topic smoothly and usually follows the transition from simple to more complicated information. To measure coherence, the following methods are employed: A model analyzes sentence transitions and checks if ideas progress logically from one sentence to the next. Text coherence measures the semantic similarity between Question and reference answer, Question and student answer, Reference answer and student answer. Coherence between these pairs ensures accuracy in grading, consistency in automated systems, and relevance in student responses. To formalize the process, the algorithm for text coherence is described below.

1. **Input**

Student Answer S

2. **Sentence Segmentation**

Split the student answer S into ordered sentences

$$S=\{s_1,s_2,\ldots,s_m\}$$

3. **Sentence Embedding Generation**

For each sentence s_j, compute a dense vector embedding v_j using a sentence embedding model

4. **Pairwise Similarity Calculation**

For each consecutive sentence pair (s_j,s_{j+1}), Compute semantic similarity:

$$sim(s_j, s_{j+1}) = \frac{v_j . v_{j+1}}{\|v_j\|\|v_{j+1}\|}$$

5. **Aggregation of Similarities**

Collect all similarities into a set

$$SIM=\{sim(s_1,s_2),sim(s_2,s_3),\ldots,sim(s_{m-1},s_m)\}$$

6. **Coherence Scoring**

Compute the average similarity:

$$C = \frac{\sum SIM}{m-1}$$

If m=1 (only one sentence), set C=1.0

7. **Output**

Coherence score C.

It allows educational platforms to grade answers fairly and provide meaningful feedback. This is done using BERT embedding and calculating cosine similarity between the embedding of sentences to evaluate semantic alignment. High similarity indicates good coherence between the question and answers.

3.3.5 Latent Semantic Analysis (LSA)

LSA is not a conventional NLP solution, in that it does not make use of human-designed ontologies, dictionaries, bodies of knowledge, semantic nets, grammars, syntax parsers, and morphological analyzers [143]. It is possible to use LSA to see how words are connected in various parts of text. Latent Semantic Analysis (LSA) is a technique in natural language processing that uses mathematical methods to analyze and identify the underlying relationships between words and concepts in large sets of text [172].LSA is used to measure semantic similarity between the student's answer and the reference answer. This method evaluates the overall topic alignment and relevance of the student's response. LSA was developed at first to help improve how information is retrieved from systems by checking the semantic meaning of words in questions and answers instead of checking the words themselves. An advantage of this process is that it handles problems associated with synonymy in which the same meaning can be covered by many different words.

LSA assumes that there is some base to all languages, or "latent," structure in the pattern of word usage across documents, and that statistical techniques can be used to estimate this latent structure. The term documents in this case can be thought of as contexts in which words occur and also could be smaller text segments such as individual paragraphs or sentences. From analyzing connections between words and documents, this method constructs a model where similar words are closer together. LSA builds a word count record for every piece of text by making a matrix of where every word appears in each document. It converts the text into a numerical format using methods like Term Frequency-Inverse Document Frequency (TF-IDF) to create a term-document matrix

$$TF - IDF(t,d) = TF(t,d) \times log\left(\frac{N}{DF(t)}\right)$$

Where t is a term, d is a document, N is the total number of documents and DF(t) is the number of documents containing term t. LSA then uses singular-value decomposition

(SVD), a technique closely related to eigenvector decomposition and factor analysis. Apply SVD to the term-document matrix to decompose it into three matrices: U, Σ and V^T

$$A= U\Sigma V^T$$

The original term-document matrix is A and U holds the left singular vectors. In a diagonal form, Σ stores the singular values and VT contains the columns that are the right singular vectors. The SVD method splits the original word-by-document matrix into k, often 100 to 300, orthogonal factors and we can get close to the original matrix by multiplying these factors together. LSA handles representations by converting documents and terms into continuous values on each of the k SVD-derived indexes, rather than by using single independent words. Since there are fewer factors or dimensions than unique terms, words won't be able to occur independently. If two terms are used together in many documents, they will have similar vectors in the reduced dimensional LSA space. The algorithm for LSA is presented in Algorithm X

1. **Input:**

Student answer S, Reference answer R and Preprocessed corpus C (questions, reference answers, student answers after tokenization + lemmatization)

2. **Text Preprocessing**

Convert all text to lowercase, tokenize into words, remove stopwords and punctuation and apply lemmatization.

3. **Construct Term-Document Matrix**

Build a term-document matrix A using TF-IDF weighting:

$$A_{ij} = \text{TF-IDF} (t_i, d_j)$$

Where, t_i = term, d_j = document (student/reference answers).

4. **Apply Singular Value Decomposition (SVD)**

Decompose matrix A:

$$A = U\Sigma V^T$$

U is the term-topic matrix, Σ s_j the diagonal matrix of singular values and V is the document-topic matrix

5. **Dimensionality Reduction**

Keep only top k singular values ($k \ll \text{rank}(A)$)

$$A_k = U_k \Sigma_k V_k^T$$

This reduces noise and captures latent semantic structure.

6. **Represent Documents in LSA Space**

Represent student answer S as vector v_S in reduced space and represent reference answer R as vector v_R in reduced space.

7. **Similarity Calculation**

Compute cosine similarity between student and reference answer:

$$\boldsymbol{LSA(S,R)} = \frac{\boldsymbol{v_S . v_R}}{\|\boldsymbol{v_S}\|\|\boldsymbol{v_R}\|}$$

8. **Output**

LSA (S,R), the semantic similarity score based on latent semantics.

A good thing about this process is that it can link two pieces of text together even if they don't share any words. SVD's analysis can be seen from the perspective of geometry. Because of SVD, each term and each document is represented by a k-dimensional vector. The location of document vectors mirrors the similarities between the terms appearing in the documents. In this space, the cosine or dot product between vectors corresponds to their estimated semantic similarity. Thus, by determining the vectors of two pieces of textual information, we can determine the semantic similarity between them. LSA helps determine the semantic similarity between a student's answer and a reference answer by comparing their representations in the reduced space. This is essential in automated scoring systems where the goal is to evaluate whether a student's response is conceptually aligned with expected answers. By analyzing the relationships between words, LSA can identify key concepts and themes within responses, enabling better scoring based on understanding rather than exact wording. LSA provides a powerful framework for analyzing text data and scoring answers based on semantic content rather than strict lexical matching, which is particularly useful in educational and assessment contexts.

3.4 Tools and Techniques for Syntactic and Semantic Feature Extraction

In the field of Natural Language Processing (NLP), effective feature extraction plays a pivotal role in enabling machines to understand and analyze human language. For tasks like automated answer scoring, syntactic and semantic features are particularly important as they capture the structural and meaning-based nuances of text. Python-based NLP and machine learning packages are widely used due to their flexibility, extensive community support, and open-source availability. Among these, NLTK (Natural Language Toolkit) is one of the most dominant tools, extensively employed for pre-processing and basic linguistic tasks. Additionally, Java-based frameworks such as Stanford NLP are also popular, especially in academic and enterprise-grade applications.

To better understand the practical implementation of feature extraction, the following tables 3.1 and 3.2 provide a categorized overview of commonly used syntactic and semantic features, along with the respective tools and libraries used to extract them. These tools support a wide range of linguistic tasks, from basic text processing to advanced semantic modeling, and are essential for developing robust NLP applications such as automated answer scoring.

Table 3.1: Tools and Techniques for Syntactic Feature Extraction

Syntactic Feature	Description	Common Tool/ Libraries
Word Length	Average length of word in sentence	Python (len function), NLTK, spaCy
Sentence Length	Number of word in sentence	Python, NLTK, spaCy
Part of Speech Tagging	Grammatical role of each word (noun, verb, adjective, etc.)	spaCy, NLTK, Stanford POS Tagger, Flair
Part of Speech Count	Counts of different POS tags (e.g., number of verbs, adjectives)	NLTK, spaCy, Flair
Dependency Parsing	Grammatical relationships between words (subject, object, etc.)	spaCy, Stanford CoreNLP, Stanza, UDPipe
n-gram	Sequences of 'n' consecutive words or tokens	NLTK, scikit-learn, TextBlob, spaCy, gensim

Table 3.2: Tools and Techniques for Semantic Feature Extraction

Semantic Feature	Description	Common Tool/ Libraries
Word Embeddings	Vector representations of words	Word2Vec (gensim), GloVe, fastText, spaCy, Flair
Sentence Embeddings	Vector representations for entire sentences	Sentence-BERT (SBERT), Universal Sentence Encoder, spaCy
Polysemy and Ambiguity Detection	Identifying words with unclear interpretation	WordNet (via NLTK), BERT-based models, spaCy, sense2vec
Text Coherence	Measures logical flow across sentences	Coh-Metrix, BERT Score, entity grid models
Latent Semantic Analysis (LSA)	Extracts relationships using statistical techniques	scikit-learn (Truncated SVD), gensim, LSA implementation in NLTK

3.5 Summary

This chapter explores the role of linguistic feature extraction in the development of Automated Answer Scoring (AAS) systems, focusing on both syntactic and semantic features. It begins with an introduction to the significance of linguistic features in improving the accuracy and reliability of automated evaluation. Feature extraction forms the foundation of AAS by enabling machines to interpret student responses in a structured and meaningful way. The chapter first delves into syntactic features such as word and sentence length, part-of-speech (POS) tagging, dependency parsing, and n-gram patterns, which help in assessing the grammatical correctness and structural coherence of student answers. It then examines semantic features that capture the meaning and relevance of responses, including word embeddings, sentence embedding, polysemy and ambiguity detection, text cohrance and Latent Semantic Analysis (LSA). These semantic features measure the conceptual similarity between student and reference answers. To support implementation, the chapter also provides a curated list of widely used tools and libraries for feature extraction. These include NLTK,

spaCy, Stanford CoreNLP, Stanza, gensim, Scikit-learn, and Transformers from Hugging Face, among others. By combining syntactic accuracy with semantic understanding and leveraging these tools, the chapter presents a comprehensive framework for building effective and reliable AAS systems that align closely with human judgment.

Chapter-IV

Proposed System and Methodology

4.1 Introduction

Automated Answer Scoring (AAS) is a modern method used in education to check student answers using technology. It understands what the student has written through Natural Language Processing (NLP) and Machine Learning (ML). This system looks at the content, structure, and relevance of the answer to give a score. AAS has become popular because it saves time and gives accurate and reliable results. It helps teachers understand student performance better. As schools start using more digital tools for assessment, AAS is being used more often. The main aim is to make scoring more consistent and fair by using smart computer techniques.

This chapter describes the step-by-step approach followed to build the Automated Answer Scoring (AAS) system for Hindi language responses. The methodology integrates multiple stages including data acquisition, preprocessing, and model development using both traditional and advanced techniques. The focus is on designing a system that is capable of evaluating student answers automatically, accurately, and consistently. To achieve this, various machine learning and deep learning models were implemented, along with a proposed hybrid approach that combines their strengths. The methods were evaluated using standard performance metrics to ensure the reliability of the scoring system.

To illustrate the systematic approach, a flowchart is presented to visually map the research workflow and its key stages. This flowchart visually represents the key stages of the study, from data collection and resource analysis to feature extraction, integration, scoring methodology, and validation. This structured approach shown in figure 4.1 not only clarifies the relationships between different components but also emphasizes the logical progression of the research, ensuring that each objective is systematically addressed. The detail of the data used and method adopted for the present investigation have been described under various section and subsection of this chapter.

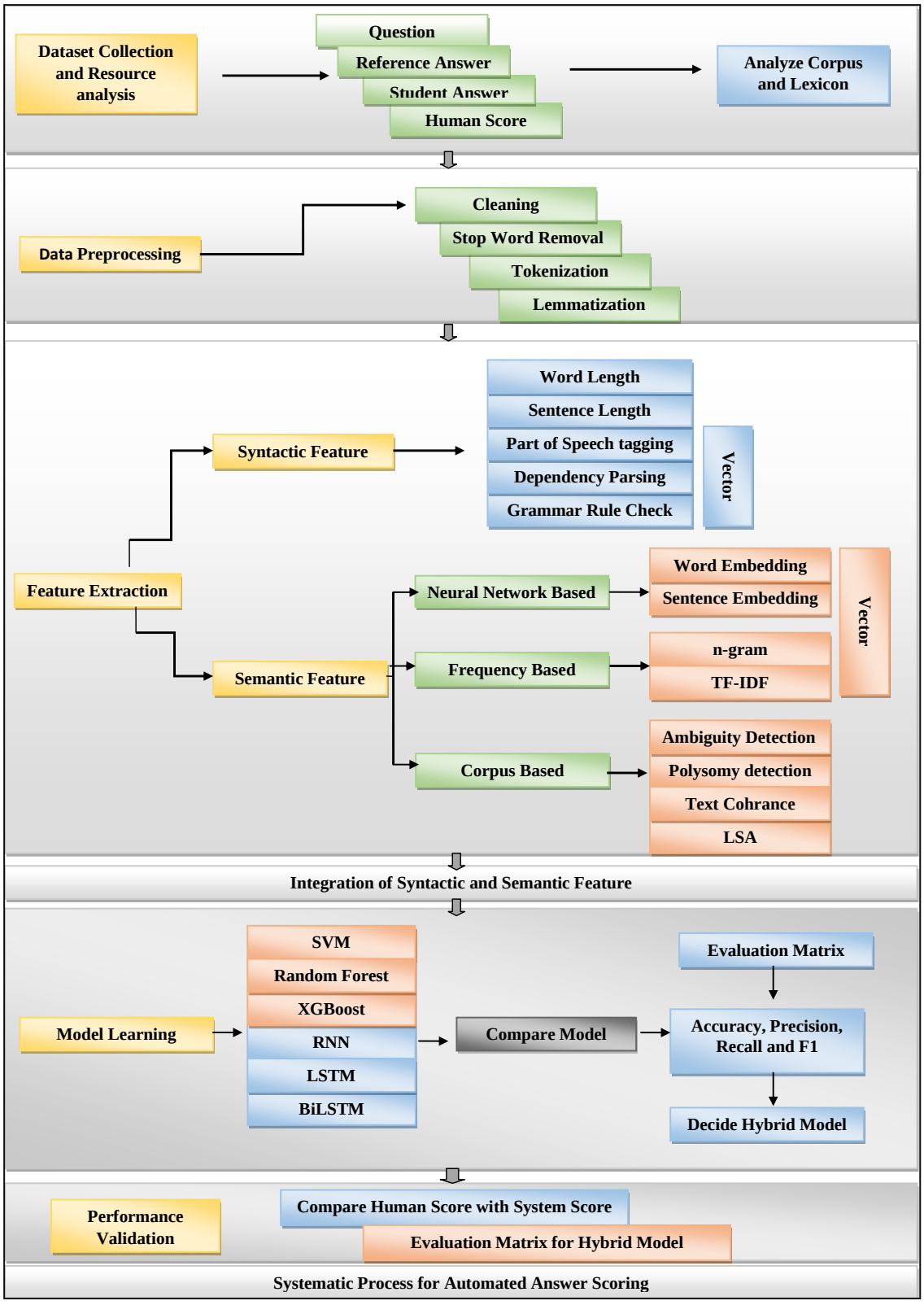


Figure 4.1: Systematic Process for Automated Answer Scoring

4.2 Dataset Characteristics

The dataset used in this study contains Hindi-language question-answer pairs collected from a diverse range of academic sources, including annual examinations, mid-term exams, and classroom tests. It encompasses questions from multiple subjects and various grade levels, making it suitable for analyzing a broad spectrum of student responses.The dataset includes approximately 2500 student answers, both long-form and short-form, enabling the model to learn from varied answering patterns, levels of detail, and linguistic complexity. This rich and varied data helps the Automated Answer Scoring (AAS) system generalize well across different types of responses. All answers are written in Hindi, making this a language-specific dataset focused on the semantic and syntactic nuances of Hindi. This specialization is significant, as Hindi has unique linguistic structures compared to English, such as free word order and rich morphology, which need to be carefully handled during NLP-based analysis.

Each data entry is stored in a structured format, consisting of a question, reference (model) answer, student answers and manually assigned scores by three independent evaluators These human-assigned scores serve as the ground truth for training and evaluating of the scoring models. The use of multiple evaluators also helps reduce individual bias and enhances the reliability of annotations. Where needed, the collected data was cleaned to ensure consistency and remove any identifying information. This structured and high-quality dataset forms the foundation for training robust machine learning and deep learning models aimed at automated scoring of student responses in the Hindi language.

4.3 Pre-processing of Dataset

The datasets received are sometimes not clean and therefore it needs to be cleaned and properly managed to be suitable for analysis. Processing the data before testing is very important with AAS systems. The data should be made appropriate by discarding any extra or unwanted details. Data pre-processing helps make sure data is uniform, so the grading system remains consistent for every dataset. Pre-processing involves cleaning and transforming the raw text data to standardize it making it compatible with syntactic and semantic feature extraction in subsequent objectives. This step is crucial for ensuring data consistency and reliability in the AAS system. Pre-processing involves different steps such as cleaning, tokenization, removal of stop words and lemmatization, which are described in the following subsections:

4.3.1 Data Cleaning

Data cleaning is a critical step in preparing the dataset for analysis, as it ensures the quality and integrity of the text. It identifies and eliminates special characters like punctuation marks and symbols from the text. These characters do not add semantic value and can introduce noise. This involves using predefined lists or regular expressions to find and remove such characters. Utilize spell-checking tools or custom dictionaries tailored to the Hindi language to identify and correct common spelling mistakes.

4.3.2 Tokenization

In this phase, the input text is turned into a list of tokens by using a tokenize function known as NLTK. It receives text input, splits it into several words and gives a list of tokens that allow the model to grasp and process the text information by looking at each word in order. Of course, before text processing starts, the text should first be segmented into words, numbers and alphanumeric and so on. Normally, terms are separated by blanks, but not all white space works the same. This process is crucial for subsequent analysis, as it allows each word to be processed independently, facilitating a more granular examination of the text. In the context of Hindi, tokenization can be particularly challenging due to the language's unique script and grammatical structures. To ensure accurate tokenization, advanced natural language processing tools (indic NLP) that are specifically used to handle Hindi text. These tools take into account various linguistic features, such as compound words, affixes, and context, which are essential for correctly identifying tokens.

4.3.3 Stopword Removal

They are words such as “it” and “and,” that typically have very little impact on the meaning of a sentence. This helps in focusing on the meaningful words that contribute to the text's overall semantic content. This step enhances the quality of the text data by eliminating redundant words. In addition, taking out stop words makes the model run more smoothly and fast. It uses a predefined list of Hindi stop words, which includes common words that do not carry significant meaning and are often removed to reduce noise.

Stop words such as “में”, “और”, “से”, “ने”, “के”All these are removed from the text because they are not important for text analysis. The list of stopword can be sourced from existing linguistic resources or customized based on the specific dataset. For text pre-

processing of Hindi dataset, a list of Hindi stopwords obtained from the publicly available GitHub repository NLTK Hindi Stopwords. This stopword list was used to filter out common words that do not supply significantly to the meaning of a sentence This improves the efficiency of the system as removal of stop words save storage space and processing time.

4.3.4 Lemmatization

Lemmatization means changing words into their basic or original forms. This ensures uniformity across answers, allowing the system to recognize different forms of the same word as equivalent. By relying on grammar, NLP libraries are able to transform Hindi words by bringing them to their basic forms. It means the main terms of a document are shown as stems, instead of the specific words they consist of. Therefore, variants can be merged which helps reduce the dictionary's size [173]. For every given word, the form should be reduced to its root after analyzing where it is used in the sentence [174]. For example, the sequence of rules describes that "आंदोलनों" should be changed to "आंदोलन" WordNet Lemmatizer from NLTK is used by this function to lemmatize the input text. In this step, the exact meaning of the text is considered when reducing the words in it. The benefit of lemmatization is that it always produces words that are acceptable in terms of being valid. The words are provided as a list of lemmatized tokens and then these tokens are tagged for part of speech. By following these preprocessing steps, the Hindi dataset will be clean, well-structured, and standardized, making it suitable for AAS and other text analysis tasks.

4.4 Machine Learning Model

Machine Learning is a sub domain of artificial intelligence that empower systems to learn from data and make predictions or decisions without being explicitly programmed. Machine Learning involves training algorithms on structured input data, where the system learns patterns and relationships between input and output based on examples provided during training. The input is typically represented through various linguistic, syntactic, and semantic aspects of the text. Traditional machine learning methods require manual effort to extract meaningful information from the text.

The system uses this information to identify similarities and differences between student responses and expected answers. Over time, it learns how certain types of responses are typically scored and builds a model that can replicate this behavior automatically.

Machine learning models can perform well even with limited data, as long as the input representations are informative and well-crafted.

4.4.1 Support Vector Regression (SVR)

SVR is designed as a regression variation of the SVM and is best suited to be used when scoring student responses for its goal of predicting a continuous outcome [175]. It establishes a robust regression framework for predicting student scores based on extracted syntactic and semantic features. SVR is used when scoring relies mostly on features that are numerical or statistical, rather than needing a deep understanding of sentence structure.

The ability of SVR to deal with many data points makes it useful for grading student answers because its extracted features often mix linguistic, statistical and semantic parts. Unlike SVM, which finds a perfect hyper plane to separate data into different classes [176], SVR focuses on minimizing the error within a certain margin while predicting numeric values. This makes SVR more appropriate for educational settings where student responses are evaluated on a fine-grained scoring scale. SVR's capability to learn from complex patterns in data allows it to produce highly accurate predictions, even in the presence of noisy or incomplete answers [177].

One of the major limitations of using SVM in answer scoring is that it is fundamentally a classification algorithm. It assigns answers into discrete categories (such as 'right' or 'wrong'), which may not capture the subtle differences in quality across a range of student responses. In contrast, SVR can assign continuous scores that reflect the degree of correctness or completeness in an answer. This level of scoring precision is essential in large-scale assessments, where subjective elements like explanation depth or language quality can vary significantly. SVR is thus better aligned with the scoring rubrics typically used in educational assessments, which demand more than simple categorical judgment [178]. In The prediction mechanism of SVR for automated answer scoring can be formally expressed as follows:

$$\hat{y} = f(x) = (w, x) + b$$

Where, x is the input (feature) vector representing student answer. It is made up of all the extracted features you used, w represents the importance (weights) the model assigns to each feature. A continuous score $\hat{y}$ represent the predicted quality of a student's answer on the

scoring scale. Another significant advantage of SVR is its use of kernel functions to handle nonlinear and high-dimensional feature spaces. The radial basis function (RBF) kernel, in particular, is widely used because of its ability to capture complex relationships in the data and produce stable predictions [177]. This is critical in natural language processing tasks like automated answer scoring, where features may include lexical diversity, syntactic structures, and semantic similarity. Despite the complexity of these features, SVR maintains relatively constant computational complexity irrespective of input dimensionality, making it scalable and efficient for real-world educational applications.

4.4.2 Random Forest

Random Forest (RF), introduced by Breiman [179], is a powerful ensemble-based machine learning algorithms are widely used in both classification and regression task. In the domain of automated answer scoring, RF has shown strong potential, particularly in predicting continuous scores in regression-based scoring systems. RF operates by building multiple decision trees, each trained on a bootstrapped subset of the training data. For regression tasks, the final prediction is obtained by averaging the outputs of individual trees. This approach, combined with randomized feature selection, enhances the model's generalization performance and reduces over fitting [180].

In answer scoring applications, Random Forest learns from feature vectors derived from syntactic features (e.g., dependency relations, n-grams) and semantic similarities (e.g., word embedding, cosine similarity). Its ability to handle a diverse set of linguistic features makes it effective in evaluating a wide range of student responses. One of the key advantages of RF is its interpretability. The model provides feature importance rankings, which help researchers and educators understand the relative contribution of each feature in the scoring process. This is especially valuable for educational assessment, where transparency and fairness are critical.

Moreover, RF is known for its efficiency and scalability. It requires minimal hyper parameter tuning, yet it benefits from parameters like the total of trees, maximum depth, and minimum samples per split, which can be optimized to improve performance [181]. The model also supports the identification of partial matches in student responses, capturing nuances that rule-based systems may miss. However, RF has limitations. It cannot inherently capture sentence order or deep language patterns, as it primarily focuses on statistical correlations rather than contextual semantics. Despite this, its strength lies in processing a

combination of syntactic and semantic features, making it ideal for systems where large-scale, accurate, and interpretable scoring is essential. Figure 4.2 illustrates the architecture of the Random Forest model in automated answer scoring.

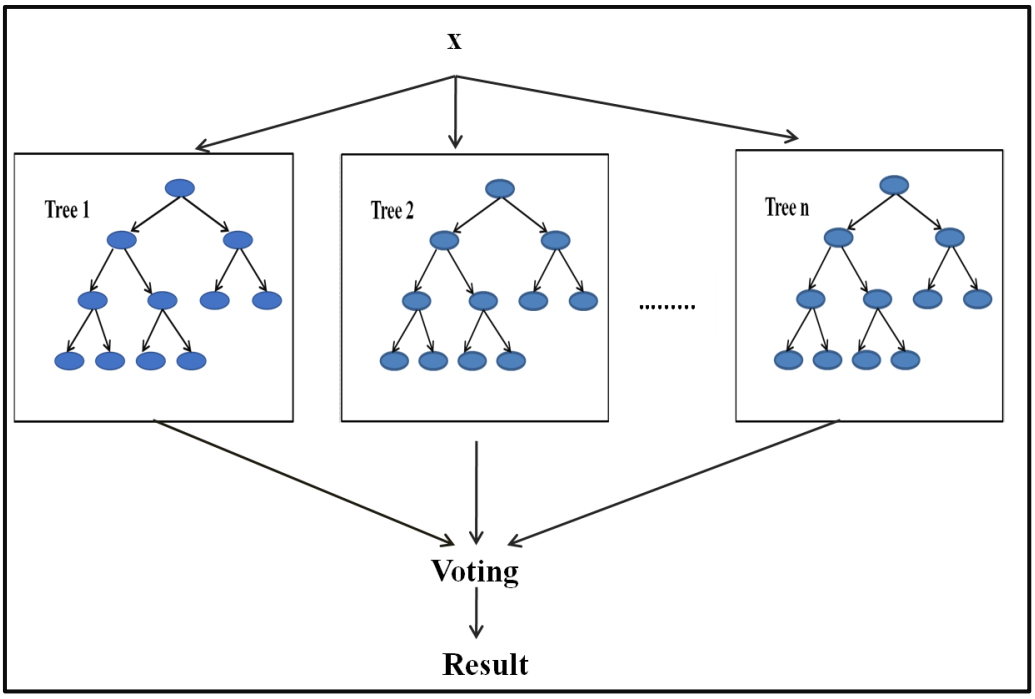


Figure 4.2 Architecture of Random forest

The first step in making a Random Forest is to randomly choose k subsets from the original training dataset through bootstrapping (i.e., sampling with replacement). Each of these k subsets is used to build one of the k decision trees (classification or regression trees). To grow each tree T_i, the following steps are repeated recursively for every non-leaf node until the maximum tree depth is reached:

1. Each node randomly identifies and uses a group of features.
2. This subset chooses its best features based on information gain in the case of classification and on how much mean squared error is decreased in the case of regression.
3. On the basis of this feature, the node separates into two new child nodes.

Once all k trees are grown independently, they are combined to form the final Random Forest model. The final prediction in classification is decided by checking the most frequent selection among the k trees. For regression, the prediction made is the average result of all

trees. In Random Forest, the input x represents the numerical feature vector of a student's answer, which is derived from both syntactic and semantic features. Each decision tree in the forest, denoted as $f_k(x)$ generates a predicted score for the answer by learning patterns from different subsets of features and samples. The final predicted score $\hat{y}$ is obtained by averaging the outputs of all decision trees in the ensemble, ensuring that the overall prediction is more stable and accurate compared to relying on a single tree. The mathematical representation of the Random Forest prediction is given as:

$$\hat{y} = \frac{1}{K}\sum_{k=1}^{K} f_k(x)$$

Where, K is the total number of decision trees in the forest, and $f_k(x)$ is the prediction of the k^{th} tree for the student answer.

4.4.3 eXtreme Gradient Boosting (XGBoost)

XGBoost, a powerful ensemble learning technique based on gradient boosting, useful for regression tasks to predict the scores of student answers based on both syntactic and semantic features. In Fig 4.3 as classified XGBoost, or eXtreme Gradient Boosting, has emerged as a state of-the-art algorithm for supervised learning tasks.

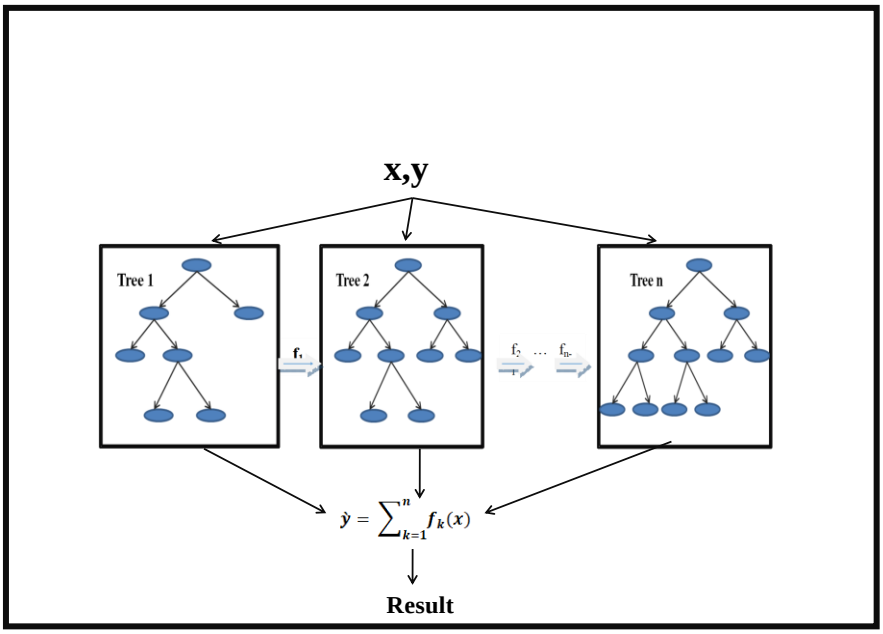


Figure 4.3: Architecture of XGBoost

By employing an ensemble of decision trees and optimizing a customizable loss function, XGBoost achieves remarkable accuracy and scalability. XGBoost builds a series of weak learners (usually decision trees) that correct each other's errors. It is highly optimized for performance and accuracy. All trees use the information from past trees as they grow. Rather than depending on most of the output from Random Forest, the final prediction in XGBoost is the sum of all the voting outcomes. In XGBoost, the input x represents the numerical feature vector of a student's answer, which includes both syntactic and semantic features. XGBoost builds an ensemble of regression trees sequentially, where each new tree is trained to correct the errors of the previous trees.

Each tree, denoted as $f_k(x)$, contributes to the predicted score by focusing on the aspects of the answer that were not captured accurately by earlier trees. The final predicted score $\hat{y}$ is obtained by summing the outputs of all trees in the ensemble, which allows the model to capture complex patterns in the features and produce more accurate predictions. The mathematical representation of the XGBoost prediction is given as:

$$\hat{y} = \sum_{k=1}^{K} f_k(x) \quad f_k \in \mathrm{F}$$

Where, K is the total number of regression trees in the ensemble, $f_k(x)$ is the prediction of the k^{th} tree and F denotes the space of all possible regression trees.

XGBoost is highly accurate, especially when you have many structured features (e.g., TF-IDF scores, word embedding, syntactic pattern). It's also flexible enough to handle both continuous and categorical data [182].Regularization parameters and techniques like early stopping help prevent over fitting. XGBoost is well-suited for scoring tasks due to its scalability, speed, and ability to manage sparse data. Its feature importance mechanism also provides intuition into the linguistic aspects that present most to the scoring. In our context, XGBoost ability to handle large datasets and complex relationships between features and scores makes it a compelling choice for automating the grading process [183]. By iteratively refining predictions through the boosting technique, XGBoost offers a powerful tool for accurately assessing student responses.

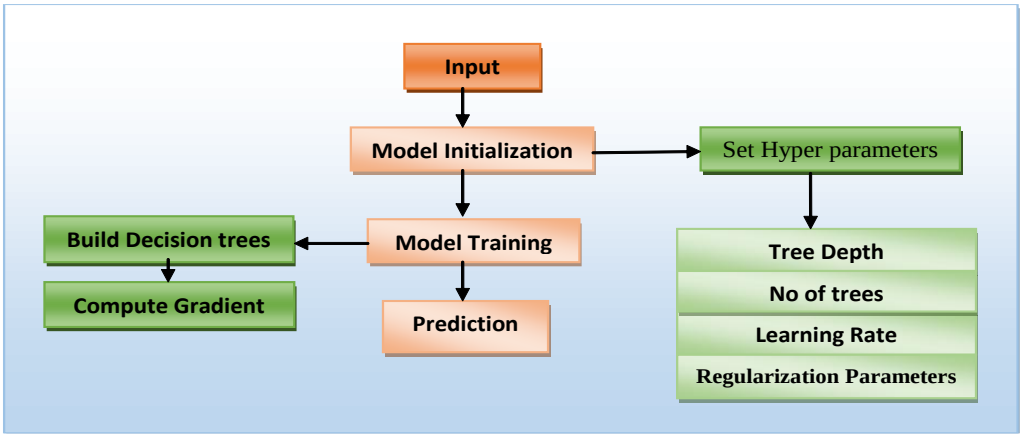


Figure 4.4: Systematic process for XGBoost model during Automated Answer Scoring

The input of XGBoost consists of PCA-reduced sentence embeddings (generated from BERT), syntactic features such as POS tag vectors, dependency parsing vectors, n-gram features and semantic features. During training process different hyper parameter are decided such as learning ratio, maximum depth and number of estimator. Then XGBoost is trained to minimize the Mean Squared Error (MSE) between predicted scores and actual scoresThe model is validated 5 times to check if it performs well on data it hasn't seen yet. Using the function for feature importance in XGBoost, the model can highlight the significance of every factor (syntactic and semantic) and help improve the hybrid technique.

4.5 Deep Neural Network

Deep Learning a significant sub-domain of machine learning, focuses on learning and representing complex patterns through multi-layer neural networks [184]-[185]. Deep Learning, a specialized branch of machine learning, focuses on building and training neural networks that can automatically learn useful patterns from raw data. Unlike traditional methods, deep learning does not rely heavily on manual input design. Instead, it uses multiple layers of processing units to understand complex structures and dependencies in the text. These models can capture the meaning and context of words and sentences by analyzing sequences of text and their relationships. They are especially good at understanding language patterns, word order, and the subtle variations in how information is expressed. Deep learning approaches can model both short and long texts effectively and are capable of learning deeper semantic understanding. While deep learning methods often require more data and computational power, they typically yield higher accuracy and generalization due to their ability to learn from the full richness of the textual content.

4.5.1 Convolutional Neural Network (CNN)

CNNs are neural networks made for handling two-dimensional information. CNN is known as an effective approach using a network made with several layers. CNNs were originally designed for image recognition but are also used for text by looking for local patterns (e.g., word pairs or triplets) using "convolutions." Image processing uses CNN to help identify and assign categories to images. CNN can process an image, analyze it and use different parts of the analysis to make automated decisions. Convolutional layers use filters to find important parts of the image in each input. Activation layers aggregate the output from previous layers and introduce non-linearity to aid learning. Pooling layers down sample the feature maps, reducing their dimensions to improve computational efficiency and fully attached layers interpret the extracted features to make final predictions or classifications.

The CNN finds applications in pattern classification, finding objects and recognizing objects. To represent a sentence, the authors rely on CNN and the sentence is then processed by a fully-connected hidden layer [186]. In text, CNNs scan small windows of words to detect relevant word patterns. CNNs can recognize patterns in short answers or key phrases, like specific word pairs or n-grams, which may correlate with high or low scores. In CNN, the input x represents the feature vector of a student answer applies convolutional filters over these features to capture local patterns, which are indicative of the answer's quality. The output of each filter is passed through a nonlinear activation and then pooled to reduce dimensionality and focus on the most significant features. Finally, the pooled features are flattened and passed through fully connected layers to produce the predicted score $\hat{y}$. The prediction in CNN can be mathematically represented as:

$$\hat{y} = f(Flatten\left(Pooling\big(Activation(Conv(x))\big)\right)$$

Where, Conv represents the convolution operation, Activation is a nonlinear function like ReLU, Pooling reduces dimensions, and Flatten prepares the features for the final fully connected layer.

CNNs struggle with longer sentences and capturing the broader context, as they focus on local patterns rather than full sentence meaning. Best for short answer scoring or when specific phrase patterns are indicators of scoring (e.g., certain keywords that correlate with a high-quality answer). CNNs are used to capture local linguistic patterns and relationships in the student and reference answers. CNNs are more powerful because they have additional

layers, known as convolutional layers, which allow them to process higher flexibility and improved skills in finding important patterns and features because information is not processed in a straight line [187].

4.5.2 Recurrent Neural Network (RNN)

Among various DL architectures, Recurrent Neural Networks (RNNs) are particularly appropriate for processing sequential data due to their potential to capture contextual dependencies over time. This characteristic enables RNNs to effectively model the flow of information in natural language, where the meaning of a word often depends on its surrounding context [188]-[189].Traditional methods of grading largely rely on manual assessment, which is labor-intensive and prone to subjectivity and inconsistency [190]. With the advent of DL, especially RNN-based models, it has become feasible to develop automatic grading systems that are both objective and scalable. These models learn from large corpora of language data, allowing them to identify intricate linguistic patterns and make reliable predictions about the quality of text responses.

At the core of automated grading systems lies the task of text similarity detection, where the goal is to compare student responses with reference answers to evaluate relevance and coherence. RNNs are particularly advantageous in this context due to their sequential processing nature, which allows them to consider dependencies not only from preceding words but also from subsequent words in the input sequence. This bi-directional flow of information significantly improves performance in tasks such as natural language processing (NLP), speech recognition, and machine translation [191]-[193].

In practical applications, many tasks involve interdependent data points, where understanding the overall sequence is critical rather than analyzing individual elements in isolation. Feed forward neural networks are limited in this regard, as they process inputs in a fixed manner without maintaining temporal context. In contrast, RNNs dynamically update their hidden states to reflect prior inputs, thereby maintaining a form of memory essential for tasks like language modeling, speech synthesis, human-machine dialogue, and real-time translation. With these strengths, RNNs have become a foundational tool in the development of intelligent, automated systems for educational assessment, especially in language learning environments where the nuanced evaluation of written and spoken responses is crucial. To better understand how RNNs function in processing sequential data, it is essential to examine their architecture and internal mechanism.

The architecture of a basic RNN is illustrated in Figure 4.5. It illustrates the basic architecture of a simple RNN. At every time step t, the RNN takes an input x_t, integrates it with the hidden state from the previous time step h_{t-1}, and computes the new hidden state h_t. This hidden state serves both as an output for the current time step and as input for the next one. Mathematically this can be represented as

$$h_t = \sigma\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right)$$

$$y_t = W_{hy}h_t + b_y$$

Where, x_t input, h_t hidden state, y_t output at time step t and W_{xh}, W_{hh}, W_{hy} weight matrices, b_h, b_y bias terms and σ activation function (typically tanh)

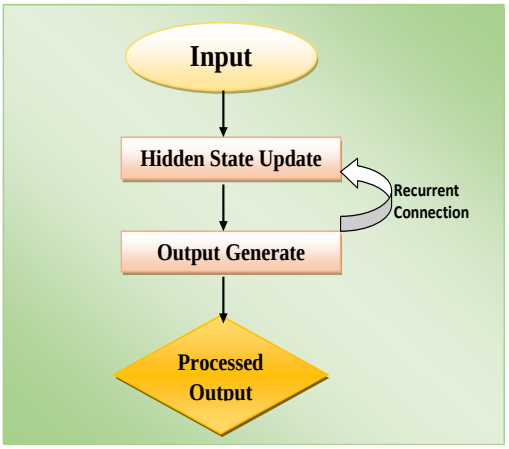


Figure 4.5: RNN Architecture

The RNN architecture consisted of one or two hidden layers, each with 100 to 200 neurons. Each hidden layer utilized a tanh activation function to model non-linearity in the data. The final hidden state from the RNN was passed to a dense layer, which produced a score prediction for the student answers. This output score was compared with the reference answer score for evaluation. The RNN model trained using Mean Squared Error (MSE) as the loss function to reduce the difference between forecasted and actual scores.

4.5.3 Long Short-Term Memory Networks (LSTM)

Since LSTMs overcome the vanishing gradient issue, they are used for processing and discovering trends in data that changes over a long period. With the use of this model, researchers in AAS have seen great results in a variety of NLP tasks. LSTM is a kind of RNN that stores the important sequence information by using its memory cell [190]. The forget gate decides which part of the earlier cell state will remain for the current cell state. The input gate decides how much of the input should be integrated into the current cell state. Lastly, the output gate governs the extent to which the current cell state influences [191]

LSTMs are designed to overcome the "forgetting" problem by retaining important information over longer sequences. They have memory cells that decide when to keep or forget information. Unlike RNNs, LSTM adds the memory of time t-1 information to its

output at time t [191]. It is shows in figure 4.6 how we can input information at each moment in time.

Forget gate f_t: $\sigma(W_f \cdot [x_t, h_{t-1}] + b_f)$

Input gate i_t: $\sigma(W_i \cdot [x_t, h_{t-1}] + b_i)$

Candidate cell C_t: $\tanh(W_c \cdot [x_t, h_{t-1}] + b_c)$

Update cell C_t: $f_t \odot C_{t-1} + i_t \odot C_t$

Output gate o_t: $\sigma(W_o \cdot [x_t, h_{t-1}] + b_o)$

hidden cell h_t : $o_t \odot \tanh(C_t)$

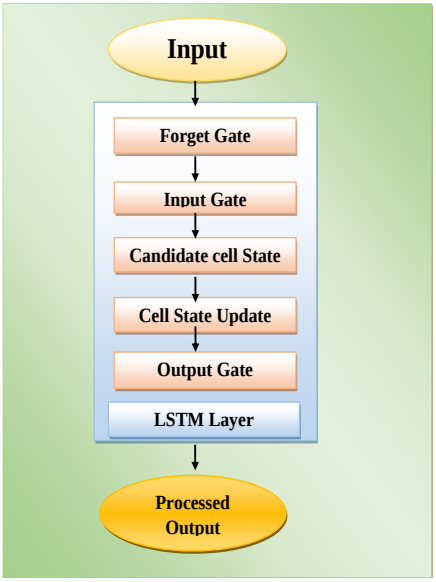


Figure 4.6 Architecture of LSTM

x_t is the input getting into the LSTM at the present time step t , ht is the hidden layer at that point in time and h_{t-1} are the output values from each memory cell in the previous hidden layer. The symbol σ stands for a sigmoid function and the symbol ⊙ denotes element wise multiplication while tanh refers to the hyperbolic tangent function. Wi is placed in the input layer one at a time. The final hidden state is then passed through a dense layer to produce the predicted score$\hat{y}$.

$$\hat{y} = Vh_T + c$$

where, **V** is the Weight matrix mapping the final hidden state and **c** is the Bias term for the output LSTMs handle longer sentences well and can remember the context over entire answers. They're useful when understanding sentence structure and flow is important to scoring. More computationally intensive than traditional RNNs, and may still struggle if the text is exceptionally long.

LSTMs process word embedding of student and reference answers, analyzing both short- and long-range linguistic dependencies. Forget, input, and output gates in LSTM cells allow the model to selectively retain relevant information and discard irrelevant details. This helps the model focus on meaningful parts of student answers that align with reference

answers. LSTMs are effective in identifying logical flow, grammatical consistency, and semantic coherence, which are critical for scoring.

LSTM, a specialized type of RNN, was used to capture long-term dependencies in sentences, especially when the answers were lengthy or had complex sentence structures. The same PCA-reduced sentence embedding were fed into the LSTM network, where each embedding represents a time step in the input sequence. The LSTM architecture used in this work had one or two LSTM layers with 128 units in each layer. The LSTM cells were responsible for learning the sequence-based dependencies within the answers. The output from the last LSTM cell was passed to a fully connected (dense) layer, which produced the final answer score prediction. The same Mean Squared Error (MSE) loss function was used to optimize the model.

4.5.4 Bidirectional LSTM (BiLSTM)

The bidirectional LSTM proposed by Schuster and Paliwal is an extension to the traditional LSTM. BiLSTMs read the sentence in both directions, forward and backward, capturing context from both before and after each word [192]. This allows the model to get a fuller picture of sentence meaning. This bidirectional processing allows the model to capture context from preceding and succeeding words simultaneously, providing a more holistic understanding of the answers. For automated scoring, BiLSTMs analyze the word embedding of student and reference answers, ensuring that both local and global contexts are taken into account.

BiLSTMs capture the complete meaning of each word within the sentence, making them effective for complex Hindi sentences where the context from both directions is essential for accurate scoring. Current information in this situation is dependent on past information and is tied to future information as well. Unidirectional LSTM looked only at the earlier parts of the input which sometimes meant missing important parts of the sentence.

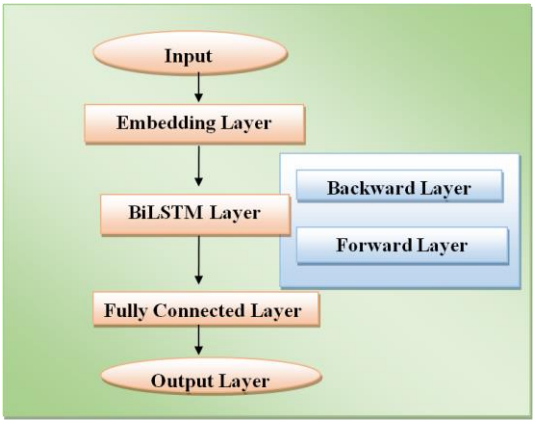


Figure 4.7 Architecture of BiLSTM

Input layer accepts sequence inputs (text data transformed into embedding). then embedding layer converts input words into dense vector representations. In BiLSTM layer two LSTM layers process the sequence in opposite directions, where forward LSTM processes from left to right and Backward LSTM: processes from right to left. Outputs from both directions are concatenated to captures long-term dependencies and context from both past and future words. Fully Connected (Dense) layer converts BiLSTM output into a fixed-length vector. Then output layer uses a softmax (for classification) or linear activation (for regression) and predicts a score between 0-10. Many fields have shown that bidirectional networks work much better than unidirectional ones. The purpose is to do two "passes" across the sequence, feeding the words from left to right to examine past information and from right to left to check future words. A bidirectional LSTM includes two separate hidden layers: one looks at input in the forward direction and is called $\overrightarrow{h_t}$ and the other works on the reverse order and is labeled $\overleftarrow{h_t}$. Both directions of the network work separately until the final layer and at this stage their outputs are connected together:

$$\overrightarrow{h_t} = g(W_{\vec{h}}x_t + W_{\vec{h}}\vec{h}_{t-1} + b_{\overrightarrow{h_t}})$$

$$\overleftarrow{h_t} = g(W_{\overleftarrow{h}}x_t + W_{\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h_t}})$$

$$y_t = g(W_{\vec{h}}\vec{h}_t + W_{\overleftarrow{h}}\overleftarrow{h_t} + b_y$$

In a BiLSTM network, x_t represents the input vector at time step t, which is typically a word embedding capturing the semantic information of a word in the student or reference answer. The forward and backward hidden states, denoted as $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, store contextual

information from preceding and succeeding words, respectively, allowing the model to capture dependencies in both directions. The output at each time step, y_t, represents the predicted score for that part of the input sequence. The network uses weight matrices W for the forward, backward, and output transformations, while bias terms b are added to each computation to allow flexible adjustment of the activations. Finally, the activation function g, typically tanh or linear depending on the task, introduces non-linearity and enables the network to model complex patterns in the input sequence BiLSTMs are computationally intensive and can be slow to train, especially on long texts [193]. Best for complex, nuanced answers where understanding the entire sentence context is crucial, especially for longer or grammatically intricate responses.

4.6 Hybrid Modeling Framework for Automated Answer Scoring

Machine Learning and Deep Learning techniques each offer unique advantages in automated answer scoring. While ML models provide interpretability and handle structured data efficiently, DL models excel at capturing complex, high-dimensional relationships, particularly in unstructured data such as text. Table 4.1 presents a summary of Machine Learning and Deep Learning Techniques employed in Automated Answer Scoring Systems. The table highlights the strengths, limitations and best use in Automated Scoring. A hybrid approach that combines both methodologies leverages the strengths of each, leading to more accurate and robust predictions. In this approach, DL models are used to extract rich, contextual representations of text, such as semantic relationships and sequence patterns. These deep features are then combined with traditional, manually engineered features that capture explicit information like word length, syntactic structure, and POS tagging.

The hybrid framework integrates both feature types into a unified model, where deep learning layers handle semantic features (e.g., word embedding and cosine similarity) and machine learning models capture the syntactic aspects (e.g., POS tags, dependency parsing). The combination of these methods is useful as it eliminates the limitations that individual methods have. Models using deep learning can explore complicated data, but they usually overlook basic relationships and struggle when small datasets are used. On the other hand, machine learning models, while effective on structured data, may not capture the deeper semantic context necessary for understanding nuanced text. By combining the strengths of both, the hybrid approach provides a more comprehensive analysis of student responses. To better understand the contribution of each component in the hybrid framework, a brief

overview of the individual models used is provided below. Each model brings unique capabilities that complement one another in the overall scoring process.

Table 4.1: Description of machine and deep learning technique in Automated Answer Scoring

Model	Strengths	Limitations	Best Use in Answer Scoring
SVR	Good for numerical features	Doesn't handle sentence context well	Simple similarity-based scoring
Random Forest	Works with mixed features	Lacks sentence-level understanding	Effective with syntactic and semantic features
XGBoost	High accuracy with complex data	High memory and slower	High-detail scoring with many features
CNN	Captures local patterns	Struggles with sentence meaning	Short answers or word patterns
RNN	Keeps word order	Struggles with long text	Short, sequential answers
LSTM	Handles long sentences	Slow and resource-heavy	Complex or longer answers
BiLSTM	Captures full sentence context	Computationally intensive	Best for full-context understanding in answers

Once individual machine learning and deep learning techniques were examined, we studied how they could work together to improve the method of automatic grading. A hybrid approach aims to leverage the unique strengths of ML and DL models to overcome their individual limitations. Specifically, deep learning models were employed to capture complex, high-dimensional semantic representations of text, such as word embedding and cosine similarity, which offer rich contextual information. These were then integrated with traditional, manually engineered syntactic features (e.g., POS tags and dependency parsing) captured by machine learning models. By combining semantic and syntactic features, the hybrid framework benefits from both the contextual power of deep learning and the interpretable, structured nature of machine learning models. Various hybrid combinations were tested, with deep learning layers handling the semantic aspects and machine learning models focusing on the syntactic features, ultimately improving the scoring accuracy.

4.7 Evaluation Methodology in Automated Answer Scoring

Several factors, for instance, accuracy, precision, recall and F1 score, are used to check how successful the hybrid process is. Besides these, the Pearson's Correlation Coefficient (r) examines if the predicted scores have a straight relationship with the scores annotated by humans, telling whether the machine matches human understanding well. Models are also tested with a cross-validation method to confirm their strength and general usefulness. The results are then compared to benchmark systems to demonstrate the improvements achieved by combining machine learning and deep learning models. These metrics determine how well the system's predicted scores align with human-annotated scores.

4.7.1 Accuracy

Accuracy measures the proportion of correctly predicted scores out of all samples. The formula for accuracy is

$$Accuracy = \frac{No\ of\ Correct\ Predictions}{Total\ no\ of\ Predictions}$$

In terms of classification, it is calculated as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

When TP matches with a high score, TN (True Negative) corrects with a low score, FP (False Positive) signals that a high score was wrongly classified and FN marks that a low score was wrongly predicted. A higher accuracy indicates better agreement between the automated system and human scores. However, it may not be sufficient if the dataset is imbalanced.

4.7.2 Precision

Precision measures how many of the predicted high scores are actually correct:

$$Precision = \frac{TP}{TP + FP}$$

If a system gives a high score, a high precision value means it is probably correct.

4.7.3 Recall

Recall measures how well the system identifies all high-scoring answers:

$$Recall = \frac{TP}{TP + FN}$$

A high recall value means the system is effective in identifying all correct responses but may also include some incorrect ones.

4.7.4 F1-Score

F1-score is the middle point between Precision and Recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

F1score is particularly useful in cases of imbalanced datasets, where accuracy alone may not provide a clear picture of model performance.

4.7.5 Pearson's Correlation Coefficient (r)

Pearson's correlation looks at how the numbers awarded by people and computers line up:

$$r = \frac{\sum(x_i - \bar{\bar{x}})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{\bar{x}})^2} \times \sqrt{(y_i - \bar{y})^2}}$$

Where, x_i = Automated scores, y_i= Human scores, $\bar{\bar{x}}$ = Mean of automated scores and $\bar{y}$ = Mean of human scores A correlation close to 1 indicates a strong agreement between human and model scores. By using these evaluation metrics, we ensure that the automated scoring system provides reliable and interpretable results.

4.8 Summary

The purpose of this chapter was to describe how an automated system for scoring answers was built, starting with a review of the necessary preprocessing techniques for Hindi language. To ensure the processed information was consistent, we applied normalization and to make the text simpler, we used lemmatization. We also extracted features such as part-of-

speech tags, named entities and syntax. To make the raw data fit for computational models, these steps must be applied.

This chapter came after preprocessing and it covered using machine learning and deep learning and mentioned the pros that each has. We applied SVM and Random Forests, both traditional machine learning methods, because they can easily explain cause and effect and function smoothly with structured types of data. In the same period, RNNs and models based on the Transformer architecture were chosen because they can analyze both the content and the context of the answers. Therefore, the method merges the two techniques so that the benefits of each can be used to analyze all types of student answers. The combination helps the model to succeed in different scenarios and recognize many different expressions in Hindi.

Research in automated answer scoring has been adopting deep learning and transformer models, including LSTM, BERT and neural architectures in recent years. Though these methods have shown encouraging outcomes, they usually need big annotated datasets and a lot of computational power to be trained. When it comes to the current research, i.e. Hindi question-answer evaluation, large-scale labelled datasets are not available, and the direct use of the intricate deep learning models is difficult. Thus, the study uses the conventional semantic similarity methods, such as TF-IDF and cosine similarity, along with the linguistic and syntactic feature extraction. These algorithms are computationally attractive, interpretable as well as appropriate with small datasets yet can model meaningful semantic associations between student response and reference responses. Moreover, feature-based methods are more transparent in the scoring process, which is a crucial attribute to the education assessment systems. Therefore, the research methods chosen are suitable in building a robust automated answer scoring system of the Hindi language responses.

Chapter- V

Experimentation and Evaluation

5.1 Introduction

A thorough evaluation of automated answer scoring necessitated a detailed examination of the dataset, feature extraction technique and model performance. This chapter presents the results of the automated answer scoring system developed for Hindi text responses, organized to address each of the research objectives outlined in the introduction. The main purpose of this study is to raise the accuracy and reliability of scoring by using an approach that gathers data, analyzes corpora and lexicons and applies a hybrid scoring system using both syntax and semantics. Initially, the work involves collecting data and carefully analyzing the corpus and vocabulary, allowing for an analysis of Hindi response patterns and language. The first analysis guarantees that the feature selection and scoring methods take into account the language qualities in the data. Subsequently, attention shifts to syntax, aiming to examine how various structures help determine accurate scoring and appreciation of semantics which manifest in the model interpreting the responses.

In this last section, a new system is built to score text, using both syntax and semantic information combined. . Different tests and measures in statistics are applied to estimate the model's accuracy and show why a combination approach is successful for text in Hindi. The conclusions reached here give us the basis to discuss what comes next for improving and using automated answer scoring systems. In the following sections and subsections, this chapter looks closely at every objective that was set for the study. Here, this discussion looks at how well the model agrees with human judgment on the scores it provides.

5.2 Resource Compilation and Dataset Preparation

In order to accomplish the first objective, we fully examined the resources needed for automated answer scoring. To do this, I worked on a range of questions and answers, along with answers from students, to collect a data set that covers many distinct topics and questions. Besides, we included Indian WuDaC lexicons, grammar lists and aligned resources, focusing on the Hindi language, to improve the accuracy of our scoring.

The team focused on using educational resources from textbooks, exams and verified experts to build a rich and relevant content collection. The words and examples included were hand-picked to represent changes in vocabulary, how students write and how they build their sentences. Because of these resources, the scoring model was able to distinguish good responses according to their wording, how natural the response was and what it meant. Using these data and resources allowed us to create a firm base for further work, specifically feature extraction and training models. By using well-built corpora and lexicons, the artificial intelligence was able to assess text in Hindi more accurately than if only using basic language tools.

5.2.1 Dataset Overview

For this study, the dataset contains Hindi questions along with corresponding answers from textbooks and experts on the topic. The data was arranged to feature a wide variety of subjects and kinds of questions, so students would be tested using factual and descriptive answers. For each question, one student's answer has been checked by three different people, helping us compare it to the machine's score. With more than one evaluator involved and agreeing on each answer's score, bias in the dataset is reduced. Table 5.1 represent the sample of the questions asked, reference answers, student answers and the corresponding scores used for automated answer scoring. This lets us understand the basic framework of the data.

Table 5.1: Sample of Collected Hindi dataset

Question	Reference Answer	Student Answer	S1	S2	S3
स्वतंत्रतासंग्राममेंमहात्मागांधीकीभूमिकापरविस्तारसेप्रकाशडालें।	महात्मागांधीनेभारतीयस्वतंत्रतासंग्राममेंअहिंसकआंदोलनोंकानेतृत्वकिया।उन्होंनेसत्याग्रह,असहयोगआंदोलन,सविनयअवज्ञाआंदोलनऔरभारतछोड़ोआंदोलनजैसेप्रमुखआंदोलनोंकेमाध्यमसेअंग्रेजोंकेखिलाफलड़ाईलड़ी।गांधीजीकाउद्देश्यथाकिबिनाकिसीहिंसाके,केवलसत्यऔरअहिंसाकेबलपरब्रिटिशशासनकोसमाप्तकियाजाए।उन्होंनेलोगोंकोआत्मनिर्भरताऔरस्वदेशीवस्त्रोंकेइस्तेमालपरबलदियाऔरखादीकाप्रचारकिया।उन	महात्मागांधीभारतीयस्वतंत्रतासंग्रामकेएकमुख्यनेताथे।उन्होंनेसत्यऔरअहिंसाकेसिद्धांतोंकेमाध्यमसेआजादीकीलड़ाईकोएकनईदिशादी।असहयोगआंदोलन, सविनय अवज्ञा आंदोलन, औरभारतछोड़ोआंदोलनजैसेबड़ेआंदोलनोंकेजरियेगांधीजीनेअंग्रेजोंकेखिलाफजोरदारविरोधकिया।उनकामाननाथाकिहिंसासेकुछनहींमिलेगा,बल्किहमेंशांतिऔरअहिंसाकेमार्गपरचलतेहुएअपनेअधिकारोंकेलिएसंघर्षकरनाचाहिए।उन्होंनेआत्मनिर्भरताऔरस्वदेशी	9	8	9

	कीनीतियोंऔरआंदोलनोंनेभारतीयजनतामेंआजादीकीललकजगाईऔर भारतीयस्वतंत्रतासंग्राममेंउनकामहत्वपूर्णयोगदानरहा।	उत्पादोंकेइस्तेमालपरजोरदिया।उनकीयहरणनीतिकाफीसफलरहीऔरइससेब्रिटिशहुकूमतकोमजबूरनदेशछोड़नेपरविचारकरनापड़ा।			
पर्यावरणसंरक्षणकेलिएसरकार द्वाराउठाएगएकदमोंकावर्णनकरें।	पर्यावरणसंरक्षणकेलिएभारतसरकारनेकईमहत्वपूर्णकदमउठाएहैं।पर्यावरणसंरक्षणअधिनियम 1986, वायुप्रदूषण (रोकथामऔरनियंत्रण) अधिनियम 1981, जल (रोकथामऔरनियंत्रण) अधिनियम 1974जैसेकईकानूनबनाएगएहैं। सरकारनेस्वच्छभारतअभियान,वनीकरणपरियोजनाएँ,जलसंचयऔरनवीकरणीयऊर्जास्रोतोंकेविकासजैसेकार्यक्रमशुरूकिएहैं।इसकेअतिरिक्त, भारतनेअंतरराष्ट्रीयसमझौतोंजैसेपेरिससमझौता, क्योटोप्रोटोकॉल, औरयूएनएफसीसीसीकापालनकरतेहुएपर्यावरणीयसंरक्षणकेलिएकईकदमउठाएहैं।	भारतसरकारपर्यावरणसंरक्षणकेलिएकईकानूनऔरपरियोजनाएँचलारहीहै। इनमेंस्वच्छभारतअभियानसबसेप्रमुखहै,जिसकाउद्देश्यपूरेदेशमेंस्वच्छताऔरपर्यावरणकोबेहतरबनानाहै।इसकेअलावा,सरकारनेजलसंरक्षण,प्लास्टिकप्रदूषणकोकमकरने, औरहरितऊर्जास्रोतोंकेविकासकेलिएकईयोजनाएँलागूकीहैं।सरकारनेपेरिससमझौतेजैसेअंतरराष्ट्रीयसमझौतोंमेंभागलेकरभीपर्यावरणसंरक्षणकेप्रतिअपनीप्रतिबद्धतादिखाईहै। इसकेसाथही,नदियोंकीसफाईऔरवनसंरक्षणकीदिशामेंभीकईप्रयासकिएजारहेहैं।	7	8	7
भारतीयकृषिकीवर्तमानचुनौतियोंकावर्णनकरें।	भारतीयकृषिकोजलवायुपरिवर्तन, जलकीकमी, उपजाऊभूमिकाकमहोना, औरतकनीकीविकासकीकमीजैसीकईचुनौतियोंकासामनाकरनापड़रहाहै।छोटेऔरसीमांतकिसानअधिकतरपारंपरिकतरीकोंसेखेतीकरतेहैं, जिससेउनकीउपजकमहोतीहै।इसकेअतिरिक्त,कृषिउत्पादोंकेमूल्यनिर्धारणमेंभीअस्थिरताहै, जोकिसानोंकीआयकोप्रभावितकरतीहै।सरकारनेइनसमस्याओंकोहलकरनेकेलिएकिसानसम्माननिधि,फसलबीमायोजना,औरजलसंरक्षणपरियोजनाओंजैसेकईप्रयासकिएहैं,लेकिनअभीभीकईसुधारोंकीआवश्यकताहै।	भारतीयकृषिकोआजकेसमयमेंकईगंभीरचुनौतियोंकासामनाकरनापड़रहाहै। सबसेबड़ीसमस्याहैजलवायुपरिवर्तन, जिससेखेतीकेमौसममेंअनिश्चितताआगईहै।इसकेअलावा,जलकीकमीभीएकबड़ीसमस्याहै, जिससेसिंचाईप्रभावितहोरहीहै। तकनीकीविकासकीकमी,उर्वरककीबढ़तीकीमतें,औरबाज़ारमेंकिसानोंकोसहीमूल्यनमिलनाभीकृषिकीचुनौतियोंमेंशामिलहैं।सरकारनेप्रधानमंत्रीकिसानसम्माननिधियोजना, मृदास्वास्थ्यकार्ड, औरफसलबीमायोजनाजैसेकदमउठाएहैं,परन्तुइनकाप्रभावसीमितहैऔरबड़ेसुधारोंकीआवश्यकताबनीहुईहै।	8	9	8

Since the information is organized in this manner, it becomes much easier to study and apply a variety of NLP techniques to the data. A more detailed overview the dataset**,** including its structure, linguistic properties, corpus composition, question types and lexicon details are provided in table 5.2.

Table 5.2: Dataset Overview for Automated Answer Scoring in Hindi

Dataset	Description
Number of Questions	2500 Questions
Reference Answers	Created by subject matter experts and textbooks, well-structured and detailed.
Language	Hindi
Corpus Size	28000 words
Corpus Sources	Sourced from academic textbooks
Question Types	Moderate
Reference Answer Characteristics	Reference answers include factual, analytical, and opinion-based responses to ensure diversity in scoring
Lexicon	Contains 15000 domain-specific words, including synonyms, antonyms, key terminologies, idiomatic expressions
Preprocessing of Data	Data cleaning, Tokenization, stop-word Removal, lemmatization
Subject Areas Covered	History
Educational Levels	Questions and reference answers collected from class 8^{th}-10^{th}.
Corpus Quality	High-quality, well-structured academic texts selected for corpus to ensure relevancy and consistency in scoring
Human Scores	Each student answer is scored by three evaluators on a scale of 1 to 10

Thus, we can use the clear strengths of our data to guide the development of our scoring model. Because assessment topics and answers are varied, our automated system is able to adapt to different aspects of educational language assessment

5.3 Evaluation of Pre-processing and Feature Extraction

Why we need to do both syntactic and semantic feature extraction is to improve the precision of automatic answer scoring. The system can analyze the structure and main ideas within a text, so that it can evaluate the correctness and relevance of different answers. Using modules for both syntactic and semantic features is very important for improving how the model judges Hindi text replies. To create an effective assessment system for student answers, we are working with both the way answers are organized and what they represent. Thorough preparation of the data was undertaken before feature extraction to ensure the achieved results. Before analyzing a dataset, it must be made ready for extraction by preprocessing, so that results are efficient and valuable.

A series of significant steps was used in the preprocessing stage. At first, we conducted cleaning and tokenization, helping us focuses on each component more easily. To remove unnecessary words, we applied stop-word removal to the sentences. This refinement resulted in clearer information, so the model centered on important terms included in scoring. Also, we applied lemmatization to bring words to their basic form, making it possible to use these words consistently for good semantic analysis.

Pre-processing allowed us to collect syntactic and semantic features which we used to understand the meaning and links among different words. The patterns were found by analyzing how the sentence was structured which revealed how words are related within the replies. Because all these features were combined, the characteristics of the answers could be examined more carefully and the scoring model was made more accurate.

5.3.1 Output of Pre-processing Steps

To create a clear and concise paragraph summarizing the preprocessing steps and their results, we can outline the key aspects in a table format. The tables explain how each stage in data preprocessing affects the data. This illustrates how the text evolves through the preprocessing pipeline.

Table 5.3: Transformation of Hindi Text Data through Preprocessing Steps

Pre-Processing Steps	**Example text** **(Sample from dataset)**	**Description**
Before **Pre-processing**	महात्मागांधीनेभारतीयस्वतंत्रतासंग्राममेंअहिंसकआंदोलनोंकानेतृत्वकिया।उन्होंने सत्याग्रह, असहयोगआंदोलन, सविनयअवज्ञाआंदोलनऔरभारतछोड़ोआंदोलन जैसेप्रमुखआंदोलनोंकेमाध्यमसेअंग्रेजोंकेखिलाफलड़ाईलड़ी।गांधीजीकाउद्देश्य थाकिबिनाकिसीहिंसाके, केवलसत्यऔरअहिंसाकेबलपरब्रिटिशशासनकोसमाप्त कियाजाए।उन्होंनेलोगोंकोआत्मनिर्भरताऔरस्वदेशीवस्त्रोंकेइस्तेमालपरबलदिया औरखादीकाप्रचारकिया।उनकीनीतियोंऔरआंदोलनोंनेभारतीयजनतामेंआजादी कीललकजगाईऔरभारतीयस्वतंत्रतासंग्राममेंउनकामहत्वपूर्णयोगदानरहा।	Original text contains complete sentences with punctuation and irrelevant characters,
After Cleaning	महात्मागांधीनेभारतीयस्वतंत्रतासंग्राममेंअहिंसकआंदोलनोंकानेतृत्वकियाउन्होंने सत्याग्रहअसहयोगआंदोलनसविनयअवज्ञाआंदोलनऔरभारतछोड़ोआंदोलनजैसे प्रमुखआंदोलनोंकेमाध्यमसेअंग्रेजोंकेखिलाफलड़ाईलड़ी गांधीजीकाउद्देश्यथाकि बिनाकिसीहिंसाकेकेवलसत्यऔरअहिंसाकेबलपरब्रिटिशशासनकोसमाप्तकिया जाए उन्होंनेलोगोंकोआत्मनिर्भरताऔरस्वदेशीवस्त्रोंकेइस्तेमालपरबलदियाऔर खादीकाप्रचारकियाउनकीनीतियोंऔरआंदोलनोंनेभारतीयजनतामेंआजादीकील लकजगाईऔरभारतीयस्वतंत्रतासंग्राममेंउनकामहत्वपूर्णयोगदानरहा	Removes punctuation and extraneous character. It improves data quality by reducing noise
AfterTokenization	[महात्मा, गांधी, ने, भारतीय, स्वतंत्रता, संग्राम, में, अहिंसक, आंदोलनों, का, नेतृत्व, किया।, उन्होंने, सत्याग्रह, असहयोग, आंदोलन, सविनय, अवज्ञा, आंदोलन, और, भारत, छोड़ो, आंदोलन, जैसे, प्रमुख, आंदोलनों, के, माध्यम, से, अंग्रेजों, के , खिलाफ, लड़ाई, लड़ी।, गांधीजी, का, उद्देश्य, था, कि, बिना, किसी, हिंसा, के, केवल, सत्य, और, अहिंसा, के, बल, पर, ब्रिटिश, शासन, को, समाप्त, किया, जाए।, उन्होंने, लोगों, को, आत्मनिर्भरता, और, स्वदेशी, वस्त्रों, के, इस्तेमाल, पर, बल, दिया, और, खादी, का, प्रचार, किया।, उनकी, नीतियों, और, आंदोलनों, ने, भारतीय, जनता, में, आजादी, की, ललक, जगाई, और, भारतीय, स्वतंत्रता, संग्राम, में, उनका, महत्वपूर्ण, योगदान, रहा।]	Text is split into tokens, enabling structured analysis at the word level.
After Stop word Removal	महात्मागांधीभारतीयस्वतंत्रतासंग्रामअहिंसकआंदोलनोंनेतृत्वकिया।उन्होंनेसत्या ग्रहअसहयोगआंदोलनसविनयअवज्ञाआंदोलनभारतछोड़ोआंदोलनप्रमुखआंदोल नोंमाध्यमअंग्रेजोंखिलाफलड़ाईलड़ी।गांधीजीउद्देश्यबिनाहिंसाकेवलसत्यअहिंसा बलब्रिटिशशासनसमाप्तजाए।उन्होंनेलोगोंआत्मनिर्भरतास्वदेशीवस्त्रोंइस्तेमालब लखादीप्रचारकिया।नीतियोंआंदोलनोंभारतीयजनताआजादीललकजगाईभारतीय स्वतंत्रतासंग्राममहत्वपूर्णयोगदानरहा।	Removed common stop words, focusing on more meaningful content and reducing noise.
After Lemmatization	महात्मागांधीभारतीयस्वतन्त्रतासंग्रामअहिंसक**आंदोलन**नेतृत्वकर**वह**सत्याग्रहअ सहयोगआंदोलनसविनयअवज्ञाआंदोलनभारतछोड़ोआंदोलनप्रमुख**आंदोलन**मा ध्यम**अंग्रेज**खिलाफलड़ाई**लड़**गांधीजीउद्देश्यबिनाहिंसाकेवलसत्यअहिंसाबलब्रिटि शशासनसमाप्त**जावहलोग**आत्मनिर्भरतास्वदेशी**वस्त्र**इस्तेमालबलखादीप्रचारक र**नीतिआंदोलन**भारतीयजनताआजादीललक**जगा**भारतीयस्वतन्त्रतासंग्राममह त्वपूर्णयोगदान**रह**	Converted tokens to base forms, simplifying vocabulary while preserving essential meaning.

The dataset was carefully processed so the text was suitable for analysis. All of the steps were performed within Jupyter and a quick look at some of the changes is presented in Table 5.3. First, the dataset was enhanced by removing unnecessary punctuation and characters. Next, tokenization divided the clean text into words or tokens to make the data easier to examine at the word level. Subsequently, stop-word removal got rid of the regular words "यह" and "से" as they weren't important to the content. Then, each word was changed to its basic form which fixed variety and simplified the words used. Libraries such as Pandas and NLTK make it easier and quicker to manage data from languages and text. Thanks to preprocessing, both the effective structure of the dataset and the accuracy of scoring increased since unnecessary text matter was filtered out.

Once the data was preprocessed, we picked out syntactic and semantic features to analyze Hindi text responses in detail. Therefore, the model studies not only the main points of the response but also the way grammar links those ideas. The following tables present the results of feature extraction, showing how every feature helps to analyze the linguistic complexity, significance and meaning of the responses.

5.3.2 Syntactic Feature Output

After preprocessing, syntactic features were pulled out to study the grammar and structure of student responses. They provide knowledge about how complex the sentence is, check the structure and test for proper grammar, all of which matter in judging the correctness of an answer. Some of the techniques are measuring word and sentence lengths, applying part-of-speech (POS) tagging and dependency parsing. Following is a description of what each feature offers.

- **Sentence Length Analysis**

An automated assessor can use both long and short sentences as an indicator of the difficulty and quality of student replies. Sometimes, sentences that are longer express ideas more fully and with more context, while those that are shorter may tell you the writer does not want to get very detailed. Looking at how long each sentence is can help understand how well students are expressing their knowledge. Tables 5.4 and 5.5 gives information about sentence lengths and summary statistics of sentence length for questions, reference answers and student responses.

Looking at how sentences are arranged in texts reveals useful information about student writing skills. By looking at the sentence lengths, we can spot differences in how complex and expressive the examples are. Text with longer sentences often explains things more thoroughly, compared to text with lots of little sentences that could reflect a lack of content or complexity. They provide a way to measure how effectively the question has been answered and to group responses as high- or low-scoring based on their structure.

Table 5.4: Sample of Sentence Length Output

Question Number	Question length	Reference Answer length	Student Answer length
1	9	22	11
2	9	14	11
3	7	12	11
4	6	18	14
5	6	13	9
…	…	….	….
…	…	….	….
2497	4	10	7
2498	4	12	7
2499	6	11	7
2500	5	11	8

Table: 5.5: Descriptive Statistics of Sentence Lengths

	Question	Reference answer	Student answer
Average length	5.43	15.5	11.18
Standard Deviation	1.11	10.79	8.73

By looking at sentence lengths, we can see some key trends in the data. The references answers have an average length of 15.5 words, showing they are typically much longer and more elaborate than student answers which average 11.18 words. It reveals the differences in sentences lengths from one type to another. If the standard deviation in questions is small, most of the questions have similar lengths. On the other hand, larger standard deviation values for Reference and student answers mean the responses are more varied in their length. Most of the time, the answers teachers see are richer than the answers their students come up with if students' answers are shorter and less uniform in length, it

could mean their ideas are simpler and may affect their grades. It allows us to see the depth of their answers and guides us when helping them improve their writing.

Word Length Distribution

The length of words used is a strong measure of how advanced and varied a person's vocabulary is. Using long words may indicate a wide vocabulary, resulting in higher quality and more informative writing. Looking at the length of words in student answers helps us determine how complex and rich with vocabulary the replies are.

Tables 5.6 and 5.7 show sample data and key statistics on word length for Hindi texts which are made up of questions, reference answers and student answers. They help us find out if there is a usual pattern of higher-scoring answers using longer and more complicated words. Longer words in reference answers suggest that the material is written for a mature audience, while short words in answer from students may suggest they are expressing themselves in more straightforward ways. This approach helps produce a better and more detailed judgment of writing skills, necessary in the world of automated scoring.

Table 5.6: Sample of Word Length Output

Question Number	Question length	Reference Answer length	Student Answer length
1	5.89	5.32	5.36
2	4.00	5.50	4.18
3	4.43	4.75	3.27
4	5.83	4.22	3.88
5	5.00	4.92	4.78
…	…	…	…
…	…	…	…
…	…	…	…
2497	4.75	4.80	4.00
2498	4.75	4.83	3.14
2499	5.33	4.64	5.00
2500	5.20	4.27	4.00

Table 5.7: Descriptive Statistics of Word Lengths

	Question	Reference answer	Student answer
Average length	5.16	4.69	4.18
Standard Deviation	0.75	0.75	0.75

By looking at descriptive statistics for word lengths, we can learn interesting things about the vocabulary used in both the reference and student answers. On average, a reference answer is made up of 4.69 words while a student answer is made up of 4.18 words. It appears that students are choosing similar words when answering the questions which are similar to the reference answers. The same standard deviation of 0.75 for both data groups suggests that words used in both answers show a similar range of lengths. Since there is a similarity, most students still use words that have about the same number of characters as those in the references.

Part of Speech (POS) Tagging Analysis

Part-of-Speech (POS) analysis plays a vital part in identifying the language features of a piece of text. Thanks to POS tagging, it is possible to study and analyze grammar patterns in answers provided by students and those in the reference texts. Looking at the number of nouns, verbs, adjectives, pronouns and conjunctions in a person's reaction allows us to gauge how varied its use of words is. Lots of nouns and action verbs could show that the student is expressing what is important and what is happening clearly. The addition of adjectives and adverbs makes a piece of writing more detailed and using prepositions and conjunctions properly can unite the sentences more effectively.

In automated answer scoring, the distribution of POS tags helps separate good answers from those that are less well-developed. Table 5.8 includes a sample of POS-tagged tokens turned into vectors from the dataset, so you can see how the words in the two sets of answers are used. It reveals the main topics, the level of description and the main strategies applied to communicate. Analyzing the frequency of each POS category gives us a clearer idea of the text's style. By turning POS tags into vectors, we can feed them to machine learning models to help us understand and predict how rich and varied the language is in the training text. Combining these POS tag analyses and their distribution helps us see the message and communication style of the text in a clearer way.

Table 5.8: Sample of POS Tag Output

Question Number	Question POS Vector	Reference Answer POS Vector	Student Answer POS Vector
1	[0, 3, 4, 0, 0, 0, 1, 0]	[1, 4, 10, 0, 1, 0, 2, 0]	[1, 2, 4, 0, 0, 0, 2, 0]
2	[0, 2, 3, 0, 0, 0, 3, 0]	[0, 1, 10, 1, 0, 0, 1, 0]	[0, 1, 6, 0, 0, 0, 1, 2]
3	[2, 3, 0, 0, 0, 0, 1, 0]	[2, 9, 0, 0, 0, 0, 1, 0]	[1, 6, 0, 0, 0, 0, 1, 2]
4	[0, 0, 5, 0, 1, 0, 0, 1]	[2, 6, 3, 0, 0, 0, 2, 2]	[0, 0, 0, 0, 0, 0, 0, 0]
5	[0, 3, 0, 0, 0, 0, 2, 1]	[1, 2, 8, 0, 0, 0, 0, 1]	[0, 5, 0, 1, 0, 0, 1, 1]
	…	…	…
	…	…	…
2497	[0, 3, 0, 0, 0, 1, 0, 0]	[2, 6, 0, 0, 0, 0, 0, 1]	[1, 2, 0, 0, 1, 0, 1, 1]
2498	[0, 3, 0, 0, 0, 1, 0, 0]	[3, 4, 0, 0, 1, 0, 1, 1]	[0, 2, 0, 0, 0, 0, 1, 1]
2499	[1, 3, 0, 0, 0, 0, 1, 0]	[2, 4, 0, 0, 0, 0, 2, 2]	[1, 4, 0, 0, 0, 0, 0, 1]
2500	[0, 4, 0, 0, 0, 0, 0, 0]	[2, 4, 0, 0, 0, 0, 2, 1]	[2, 2, 0, 0, 0, 0, 1, 1]

Figure 5.1 illustrates the frequency distribution of various Part of Speech (POS) tags in the analyzed text. This visual representation highlights the prevalence of different grammatical categories, offering insights into the linguistic structure of the content.

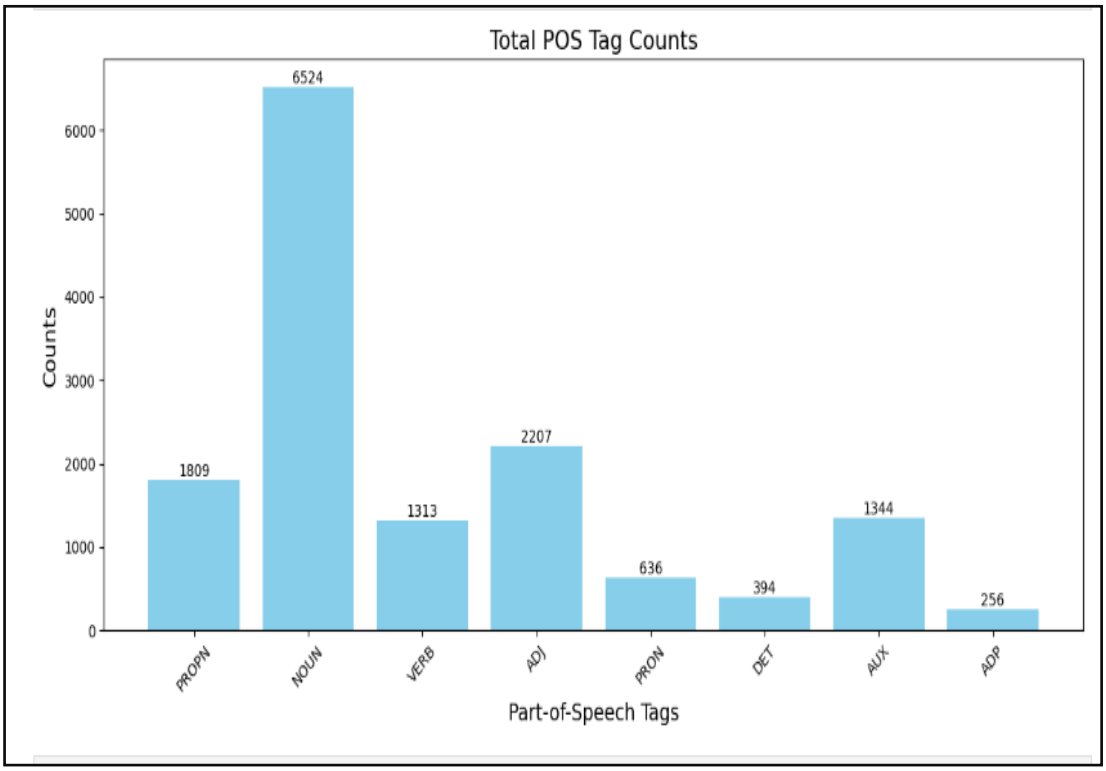


Figure 5.1: Frequencies of Part-of-Speech Tags in Analyzed Text

Results from part-of-speech tag analysis show that nouns are the most common tags, with around 6000 occurrences which highlights the importance of nouns in the analyzed writing. It means that the subject matter seems to involve numerous entities, objects or concepts, all important for its interpretation. Nouns were followed by a large number of proper nouns (PROPN) and adjectives (ADJ). When there are lots of proper nouns, the focus is often on telling us about specific people, companies or places which matter in the content. The meaning is also enhanced by the inclusion of adjectives, though they appear less frequently. Because there are many nouns, it seems that this text helps to explain important subjects and objects. Because there are so many proper nouns, the work may inform about special events or people, making it easier for readers to relate to the information. This could especially matter when working with literary materials, historical texts or when talking about major figures or bodies. Although there are fewer adjectives, they indicate that the author wants to give detailed and interesting descriptions to the nouns. As a result, the text tries to teach and entertain the reader by describing things in colorful ways that make them easy to see and understand.

Dependency Parsing Analysis

The best way to analyze the grammar of sentences is by using dependency parsing, as it looks at how words are related to each other. It studies the relationships between the distinct words in a sentence and brings to light the subject-verb-object patterns, modifiers and complements. The process of dependency parsing creates a structure for sentences, making it clear what different words are contributing to the idea being shared. This approach explains the complex grammar of the text, while also being key when assessing the linguistic aspects of the student and example responses.

After the results are parsed, they are expressed as vectors to help with quantitative work. By making the sentences easier to compare and organize, it helps analyze the patterns in grammar. Figure 5.2 shows a bar graph that illustrates the different dependency relations counted in the texts that were looked at. Each bar represents a grammatical relation, cleary showing how many times these relations exist in the data. With this information, it becomes easier to spot common patterns in the replies which could relate to both the answers' quality and how complex the language is.

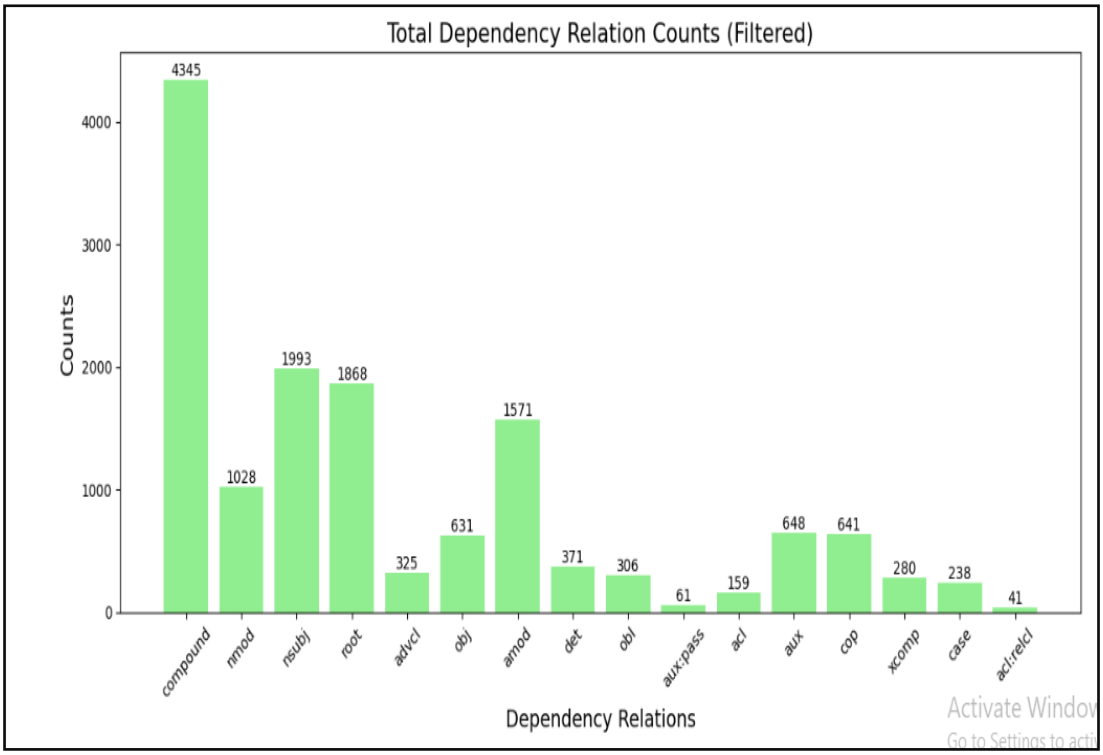


Figure 5.2: Frequencies of Dependency Relation in Analyzed Text

Compound relations were found to be the most frequent, with an estimated number of 4,000 occurrences on the bar graph. Such a high frequency demonstrates that compound nouns or phases are often used which shows that the text contains a wide range of complex noun phrases. Both compounds and structures called nsubj and root exist about 2,000 times in the corpus. Therefore, subjects and main verbs are very important in organizing the sentence because they are central to what is happening in the conversation. These relations help ensure that the text keeps its subject-verb order, making it easy to follow. The amod (adjectival modifier) relation which occurs about 1,500 times, points to the way adjectives modify nouns. Therefore, making sure to add extra descriptions helps the reader gain deeper insight into the things mentioned in the text. However, the remaining of these relationships such as nmod (nominal modifier), advl (adverbial modifier), obj (object) and aux (auxiliary), are hardly seen in the data. So, although they fit into the bigger picture of the sentence structure, they are slightly less noticeable than the main subjects and the compound structures. Since there aren't many examples of these relations, it could show that the writings focus more on building up the main sentence rather than adding extra detail or complication. An analysis may reveal how well the text expresses its ideas and information, how easy it is to follow and how effective it is as a communication.

We can analyze the level of clarity and sophistication in what students say by noticing the length of the sentences and the words they use. Using POS tag vectors, we can see the types and mix of nouns, verbs and adjectives in language, helping us understand how language is used well in communication. In addition, the high number of dependency relations helps identify the types of sentences and patterns that many students use, making it easy to point out main strengths and weaknesses in their writing style. All of these aspects work together to allow for a proper scoring system, giving students actionable feedback and helping them improve.

N-gram (Bigram) Analysis

One can use n-grams (especially bigrams) to find out which word pairs appear most often in a specific text. It makes it easier to discover the key themes, how persons write and the way they use language in answers. Looking at the most common bigrams in each data set allows us to notice how much the student responses resemble both the questions and the model responses. The analysis helps you see what vocabulary and grammar is used the most and which words relate to the topic being discussed. The keywords that appear in the question sets, answers and student answers are highlighted in Table 5.9.

Table 5.9: Top Five Bigram of question, Reference Answer and Student Answer

Category	Bigram	Frequency
Question	महत्वक्या	124
	भारतीयसमाज	118
	आपअनुसार	109
	क्याप्रभाव	27
	जलवायुपरिवर्तन	15
Reference Answer	आवश्यकहै	33
	बढहै	27
	सुधारहै	26
	जलवायुपरिवर्तन	18
	पर्यावरणसंरक्षण	16
Student Answer	मिलहै	38
	बढहै	30
	सुधारहै	29
	पर्यावरणसंरक्षण	22
	जलवायुपरिवर्तन	20

In the questions, the most frequent bigrams “महत्वक्या” (124), “भारतीयसमाज” (118), and “आपअनुसार” (109)—suggest that the prompts commonly focus on understanding importance, personal interpretation, and social issues. Other frequent bigrams such as “क्याप्रभाव” (27) and “जलवायुपरिवर्तन” (15) reinforce the evaluative and analytical nature of the questions.In the student answers, top bigrams include “मिलहै” (38), “बढहै” (30), and “सुधारहै” (29). These indicate the use of simple, often repetitive sentence structures. While bigrams like “पर्यावरणसंरक्षण” (22) and “जलवायुपरिवर्तन” (20) show that students are referring to the relevant topics, their language tends to rely on basic constructs, possibly lacking depth or descriptive variation.

On the other hand, reference answers exhibit more formal and informative patterns. High-frequency bigrams such as “आवश्यकहै” (33), “पर्यावरणसंरक्षण” (16), “जलवायुपरिवर्तन” (18), “बढहै” (27), and “सुधारहै” (26) indicate a richer and more consistent use of academic and content-specific vocabulary. The presence of overlapping bigrams like “पर्यावरणसंरक्षण” and “जलवायुपरिवर्तन” in both student and reference answers suggests topic relevance; however, the frequency and contextual usage show that reference answers are more structured and elaborate.This comparison highlights how bigram patterns can reflect the depth of language use, topic understanding, and coherence in student responses, making N-gram features a useful tool for evaluating content alignment and writing quality in automated scoring systems.

5.3.3 Semantic Feature Output

Even though syntactic analysis is required, semantic features are needed to determine the effectiveness of the communication and how well the data makes sense. The semantic features found in these methods consist of word embedding that highlight the meaning of words in their sentences and sentence embedding that are responsible for describing how entire sentences are understood. Words in the dataset are considered important in TF-IDF which forms a part of frequency-based methods. To do corpus-based analysis, one must look for ambiguous expressions, detect multiple meanings in words, analyze the meaning of polysemic words to see if they help make the text clear, study the flow of the discussion in the text and use Latent Semantic Analysis (LSA) to study how words are related. Below, we go into more detail about these features and how they contribute to our understanding of the semantic richness in the responses.

Word and Sentence Embedding

To determine how similar two answers are, embedding tools analyze the words and sentences in each answer. Embedding convert word and sentence information into numbers that indicate their context and meaning. To generate word vectors for each word, we used either Word2Vec, FastText, GloVe or IndicNLP for Hindi and from this, we observed that certain words form groups with relative meanings. All the words in the answer were transformed into 100 dimensional vectors. Still, since word embedding only reflects the meanings of one word at a time, sentence embedding is a better method for judging the overall coherence of student answers. Emphasizing sentence embedding allows us to judge whether communication follows a logical pattern and how students express their knowledge in the subject. Since word embedding is so complex, it's tricky to analyze them, while sentence embedding is much easier to look at. Following this, sentence embedding help condense whole answers into one vector. To produce sentence embedding from word vectors, the word vectors for every word in the sentence are averaged to create a single vector for the entire sentence's meaning.

Sentence embeddings produced on the basis of word vectors produce high-dimensional representations (100 dimensions), which cannot be readily interpreted and might include redundant or correlated data. Principal Component Analysis (PCA) as a dimensionality reduction method was used to overcome this problem. PCA alters the original embedding space by converting it to a set of smaller uncorrelated components whilst maintaining the greatest possible amount of variance in the data

There are three main reasons why PCA should be used. First, it minimizes the computational and noise levels of high-dimensional sentence embeddings. Second, it allows visualizing semantic differences and similarities among the answers of students effectively when they are projected onto the lower-dimensional space. Third, the main elements include the greatest semantic differences between responses, hence constructive statistical analysis of similarity of answers.

The two main components (PCA1 and PCA2) were examined in the given study since they explain most of the variance in the sentence embeddings. The descriptive statistics such as the mean, standard deviation, minimum and maximum values of PCA1 and PCA2 (Table 5.10) were calculated and included information about how the semantic representations of student responses were dispersed and distributed.

The resulting embedding was subjected to Principal Component Analysis (PCA) to reduce dimensionality and visualize the data effectively. The descriptive statistics for the first two principal components (PCA1 and PCA2) Table 5.10 contains the average, standard deviation, smallest value and largest value for every component.

Table 5.10: Descriptive Statistics for first two principal components

	Question PCA1	**Question PCA2**	**Reference PCA1**	**Reference PCA2**	**Student PCA1**	**Student P CA2**
Count	2500	2500	2500	2500	2500	2500
Mean	-0.33	-0.01	0.07	0.01	0.26	0.00
SD	0.34	0.01	0.35	0.01	0.37	0.01
Min	-2.20	-0.04	-2.20	-0.01	-2.20	-0.02
Max	0.79	0.04	0.72	0.04	1.10	0.04

Results from the Principal Component Analysis indicate what is important in terms of semantic meaning for sentences in the dataset. Because categories have the same number of entries, it can be seen that students use a wider range of words when responding, since the mean PCA1 value for student choices is slightly higher than that for questions and reference answers. This implies that students' answers can include more diverse ideas or concepts. Because the standard deviations are similar for both questions and reference answers, the variation is not high, but the standard deviation in student responses from PCA1 illustrates that their meanings are not always close. It is also clear from these values that some student responses express the concepts in fewer semantically similar ways, while a few responses reach the highest level of expression, scoring 1.10. In essence, these results underline that questions, references and students' answers use language differently which may explain why PCA is capable of understanding and encoding the numerous forms of meaning found in the texts.

The PCA plot in figure 5.3 illustrated the distribution of sentence embedding, revealing clusters that indicate varying levels of semantic similarity among student responses. For instance, responses that shared common themes or vocabulary tended to group closely together, while those that diverged in content or expression were positioned further apart. Such findings suggest that sentence embedding can reflect the precise meanings in the data which aids in judging how well the answer fits what is mentioned in the question.

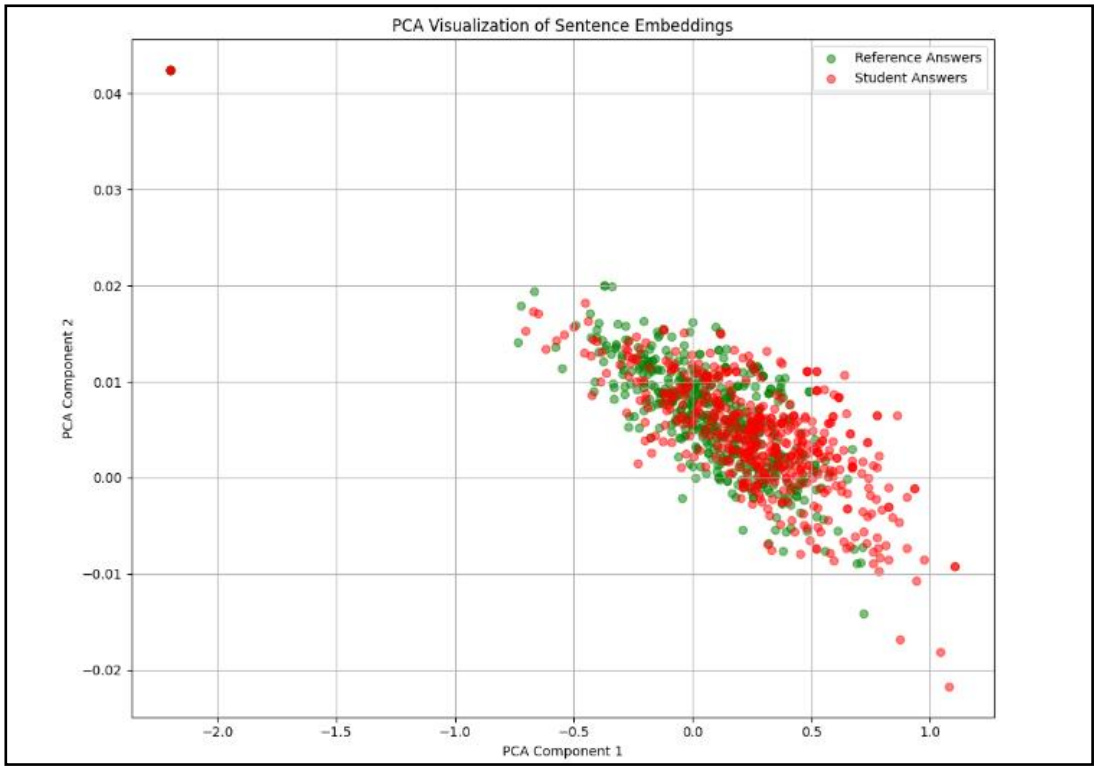


Figure 5.3: Semantic Relationships using PCA of Sentence Embedding

As a result of the plot, it is possible to distinguish groups of student answers from the groups of reference answers. I note that the student and reference answer points are often very close to one another, indicating a high level of similarity between student and reference answers. Since questions appear close in terms of scoring, students seem to know the suggestions the questions are looking for.

Polysemy and Ambiguity Detection

When assessing polysemy and ambiguity in these works, we checked how frequently these works used words that can have more than one meaning. It reveals that using words with multiple meanings may lead to difficulties when judging the response's quality and clarity. When students use polysemy and ambiguity, the language they write can be unclear; therefore, teachers should look at the context to understand what they mean. If a word has several meanings that are related to one another, this is called polysemy. Ambiguity occurs when words make it unclear what is meant because their meaning can be understood in several ways without much help from the context. It helps to look for polysemy or ambiguity in student responses, since this shows areas where a word or phrase can be misunderstood.

Table 5.11 includes a sample of polysemous words used by students in their answers to help understand the amount of variability in these features.

Table 5.11: Sample of polysemous word and Ambiguous Word Count Output

Answer Number	Polysemous Word Count	Ambiguous Word Count
1	2	7
2	0	5
3	1	1
4	0	0
5	0	1
…	…	…
…	…	…
2497	0	2
2498	0	2
2499	2	5
2500	2	0

Table 5.12: Summary of Polysemous Word Count

Metric	Value
Total Number of Question Sets Analyzed	2500
Number of Sets with Polysemy Detected	293.00
Total Polysemous Words Found	951.00
Average Polysemous Words per Set	1.90
Percentage of Sets with No Polysemy Detected	41.40
Highest Number of Polysemous Words in a Set	22.00

Out of all the answers in Table 5.12, almost 60% or 293, have at least one polysemous word, for a total of 951 such instances. Most responses feature an average of 1.9 words with multiple meanings, but in 41.4% of responses, none of the words have multiple meanings. As you can tell from the table, polysemous words are at their highest in any one answer, reaching 22. Polysemy can be seen in the answers of many students, as words with several meanings appear in their responses.

Table 5.13: Summary of Ambiguous Word Count

Metric	Value
Total Answer	2500
Total Words in Student Answers	5593
Total ambiguous word identified	2345
Unique ambiguous word identified	424
Non-Ambiguous Words	3234
Ambiguous word per response	4.69

It is clear from the data in the table that across 2500 survey responses, a total of 2,345 ambiguous words were found, along with 424 different ambiguous terms. Since the average number of ambiguous words in responses is 4.69, the analysis indicates that some language sections could mislead readers in grading the quality of the writing. Figure 5.4 and Figure 5.5 also present how polysemous and ambiguous words are found in all the responses, allowing us to see how widespread they are.

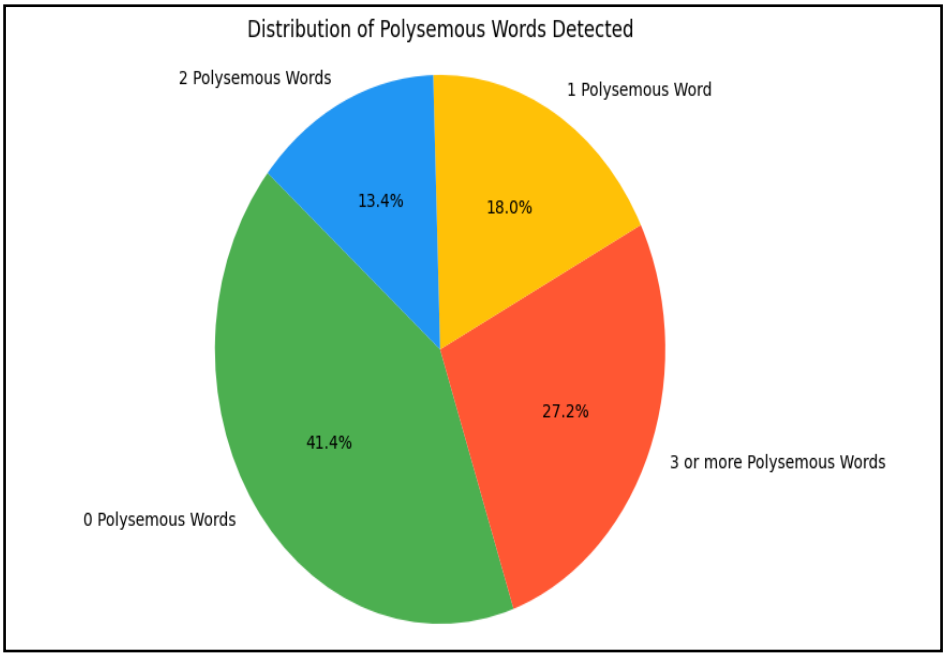


Figure 5.4: Distribution of polysemous word

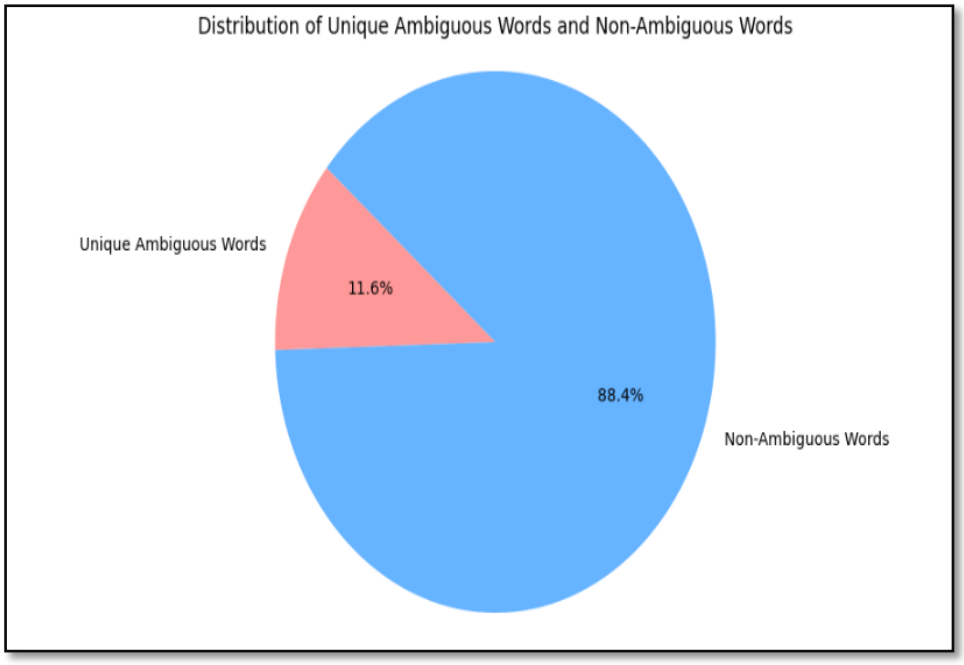


Figure 5.5 Distribution of Ambiguity Word

The figures reveal the frequency of polysemous and ambiguous words among student responses. Compared to other results, 41.4% of the students did not use polysemous words in their answers, suggesting a clear choice of language for most respondents. Polysemous words are included in 18% of responses with only one such word, in 13.4% with two and in 27.2% with three or more, suggesting that the responses vary in word complexity. Using polysemy shows that students choose different words and some of those words have various meanings.

Conversely, only 11.6% of all reports have a unique ambiguous word and the rest are completely clear. Even though there is some ambiguity, it is used less often and mostly occurs in a small part of people's answers. Putting these together, we see that although a lot of student replies use simple words, there remain responses that could be interpreted in more than one way which could impact marking and scoring.

Text Coherence

If the text flow is coherent, it becomes easier to understand the messages being delivered. In each case, coherence is calculated between each student answer and the correct answer, so we can understand how well the structure and content of the reply matches. The average, maximum, minimum and standard deviation of the coherence scores are available in Table 5.14. They allow you to compare student responses with the reference and measure the

consistency of responses. Furthermore, Figure 5.6 displays a graph of coherence scores that outlines the occurrence of each level of coherence in the data. When compared, Table 5.14 and Figure 5.6 clearly show how coherently students keep to the topic and format.

Table 5.14: Statistics of Coherence Scores between Reference and Student Answers

Statistic	Coherence Score
Average	0.80
Maximum	1.00
Minimum	-0.05
Standard Deviation	0.14

Generally, students' answers agree with the reference answers, as the average coherence score is about 0.80. So, the majority of students seem to be well-versed in the subject and learn its key points effectively. When the maximum coherence score is 1.0, it proves that one student came up with an answer that perfectly matches the expected answer, proving they understood the question well. Since the coherence score is at approximately -0.05, it means that one student gave an answer that is quite different from the reference. The research reveals that the coherence scores vary moderately, since the standard deviation is about 0.14. The majority of students did well, but a few outside scores indicate that some of them do not fully grasp the material as much as their peers.

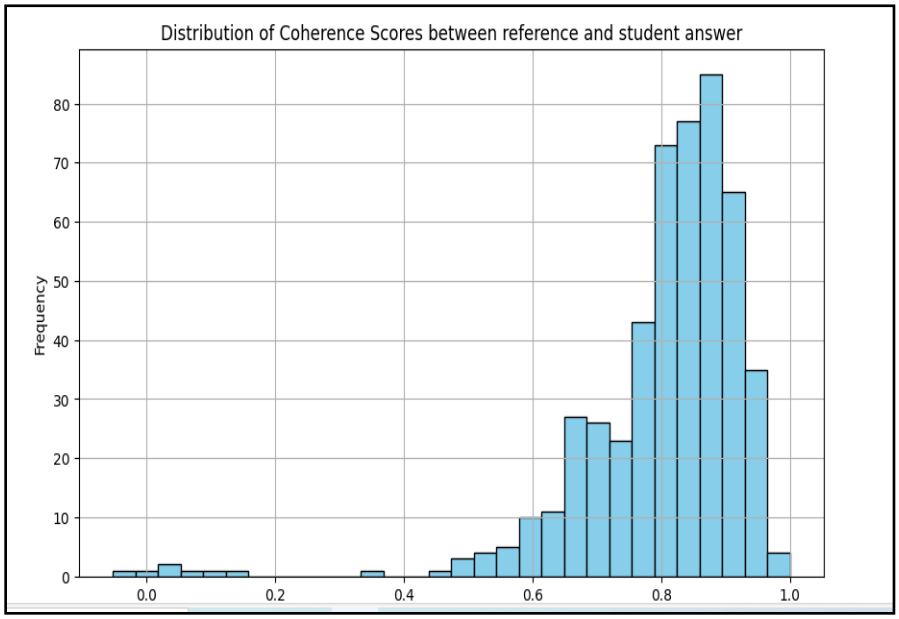


Figure 5.6: Frequency Distribution of Coherence Scores between Student and Reference Answers

If a score is above 0.5, that means students understood the ideas well, even when they expressed the ideas differently. Most students were able to explain the material well in their papers. This says that the way they responded was similar to the answers we would expect them to give. Still, scores under 0.5 present a concern and should be carefully considered as well. Therefore, it may be necessary to review the idea with students to ensure they are clear on the subject.

Latent Semantic Analysis (LSA)

LSA is a method employed for detecting the hidden relationships between words and phrases in a set of data. LSA is used in the analysis to check if student and reference answers share the same meaning by converting the wording into a series of vectors. In this calculation, distance is found by using the Euclidean formula to compare the vector representations of the answer with a reference answer. To put it another way, when the distance is small, the guessed answer is more like the reference answer. Due to its accuracy, the Euclidean distance is widely used for scoring students automatically on tests and assignments.

In Table 5.15, the readings of the Euclidean distance indicator are shown with their average, maximum, minimum and standard deviation. They provide an overall sense of how closely the students' answers resemble the right answers.

Table 5.15: Descriptive Statistics for Euclidean Distances between Reference Answers and St udent Answers

Statistic	Euclidean Distance
Average	0.13
Maximum	0.48
Minimum	0.00
Standard Deviation	0.09

Most student responses are very similar to the reference answers, according to the results shown in the LSA feature space. This suggests that most students understood the topic and gave responses that reflect what was expected. This value of maximum distance implies that for every question, the student's answer is nearly half a unit removed from its reference answer in the feature space. Since the gap between the average and this distance is still high, it implies that some students' answers differed widely from what was expected. By doing this, you might see which students had difficulty with the curriculum. The value 0.000 means that one student's response and the reference answer share the same LSA features. It is clear that some students truly got the gist of the reference answers and clearly understood the material. Since the standard deviation is close to 0.092, the distances show only minor changes from one date to another. Most students' answers are found near the center, meaning they performed with little difference between their answers. It means that most of the student scores are not much different from the correct answers.

It is clear from the results that the majority of students answered successfully and their replies were close in meaning to the right answers. However, there were a few students whose responses revealed spots where the teaching could be improved. By looking at the solutions at the maximum distance, you can discover which students had trouble and use tailored approaches to help them better understand the subject. Additionally, Figure 5.7 shows how the LSA features are arranged for reference and student answers which reveal how similar these answers are in their meaning. An analysis of the responses helps to identify how well and how often students made correct answers.

Points are assigned to students in different colors which depicts how they are doing. If students' marks cluster together near the correct answer, this usually means that on average,

most students understood the subject. It helps you by showing you in visual form the trends in student and reference answers.

On one axis are the LSA qualities of the reference answers and on the other axis are the LSA qualities of the student answers. The position of every point on the scatter plot shows the relationship between a student's answer and the expected one. If the dots are close together, it means that many students gave answers matching the reference ones. This suggests that the meanings of the words are very similar.

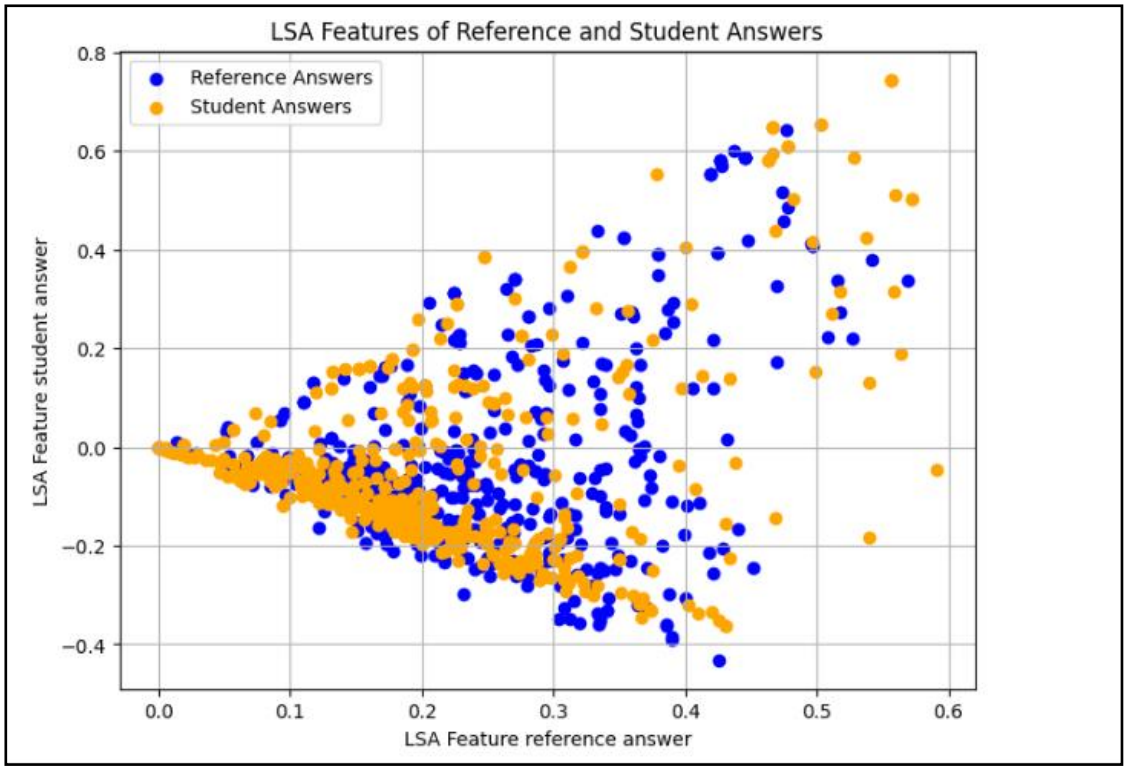


Figure 5.7: Distribution of LSA Features for Reference and Student Answers

If the points are scattered across the chart, it shows that the understanding of students varied a lot among them when responding to the reference answers. Many wrong answers come when points are widespread above or below the diagonal line (y = x), so the students' answers should be close to this line. If the responses come closer to the line, then the responses are likely to be more similar. Any answers that are distant from the diagonal or main group suggest that a student's answers were significantly different from the standard answers. They can reveal the areas where a student faced difficulties and may require direct attention.

5.4 Hybrid Modeling Framework for Automated Answer Scoring

With tasks in education and language being both challenging and time-consuming t o assess, experts depend on automated scoring systems. In order to reach accurate and similar grading results, the paper combines extracting linguistic features with both ML and DL techni ques. Thanks to this approach, student answers are explored from different angles and the sco ring system becomes stronger and more accurate. Capturing structure and meaning helps the f ramework to close the divide between manual and automatic evaluation. The method is given the name “hybrid” due to two facts: it process both syntactic and semantic content and it relie s on various modeling methods and applies them in an organized way. First, you must conduc t widespread extraction of linguistic features. Features in the structure of language such as wo rd and sentence length, POS tagging, dependency parsing and bi-gram patterns are applied for recognizing grammar and writing habits. They support checking the correctness and style of s tudents’ answers.

Semantically, the information is interpreted to discover the essential meaning and the correct context of the answers. Such techniques cover embedding words and sentences for understanding meaning, finding polysemy and ambiguity, using LSA to discover concept equivalence and checking for coherence between ideas and their presentation. Once the features are extracted, they are converted to numbers, made standard and joined into one feature vector that represents each response in its entirety. After that, the unified vector is sent into various Machine Learning and Deep Learning algorithms to predict the results. Among these Machine Learning models, SVR works well due to its ability to use margins in learning, Random Forest is popular for being an ensemble model and XGBoost is highly effective and can handle big datasets. In DL, CNNs detect similar patterns within the context of words and RNNs, LSTM, BiLSTM model how the text changes across its entire length.

To ensure clarity and reproducibility, all classifiers evaluated in Figure 5.8 are trained on the same standardized feature representations derived from the proposed framework. Traditional machine learning models (SVR, Random Forest, and XGBoost) serve as baseline predictors using handcrafted syntactic and semantic features. Deep learning models (CNN, RNN, LSTM, and BiLSTM) are employed to automatically learn representations from sequential textual embeddings. Hybrid models are explicitly designed to integrate deep feature representations with ensemble-based or sequential learning strategies, enabling complementary strengths in representation learning and decision-making.

Although all models use the same student answers as input, the data is represented in different forms depending on the classifier. For machine learning models such as SVR, Random Forest, and XGBoost, all extracted syntactic and semantic features are combined into a fixed-length numerical feature vector, which is directly used for prediction. For deep learning models, student answers are converted into sequences of word embeddings so that the order of words is preserved. CNN models apply one-dimensional filters over these word embeddings to capture local patterns such as important phrases, while RNN, LSTM, and BiLSTM models process the embeddings sequentially to learn temporal and contextual information across the entire answer. In hybrid models, deep learning networks are first used to learn meaningful feature representations, and these learned features are then passed to machine learning models to generate the final score.

We tried different ways of combining models, evaluated all of them and based our decisions on how well they performed using accuracy, precision, F-1score and recall. Figure 5.8 demonstrates the results of models in terms of their assessments from these metrics.

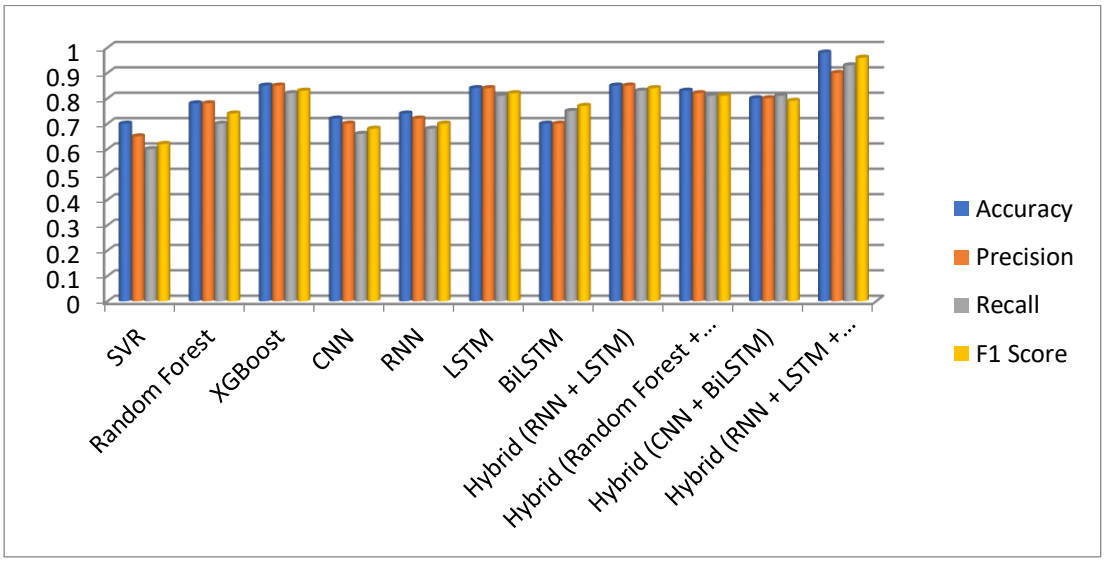


Figure 5.8: Performance Comparison of Deep Learning, Machine Learning and Hybrid Model

With the bar chart, it is easy to compare the performance of each model using multiple factors and quickly see how effective they are. Figure 5.8 implies that combining RNN, LSTM and XGBoost yields noticeably better performances in accuracy, precision and recall than using individual models. Because of using machine learning and deep learning, we are able to see trends everywhere and get the most out of our performance. It appears that

combining an RNN, LSTM and XGBoost yields a more reliable approach to automated scoring. The results show that using an ensemble approach helps evaluate students accurately.

The SVR model lags behind SVM in all metrics, suggesting that usual regression methods may not meet the challenges found in automated answer scoring. Likewise, hybrid models are considered more successful than RNN and CNN as they have a hard time spotting tough patterns in the data. With these results, hybrid models are the optimal choice for use in automated scoring because they performed better. Even if we hit limits on computation or clock time, Random Forest and SVR can still be important, as they allow for quick evaluation or preliminary tests. With the bar chart, it becomes possible to compare models and determine the plan for their use.

5.5 Performance of Automated Answer Scoring.

Since it was successful in every assessment, this piece of research deemed the model that includes Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and XGBoost to be the superior option for the automated scoring system. The scores generated by the model are checked here to find out if they are the same as those given by people. At the outset, we compare the scores for each answer's system response with those assigned by all the evaluators and pay attention to any regularity in the outcomes. A table has been included to show the results of the comparison, listing scores offered by three human evaluators and the corresponding scores given by the system. This data enables us to confirm how similar the evaluators and the automation are.

Table 5.16: Final Score Table for 2500 students answer

Student Answer	Score 1	Score 2	Score 3	System score
1	8.00	10.00	8.00	8.35
2	6.00	8.00	6.00	6.66
3	8.00	6.00	8.00	7.38
4	6.00	8.00	6.00	6.82
5	10.00	10.00	8.00	9.30
…..	…..	….	…	…

…..	…..	….	…	…
…..	…..	….	…	…
2497	6.00	6.00	7.00	6.59
2498	8.00	7.00	8.00	7.95
2499	9.00	9.00	8.00	8.70
2500	8.00	9.00	8.00	8.43

Furthermore, Figure 5.9 offers a line graph that shows how the scores of the system relate to the scores given by people. Because of this chart, spotting the progress of the scores, consistent results and any possible differences compared to human judgment is much simpler. By analyzing it, we can confirm whether the system is reliable and credible. If the model's outputs resemble those assigned by humans, this means it will perform effectively in actual scoring situations. As a result, automated evaluation gains trust and demonstrates how it could help more students with faster and fairer assessments if compared with grading by hand.

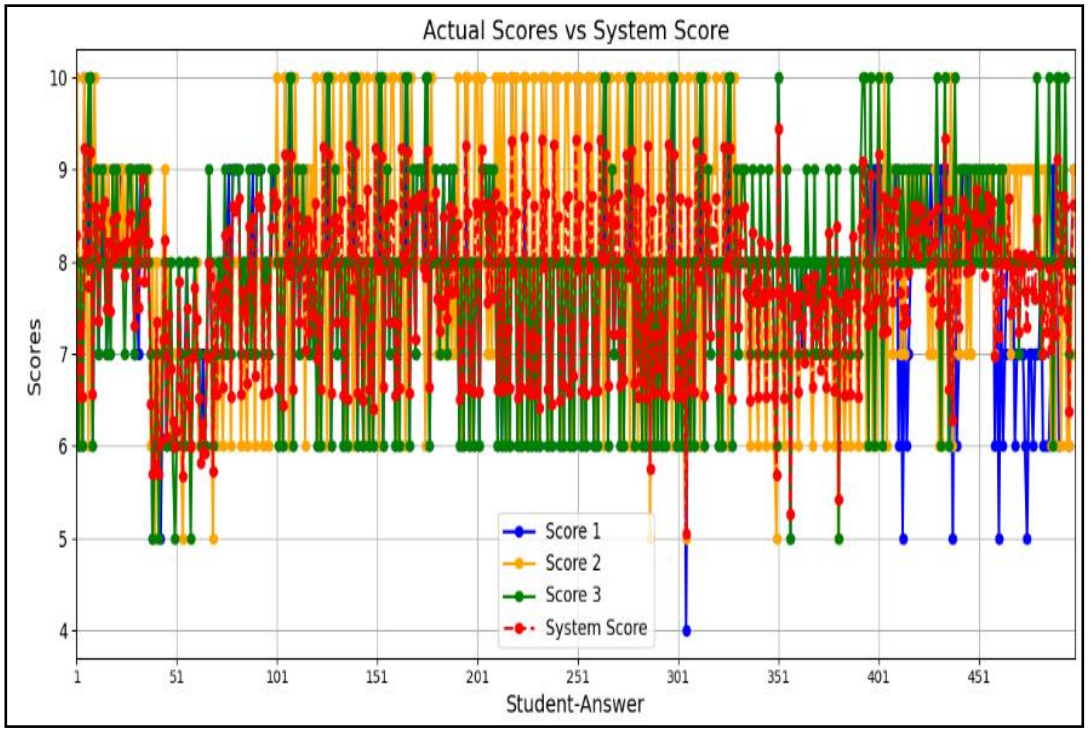


Figure 5.9: System Score vs. Human Score Visualization

From the line graph in Figure 5.9, it is clear that the computer's suggestions and those given by the evaluators agree closely in most cases. In some cases, the scores given by the system are not the same as those of single or more human evaluators. Although, in most cases, machine and human scores are similar, there may be occasional differences related to judgment or odd responses. Using the Pearson correlation coefficient, it was determined whether the system's ratings are accurately matching those given by our reviewers. It helps to see how accurately the system replicates the evaluation results of each person. The information is given in table 5.17.

Table 5.17: Correlation between System Scores and Human Evaluators

Correlation	System Score
Score-1	0.85
Score-2	0.65
Score-3	0.78

System scores and human scores are closely linked for Evaluators 1 and 3. This slightly different score from Evaluator 2 could be due to the way that evaluator assigns points. The analysis shows that the scores produced by the hybrid model match human opinions, supporting the usefulness of the model in scoring exams. Finally, to show how effectively our model forecast, we use its results on the evaluator scores to calculate the accuracy, precision, F1-score and recall. They allow users to understand the presentation level and dependability of the system for both scoring and being accurate.

Table 5.18: Performance Metrics Comparison between System and Human Evaluators

Metrics	**Human Score-1**	**Human Score-2**	**Human Score-3**
Accuracy	0.82	0.83	0.84
Precision	0.83	0.83	0.86
Recall	0.82	0.85	0.84
F1 Score	0.75	0.78	0.78

Across all three evaluators, the accuracy was between 82% and 84%. This suggests that the system is correct most of the time, following the human scoring system. Although the rise in accuracy from Human Score 2 to 3 is minor, it may indicate that the system works well with particular evaluators because their ways of scoring are closer to the automated model's u

nderstanding of answers. It is most accurate when the system uses Human Score 3 (86.73%) which means it is very precise in predicting a particular group. Therefore, Human Score 3 cou ld be a better fit for the system since it matches its scoring pattern. Even though Human Score 1 has decent recall, it also results in more false positives than Human Score 3. This means that the system does not capture exactly the same classes as Human Score 1.

With a recall of 85.07%, Human Score 2 identifies the most correct classes out of all human evaluator saw in this evaluation. Most false errors are actually made for Human Scores 1 and 3, although their recall is still high. The F1 score is a measure that considers both precision and recall. Human Score 3 has the highest F1 score (78.62%) because the system manages to find a good balance between both positive and negative errors. The lowest F1 score was for Human Score 1 with 75.77%, meaning that while the system performs OK, its recall (finding all correct answers) is not high enough when matched against precision (correctly finding and not missing any). The outcomes reveal that the hybrid method is efficient, but even so, it might be improved to gain better results, especially when focused on Human Score 1.

It seems that the results from the system match the results from humans almost exactly. According to the results, the model can replicate what human analysts do, based on its accuracy, precision, recall and F1 scores across the evaluators. This system performs well, so it could improve on the subjective and varying decisions that people often make in assessment. Moreover, the similarity of the scores on all metrics proves that the system is reliable and works well for different types of questions and their levels of difficulty. It proves the performance of the ways features were chosen and models were created in developing the system. Furthermore, the system helps process a big number of survey answers easily and efficiently, as most of the work is handled by the computer without much effort from employees. To sum up, the findings from the evaluation validate the performance of the scoring system and support using it where scoring that is quick and based on facts is critical.

5.6 Summary

This chapter looked at the automatic answer scoring system from different angles, including working with features, testing the models used and validating the system. First, the chapter explains how it obtained a variety of features from the student responses. Features covered not only grammatical organizing and word tags, but also meaning, similarity among

groups of sentences and matching concepts to the reference answers. Terms were identified and understood by applying advanced methods in natural language processing.

After that, the models were tested to measure their performance. Some of the algorithms I tried were SVM, Random Forest and Logistic Regression, as well as LSTM networks and Transformer-based models. Also, analyzing the results of combining machine learning and deep learning for grading responses was explored to find out how effective this method can be. Once the differences between approaches were compared, the accuracy, precision, recall and F1-score for various cases were easier to understand. At the end, several evaluation tests were carried out to make sure the scoring system can be used broadly. A number of datasets were used to check if the models kept performing consistently, thanks to cross-validation. The findings demonstrated how the system could be used efficiently in actual classrooms. To sum up, the chapter introduced the basics of automated answer scoring systems and suggested what future progress may look like.

Chapter – VI

Conclusion, Recommendations and Future Work

6.1 Summary

This study gives a unique technique to computerized solution scoring through integrating syntactic and semantic evaluation of Hindi language datasets. The number one targets included:

1. Studying the to be had sources inclusive of questions, reference answers, and lexicons;
2. Extracting syntactic and semantic functions for superior computerized scoring;
3. Presenting a hybrid version that mixes deep getting to know and conventional device getting to know methods; and
4. Validating the system's overall performance the usage of accuracy, precision, recall, andF1 rating metrics.

The syntactic feature extraction used POS tagging along with dependency parsing and grammar checking methods and word length and sentence length measurements for structural analysis of answers. Semantic features used advanced techniques such as BERT sentence embedding and text coherence as well as Latent Semantic Analysis (LSA) together with polysemy measurement and n-gram analysis and ambiguous word counts to assess contextual meaning.

The study model synthesized deep learning models which included Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks together with the conventional machine learning approach through Extreme Gradient Boosting (XGBoost). The combination aimed to use deep neural network sequence learning abilities with ensemble learning system classification capabilities. A thorough design approach within this model allowed it to combine syntax-related features from sentences (part-of-speech patterns and structures) with features representing semantic meaning (contextual similarities and lexical meanings). The combination of different inputs let the system calculate automated assessment scores which aimed to reach human evaluation similarities.

Performance evaluation of the model utilized standard metrics that included accuracy, precision, recall and the F1 score. Accurate performance measurements used three external human evaluators to evaluate model effectiveness. The research utilized Hindi-language educational material with student answers as the dataset although its analysis became challenging because of the distinctive linguistic features that make Hindi texts complex to process. Special properties within Hindi language demanded precise processing steps with additional feature creation tasks to establish useful model representations. This research shows that combining modern machine learning approaches leads to human-level answer scoring capabilities for Hindi questions and student answers.

6.2 Conclusions

The hybrid AAS model showed reliable results for all answer sets which human raters assessed. The model demonstrated high precision rates alongside accurate detection of answers based on human evaluation criteria resulting in good scores for precision and F1 metrics. The solid answer evaluation results support the effectiveness of bringing deep learning methods together with traditional machine learning approaches for question scoring. Multiple assessment metrics allowed a complete analysis of the predictive model to gauge correctness alongside its ability to maintain consistency across different answer types and levels of quality.

The main strength of the model lied in its capability to handle syntactic as well as semantic features which form the foundation for understanding complex Hindi-language text. Automated systems encounter major hurdles when processing Hindi because it is both morphologically complex and syntactically adaptable. The hybrid architecture proved capable of effectively handling every challenge which arose. The identification of grammatical patterns together with word sequences and structural cues in student responses required the syntactic features consisting of Part-of-Speech (POS) tagging and n-gram analysis. The model employed these characteristics to identify properly structured from improperly structured student responses with special focus on grammatical quality and text coherence.

The model used strong embedding approaches to extract detailed semantic information as well as interpret text relationships across its content. BERT-based sentence embedding analyzed word relationships in sentences no matter what order the words followed within the sentence context. The free words order along with rich inflectional morphology of Hindi received special benefits from this system. Latent Semantic Analysis (LSA) extracted

latent relationships from the corpus to help the model understand concepts and themes which existed within the answers. The model's feature extraction technique that combined syntactic and semantic analysis enabled it to perform well with answers presented in different formalities as well as long or short formats.

Deep learning models such as RNN and LSTM when integrated with XGBoost classifier formed an effective cooperative system. The combination of RNNs and LSTMs with their strength in sequence analysis worked in harmony with XGBoost abilities for gradient boosted boundary definition. By integrating these models the system could understand sequences in data as well as generate predictions through structural features. The hybrid approach exhibited excellent performance both on training cases and unknown test data which exhibited strong practical utility in educational environments.

6.3 Strengths of the study

- **High performance**

The proposed model displays a strong performance capability as the prime advantage of the study when used for automatic answer scoring tasks. The model obtained an F1 score above 78% indicating excellent precision-recall combination thus establishing fair and consistent assessment for student responses. A high accuracy rate ranging from 82% to 84% was achieved by the model through its comparison with human evaluator judgments from three different assessment sets. The model demonstrates strong reliability in scoring based on human aptitude by producing these results which minimize both variation and human interpretation biases. The model demonstrates uniform accuracy across different data samples which increases the validity and general applicability of the proposed methodology.

- **Comprehensive feature extraction**

Feature extraction within the model functions effectively because it integrates both syntactic and semantic characteristics which appear in student answers. By counting the usage of grammar and using part-of-speech tags, the model assessed how answers are written linguistically. Semantic features such as sentence embedding together with latent topic models analyzed the fundamental meanings along with conceptual concepts in student answers. The model utilized dual-level feature integration to assess both structural aspect and content quality of answers thereby delivering evaluations which captured genuine answer

quality. The model demonstrates higher reliability when it combines all types of linguistic data since it becomes more effective at analyzing diverse student writing approaches across different content difficulties.

- **Effective hybrid model**

The integration between deep learning networks RNN and LSTM components with XGBoost gradient boosting algorithm produced exceptional results in the proposed model. The RNN and LSTM modules extracted simultaneous sequence patterns contained in the student replies because these patterns are fundamental for understanding coherent sentence organization. The structured feature processing ability and precise interpretable output capabilities of XGBoost played an important role in the system alongside RNN and LSTM components. The dual approach united deep learning contextual capabilities with machine learning decision structures into an efficient unified model. The system's design enabled it to process multiple linguistic patterns together with various answer forms thus becoming an adaptive solution for automated scoring operations.

6.4 Weaknesses of the study

- **Limited generalizability**

The model achieved top results on Hindi but showed restricted application to languages outside its domain. This model employed Hindi language-specific syntactic and semantic features to accommodate the linguistic traits of Hindi morphological patterns and flexible word orders as well as its script style. The model needs major adjustments when transferred to language structures beyond Hindi because it was designed specifically for the Hindi linguistic properties. The system requires choices of particular linguistic features while embedding models need to be retrained and preprocessing tools need to be built to function with each language selection. The predictive model shows limitations when processing complex questions that deviate from standard training inputs because such questions introduce previously untrained usage patterns. Further development of the model's utility depends on additional validation along with engineering tasks in diverse linguistic contexts across different settings.

- **Handling of ambiguous answers**

The model employed basic ambiguity resolution tactics which included polysemy measurement and ambiguous word counting but continues to struggle with interpretations of answers that heavily depend on contextual meanings. Student answers that contain elusive or subtle or idiomatic expressions require deep cultural or discourse understanding thus making their assessment by automated systems difficult. The BERT system provides contextual help to an extent but struggles to resolve deep semantic elements and complex sentence structure particularly when dealing with Hindi which is a low-resource language. The model could improve ambiguity handling by implementing sophisticated disambiguation methods through attention mechanisms and knowledge graph integration and semantic role labeling to properly understand the students' response intentions and relevance.

- **Dependency on predefined features**

The model achieves its results based on the higher quality of features acquired during preprocessing. The prediction accuracy gets significantly affected when errors occur in feature extraction because syntactic and semantic features serve as the prediction's foundational elements. The model depends on predetermined linguistic cues and representations but lacks capability to evaluate creative ideas or critical thinking aside from argument strength or other answer quality aspects. Flaws in the stages of feature engineering will cause the model to overlook vital elements of student work which results in both inaccurate and biased examination scores. The solution to resolve this deficiency requires combination of adaptive extraction processes and learnable features with stronger linguistic analytics and domain-specific knowledge integration.

6.5 Challenges of the study

- **Language-specific complexity**

The research brought forth a major obstacle through the intricate nature of using the Hindi language. Linguistic diversity in Hindi reaches high levels because the language shows different regional dialects and uses different vocabularies and creates sentences in various ways. The linguistic system of Hindi works through numerous morphological forms since one basic word transforms into multiple word forms through its tense markers together with gender markers and number markers and case markers. The many possible word forms of

Hindi morphology produce processing difficulties throughout tokenization and stemming and lemmatization phases of text preparation. Student responses across the study presented challenges because Hindi allows flexible word order together with multiple complex grammatical structures. Structural analysis of flexible Hindi sentences required the selection and adaptation of specific linguistic tools for accurate interpretation. The technology required multi-round model assessments for accurate linguistic understanding because data can be very complex.

- **Data scarcity**

The main trouble during development was getting enough well-labeled Hindi text data that was ideal for automatic scoring. The English language benefits from the ASAP (Automated Student Assessment Prize) dataset which is not available for Hindi resources because it currently suffers from scattered materials. The model training process suffered due to data scarcity because insufficiently diverse data prevented the model from familiarizing itself with different answer types together with diverse question formats and writing practices. The model demonstrated restricted abilities when it came to applying its knowledge across various educational levels and subject populations and student populations. The model training process became challenging because the small dataset prevented utilization of advanced deep learning frameworks which need vast training materials. Despite the manual data collection and preprocessing efforts together with cross-validation methods the study did not fully resolve the fundamental problem of insufficient data variety and quantity.

- **Feature integration**

Another major technical hurdle existed in the process of merging syntactic alongside semantic model components in a single framework. The integration of syntactic and semantic features demanded thorough testing because these features supply separate yet important information about student responses through grammar structure detection and content meaning description. The initial model versions displayed problems where dominant features overwhelmed the others and created unwanted disruptions which deteriorated model performance. The iterative design and testing and adjusting procedures led to identifying the suitable architecture which handled both features types without conflicts. Model complementarity involved proper feature selection alongside finding the best methods to transform linguistic indicators for the learning process. Combining different feature types within the hybrid model structure which combined RNNs, LSTMs and XGBoost created

training and interpretive complexity that demanded multidimensional expertise of feature patterns and algorithmic system behavior.

6.6 Future Scope

1. **Model Generalization:** The proposed approach needs further development to apply it to alternative language sets and question-answer databases by implementing features that match unique linguistic elements.
2. **Advanced semantic analysis:** Incorporating more advanced herbal language processing techniques, such as contextual phrase embedding or transformer models, may improve the capability of the model to deal with complicated solution patterns.
3. **Real-world deployment:** The system incorporates within educational platforms to evaluate real-time answers in schools and universities so teachers can receive assistance with large-scale grading and uniformity and equity.
4. **Enhanced ambiguity resolution:** The system's semantic accuracy can improve through additional work to disambiguate polysemy and contextually ambiguous words because this leads to better semantic accuracy especially when analyzing subject matter requiring complex semantic interpretation.

The research moves toward advanced automated assessment systems which merge syntax with semantics especially for the underdeveloped language of Hindi. Automated scoring systems have proven in this model their ability to integrate into educational technology platforms that deliver continuous accurate assessment at a large scale.

REFERENCES

1. Brackett, M. A., Floman, J. L., Ashton-James, C., Cherkasskiy, L., & Salovey, P. (2013). The influence of teacher emotion on grading practices: A preliminary look at the evaluation of student writing. *Teachers and teaching*, *19*(6), 634-646.

2. Brew, C., & Leacock, C. (2013). Automated short answer scoring: Principles and prospects. In *Handbook of Automated Essay Evaluation* (pp. 136-152). Routledge.

3. Hahn, M. G., Navarro, S. M. B., Valentín, L. D. L. F., & Burgos, D. (2021). A systematic review of the effects of automatic scoring and automatic feedback in educational settings. *Ieee Access*, *9*, 108190-108198.

4. Madnani, N., & Cahill, A. (2018, August). Automated scoring: Beyond natural language processing. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1099-1109).

5. Gomaa, W. H., & Fahmy, A. A. (2014). Automatic scoring for answers to Arabic test questions. *Computer Speech & Language*, *28*(4), 833-857.

6. Babic, B., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in industry*, *59*(4), 321-337.

7. Bruckner, D. (2007). *Probabilistic models in building automation: recognizing scenarios with statistical methods* (Doctoral dissertation, Technische Universität Wien)

8. Silva, V. D. S. (2022). *A Composite Syntactic-Semantic Interpretable Text Entailment Approach Exploring Commonsense Knowledge Graphs* (Doctoral dissertation, Universität Passau).

9. Singh, S., Pupneja, A., Mital, S., Shah, C., Bawkar, M., Gupta, L. P., ... & Shah, R. R. (2023). H-AES: towards automated essay scoring for hindi. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, No. 13, pp. 15955-15963).

10. Kumar, N. V. A., & Mehrotra, S. (2022, December). A comparative analysis of word embedding techniques and text similarity measures. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)* (pp. 1581-1585). IEEE.

11. Aini, Q., Julianto, A. E., & Purbohadi, D. (2018). Development of a scoring Application for Indonesian language essay questions. In *Proceedings of the 2018 2nd International Conference on Education and E-Learning* (pp. 6-10).

12. Lubis, F. F., Putri, A., Waskita, D., Sulistyaningtyas, T., Arman, A. A., & Rosmansyah, Y. (2021). Automated short-answer grading using semantic similarity based on word embedding. *International Journal of Technology*, *12*(3), 571-581.

13. Mohler, M., Bunescu, R., & Mihalcea, R. (2011, June). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 752-762).

14. Çınar, A., Ince, E., Gezer, M., & Yılmaz, Ö. (2020). Machine learning algorithm for grading open-ended physics questions in Turkish. *Education and information technologies*, *25*(5), 3821-3844.

15. Olowolayemo, A., Nawi, S. D., & Mantoro, T. (2018, September). Short answer scoring in English grammar using text similarity measurement. In *2018 international conference on computing, engineering, and design (ICCED)* (pp. 131-136). IEEE.

16. Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M. (2020). Automatic short answer grading and feedback using text mining methods. *Procedia computer science*, *169*, 726-743.

17. Bonthu, S., Rama Sree, S., & Krishna Prasad, M. H. M. (2021). Automated short answer grading using deep learning: A survey. In *Machine Learning and Knowledge Extraction: 5th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2021, Virtual Event, August 17–20, 2021, Proceedings 5* (pp. 61-78). Springer International Publishing.

18. Anekboon, K. (2018, October). Automated scoring for short answering subjective test in Thai's language. In *2018 International Conference on Image and Video Processing, and Artificial Intelligence* (Vol. 10836, pp. 324-329). SPIE.

19. Kudi, P., Manekar, A., Daware, K., & Dhatrak, T. (2014, December). Online Examination with short text matching. In *2014 IEEE Global Conference on Wireless Computing & Networking (GCWCN)* (pp. 56-60). IEEE.

20. Condor, A. (2020). Exploring automatic short answer grading as a tool to assist in human rating. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21* (pp. 74-79). Springer International Publishing.

21. Roy, S., Narahari, Y., & Deshmukh, O. D. (2015). A perspective on computer assisted assessment techniques for short free-text answers. In *Computer Assisted Assessment. Research into E-Assessment: 18th International Conference, CAA 2015, Zeist, The Netherlands, June 22–23, 2015. Proceedings 18* (pp. 96-109). Springer International Publishing.

22. Ye, X., & Manoharan, S. (2018, September). Machine learning techniques to automate scoring of constructed-response type assessments. In *2018 28th EAEEIE annual conference (EAEEIE)* (pp. 1-6). IEEE.

23. Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019, July). Get it scored using autosas: an automated system for scoring short answers. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 9662-9669).

24. Peñalvo, F. J. G., & Ingelmo, A. V. (2023). What do we mean by GenAI? A systematic mapping of the evolution, trends, and techniques involved in Generative AI. *IJIMAI*, *8*(4), 7-16.

25. Shweta, P., & Adhiya, K. (2022, June). Comparative study of feature engineering for automated short answer grading. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)* (pp. 594-597). IEEE.

26. Tulu, C. N., Ozkaya, O., & Orhan, U. (2021). Automatic short answer grading with semspace sense vectors and malstm. *IEEE Access*, *9*, 19270-19280.

27. Sato, T., Funayama, H., Hanawa, K., & Inui, K. (2022, July). Plausibility and faithfulness of feature attribution-based explanations in automated short answer scoring. In *International Conference on Artificial Intelligence in Education* (pp. 231-242). Cham: Springer International Publishing.

28. Taghipour, K., & Ng, H. T. (2016, November). A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1882-1891).

29. Tay, Y., Phan, M., Tuan, L. A., & Hui, S. C. (2018, April). Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).

30. Song, W., Song, Z., Liu, L., & Fu, R. (2020, July). Hierarchical Multi-task Learning for Organization Evaluation of Argumentative Student Essays. In *IJCAI* (pp. 3875-3881).

31. Wang, Y., Wei, Z., Zhou, Y., & Huang, X. J. (2018). Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 791-797).

32. Mathias, S., & Bhattacharyya, P. (2020, July). Can neural networks automatically score essay traits?. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 85-91).

33. Dong, F., Zhang, Y., & Yang, J. (2017, August). Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)* (pp. 153-162).

34. Bhatt, B., & Bhattacharyya, P. (2011, December). IndoWordNet and its linking with ontology. In *Proceedings of the 9th International Conference on Natural Language Processing (ICON-2011)*.

35. Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.

36. Shweta, P., & Adhiya, K. (2022, June). Comparative study of feature engineering for automated short answer grading. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)* (pp. 594-597). IEEE.

37. Ndukwe, I. G., Daniel, B. K., & Amadi, C. E. (2019). A machine learning grading system using chatbots. In *Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II 20* (pp. 365-368). Springer International Publishing.

38. Qi, H., Wang, Y., Dai, J., Li, J., & Di, X. (2019, July). Attention-based hybrid model for automatic short answer scoring. In *International Conference on Simulation Tools and Techniques* (pp. 385-394). Cham: Springer International Publishing.

39. Uto, M., & Uchida, Y. (2020). Automated short-answer grading using deep neural networks and item response theory. In *Artificial Intelligence in Education: 21st International*

Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21 (pp. 334-339). Springer International Publishing.

40. De Gasperis, G., Menini, S., Tonelli, S., & Vittorini, P. (2019, September). Automated grading of short text answers: preliminary results in a course of health informatics. In *International Conference on Web-Based Learning* (pp. 190-200). Cham: Springer International Publishing.

41. Sakaguchi, K., Heilman, M., &, N. (2015). Effective feature integration for automated short answer scoring. In *Proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies* (pp. 1049-1054).

42. Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019, July). Get it scored using autosas—an automated system for scoring short answers. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 9662-9669).

43. Anekboon, K. (2018, October). Automated scoring for short answering subjective test in Thai's language. In *2018 International Conference on Image and Video Processing, and Artificial Intelligence* (Vol. 10836, pp. 324-329). SPIE.

44. Kakwani, D., Kunchukuttan, A., Golla, S., NC, G., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020, November). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4948-4961).

45. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ...& Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

46. Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., ...& Talukdar, P. (2021). Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

47. Rudner, L. M., & Liang, T. (2002). Automated essay scoring using Bayes' theorem. *The Journal of Technology, Learning and Assessment*, *1*(2).

48. Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® V. 2. *The Journal of Technology, Learning and Assessment*, *4*(3).

49. Shermis, M. D., & Burstein, J. (2013). Handbook of automated essay evaluation. *NY: Routledge*.

50. Evanini, K., & Wang, X. (2013, August). Automated speech scoring for non-native middle school students with multiple task types. In *INTERSPEECH* (pp. 2435-2439).

51. Higgins, D., & Heilman, M. (2014). Managing what we can measure: Quantifying the susceptibility of automated scoring systems to gaming behavior. *Educational Measurement: Issues and Practice*, *33*(3), 36-46.

52. Bejar, I. I., Flor, M., Futagi, Y., & Ramineni, C. (2014). On the vulnerability of automated scoring to construct-irrelevant response strategies (CIRS): An illustration. *Assessing Writing*, *22*, 48-59.

53. Burrows, S., Gurevych, I., & Stein, B. (2015). The eras and trends of automatic short answer grading. *International journal of artificial intelligence in education*, *25*, 60-117.

54. Cummins, R., Zhang, M., & Briscoe, T. (2016). Constrained multi-task learning for automated essay scoring. Association for Computational Linguistics.

55. Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.

56. Amorim, E., & Veloso, A. (2017, April). A multi-aspect analysis of automatic essay scoring for Brazilian Portuguese. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 94-102).

57. Ajetunmobi, S. A., & Daramola, O. (2017, October). Ontology-based information extraction for subject-focussed automatic essay evaluation. In *2017 International Conference on Computing Networking and Informatics (ICCNI)* (pp. 1-6). IEEE.

58. Xi, X. (2010). Automated scoring and feedback systems: Where are we and where are we heading?. *Language Testing*, *27*(3), 291-300.

59. Shadiev, R., & Feng, Y. (2024). Using automated corrective feedback tools in language learning: a review study. *Interactive learning environments*, *32*(6), 2538-2566.

60. Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, *55*(3), 2495-2527.

61. Hussein, M. A., Hassan, H., & Nassef, M. (2019). Automated language essay scoring systems: A literature review. *PeerJ Computer Science*, *5*, e208.

62. AlKhuzaey, S., Grasso, F., Payne, T. R., & Tamma, V. (2024). Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, *34*(3), 862-914.

63. Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, *55*(3), 2495-2527.

64. Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E., & Xu, G. (2017). A topic modeling based approach to novel document automatic summarization. *Expert Systems with Applications*, *84*, 12-23.

65. Kumar, V., & Boulanger, D. (2020, October). Explainable automated essay scoring: Deep learning really has pedagogical value. In *Frontiers in education* (Vol. 5, p. 572367). Frontiers Media SA.

66. Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, *37*, 389-405.

67. Nunes, A., Cordeiro, C., Limpo, T., & Castro, S. L. (2022). Effectiveness of automated writing evaluation systems in school settings: A systematic review of studies from 2000 to 2020. *Journal of Computer Assisted Learning*, *38*(2), 599-620.

68. Sravanthi, P., & Srinivasu, B. (2017). Semantic similarity between sentences. *International Research Journal of Engineering and Technology (IRJET)*, *4*(1), 156-161.

69. Ajay, H. B. (1973). Analysis of Essays by Computer (AEC-II). Final Report.

70. Darwish, S. M., & Mohamed, S. K. (2020). Automated essay evaluation based on fusion of fuzzy ontology and latent semantic analysis. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) 4* (pp. 566-575). Springer International Publishing.

71. Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *international journal of Computer Applications*, *68*(13).

72. Sahu, A., & Bhowmick, P. K. (2019). Feature engineering and ensemble-based approach for improving automatic short-answer grading performance. *IEEE Transactions on Learning Technologies*, *13*(1), 77-90.\

73. Cutrone, L., & Chang, M. (2011, July). Auto-assessor: computerized assessment system for marking student's short-answers automatically. In *2011 IEEE International Conference on Technology for Education* (pp. 81-88). IEEE.

74. Ratna, A. A. P., Artajaya, H., & Adhi, B. A. (2013, August). GLSA based online essay grading system. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)* (pp. 358-361). IEEE.

75. Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 746-751).

76. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.

77. Levy, O., & Goldberg, Y. (2014, June). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning* (pp. 171-180).

78. Ramnarain-Seetohul, V., Bassoo, V., & Rosunally, Y. (2022). Similarity measures in automated essay scoring systems: A ten-year review. *Education and Information Technologies*, *27*(4), 5573-5604.

79. Dong, F., & Zhang, Y. (2016, November). Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1072-1077).

80. Shukla, D., & Gupta, S. (2024, November). Feature Engineering Techniques to Enhance Credit Scoring Models. In *2024 6th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)* (pp. 1-5). IEEE.

81. Sahu, A., & Bhowmick, P. K. (2019). Feature engineering and ensemble-based approach for improving automatic short-answer grading performance. *IEEE Transactions on Learning Technologies*, *13*(1), 77-90.

82. Xu, W., Mahmud, R. B., & Lam, H. W. (2024). A Systematic Literature Review: Are Automated Essay Scoring Systems Competent in Real-life Education Scenarios?. *IEEE Access*.

83. Bejar, I. I. (2017). Threats to score meaning in automated scoring. In *Validation of score meaning for the next generation of assessments* (pp. 75-84). Routledge.

84. Yang, H., He, Y., Bu, X., Xu, H., & Guo, W. (2023). Automatic essay evaluation technologies in chinese writing—a systematic literature review. *Applied Sciences*, *13*(19), 10737.

85. Ramnarain-Seetohul, V., Bassoo, V., & Rosunally, Y. (2022). Similarity measures in automated essay scoring systems: A ten-year review. *Education and Information Technologies*, *27*(4), 5573-5604.

86. Cozma, M., Butnaru, A. M., & Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. *arXiv preprint arXiv:1804.07954*.

87. Deepender, & Walia, T. S. (2022, November). Investigating the Role of Semantic Analysis in Automated Answer Scoring. In *International Conference on Innovations in Computational Intelligence and Computer Vision* (pp. 559-571). Singapore: Springer Nature Singapore.

88. Contreras, J. O., Hilles, S., & Abubakar, Z. B. (2018, July). Automated essay scoring with ontology based on text mining and nltk tools. In *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)* (pp. 1-6). IEEE.

89. Nehm, R. H., Ha, M., & Mayfield, E. (2012). Transforming biology assessment with machine learning: automated scoring of written evolutionary explanations. *Journal of Science Education and Technology*, *21*, 183-196.

90. Heilman, M., & Madnani, N. (2013, June). ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 275-279).

91. Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011, June). A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (pp. 180-189).

92. Chen, M., & Zechner, K. (2011, June). Computing and evaluating syntactic complexity features for automated scoring of spontaneous non-native speech. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 722-731).

93. Zechner, K., & Bejar, I. (2006, June). Towards automatic scoring of non-native spontaneous speech. In *Proceedings of the Human Language Technology Conference of the NAACL, main conference* (pp. 216-223).

94. Chen, H., He, B., Luo, T., & Li, B. (2012). A ranked-based learning approach to automated essay scoring. In Cloud and green computing (CGC) 2012 Second International Conference (pp. 448–455). New York, NY: IEEE.

95. Clauser, B. E., Yaneva, V., Baldwin, P., An Ha, L., & Mee, J. (2024). Automated Scoring of Short-Answer Questions: A Progress Report. *Applied Measurement in Education*, *37*(3), 209-224.

96. Santos, V. D., Verspoor, M., & Nerbonne, J. (2012). Identifying important factors in essay grading using machine learning. *International experiences in language testing and assessment—Selected papers in memory of Pavlos Pavlou*, 295-309.

97. Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).

98. Salim, Y., Stevanus, V., Barlian, E., Sari, A. C., & Suhartono, D. (2019, December). Automated English digital essay grader using machine learning. In *2019 IEEE International Conference on Engineering, Technology and Education (TALE)* (pp. 1-6). IEEE.

99. Birla, N., Jain, M. K., & Panwar, A. (2022). Automated assessment of subjective assignments: A hybrid approach. *Expert Systems with Applications*, *203*, 117315.

100. Xiao, R., Guo, W., Zhang, Y., Ma, X., & Jiang, J. (2020, December). Machine learning-based automated essay scoring system for Chinese proficiency test (HSK). In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval* (pp. 18-23).

101. Liu, Z., Xiong, X., Li, Y., Yu, Y., Lu, J., Zhang, S., & Xiong, F. (2024). HyGloadAttack: Hard-label black-box textual adversarial attacks via hybrid optimization. *Neural Networks*, *178*, 106461.

102. Fonseca, E., Medeiros, I., Kamikawachi, D., & Bokan, A. (2018, August). Automatically grading brazilian student essays. In *International Conference on Computational Processing of the Portuguese Language* (pp. 170-179). Cham: Springer International Publishing.

103. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

104. Liang, D., Zheng, C., Guo, L., Cui, X., Xiong, X., Rong, H., & Dong, J. (2020, December). BERT enhanced neural machine translation and sequence tagging model for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications* (pp. 57-66).

105. Yuan, S., He, T., Huang, H., Hou, R., & Wang, M. (2020). Automated Chinese essay scoring based on deep learning. *Computers, Materials & Continua*, *65*(1).

106. Borah, M. A. R., Dev, R. S., Suprathik, B. M., Harshini, A. R., Boggula, Y., & Charitha, V. (2024, December). Automated Models in Educational Assessment: A Comprehensive Survey. In *2024 9th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1029-1034). IEEE.

107. Gao, J., Yang, Q., Zhang, Y., Zhang, L., & Wang, S. (2021, July). A bi-modal automated essay scoring system for handwritten essays. In *2021 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

108. Yang, H., He, Y., Bu, X., Xu, H., & Guo, W. (2023). Automatic essay evaluation technologies in chinese writing—a systematic literature review. *Applied Sciences*, *13*(19), 10737.

109. Patel, S., Patel, P., Dave, S., Patel, S., Bhatt, N., & Thakkar, A. (2024, April). Revolutionizing Educational Assessment: Deep Learning for Question Paper Quality Evaluation. In *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 119-129). Singapore: Springer Nature Singapore.

110. Dasgupta, T., Naskar, A., Dey, L., & Saha, R. (2018, July). Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications* (pp. 93-102).

111. Wang, Y., Wei, Z., Zhou, Y., & Huang, X. J. (2018). Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 791-797).

112. Uto, M., & Okano, M. (2020). Robust neural automated essay scoring using item response theory. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21* (pp. 549-561). Springer International Publishing.

113. Surya, K., Gayakwad, E., & Nallakaruppan, M. J. I. J. R. T. E. (2019). Deep learning for short answer scoring. *Int. J. Recent. Technol. Eng.(IJRTE)*, *7*(6).

114. Patil, P., & Agrawal, A. (2018). Auto Grader for Short Answer Questions.

115. Gong, T., & Yao, X. (2019). An attention-based deep model for automatic short answer score. *International Journal of Computer Science and Software Engineering*, *8*(6), 127-132.

116. Uto, M. (2021). A review of deep-neural automated essay scoring models. *Behaviormetrika*, *48*(2), 459-484.

117. Beseiso, M., Alzubi, O. A., & Rashaideh, H. (2021). A novel automated essay scoring approach for reliable higher educational assessments. *Journal of Computing in Higher Education*, *33*, 727-746.

118. Cao, Y., Jin, H., Wan, X., & Yu, Z. (2020, July). Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 1011-1020).

119. Aliabadi, M. M., Emami, H., Dong, M., & Huang, Y. (2020). Attention-based recurrent neural network for multistep-ahead prediction of process performance. *Computers & Chemical Engineering*, *140*, 106931.

120. Chen, M., & Li, X. (2018, November). Relevance-based automated essay scoring via hierarchical recurrent model. In *2018 International Conference on Asian Language Processing (IALP)* (pp. 378-383). IEEE.

121. Misgna, H., On, B. W., Lee, I., & Choi, G. S. (2025). A survey on deep learning-based automated essay scoring and feedback generation. *Artificial Intelligence Review*, *58*(2), 1-40.

122. Yuan, S., He, T., Huang, H., Hou, R., & Wang, M. (2020). Automated Chinese essay scoring based on deep learning. *Computers, Materials & Continua*, *65*(1).

123. Xie, B., & Chen, L. (2021, December). Automatic Scoring Model of Subjective Questions Based Text Similarity Fusion Model. In *international conference on wireless communications, networking and applications* (pp. 586-599). Singapore: Springer Nature Singapore.

124. Lee, G. G., Latif, E., Wu, X., Liu, N., & Zhai, X. (2024). Applying large language models and chain-of-thought for automatic scoring. *Computers and Education: Artificial Intelligence*, *6*, 100213.

125. Sharma, A., & Jayagopi, D. B. (2024). Modeling essay grading with pre-trained BERT features. *Applied Intelligence*, *54*(6), 4979-4993.

126. Lu, C., & Cutumisu, M. (2021). Integrating Deep Learning into an Automated Feedback Generation System for Automated Essay Scoring. *International Educational Data Mining Society*.

127. Yuan, S., He, T., Huang, H., Hou, R., & Wang, M. (2020). Automated Chinese essay scoring based on deep learning. *Computers, Materials & Continua*, *65*(1).

128. Gurunath, R., Alahmadi, A. H., Samanta, D., Khan, M. Z., & Alahmadi, A. (2021). A novel approach for linguistic steganography evaluation based on artificial neural networks. *IEEE Access*, *9*, 120869-120879.

129. Liang, G., On, B. W., Jeong, D., Kim, H. C., & Choi, G. S. (2018). Automated essay scoring: A siamese bidirectional LSTM neural network architecture. *Symmetry*, *10*(12), 682.

130. Jeon, S., & Strube, M. (2021, November). Countering the influence of essay length in neural essay scoring. In *Proceedings of the second workshop on simple and efficient natural language processing* (pp. 32-38).

131. Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, *55*(3), 2495-2527.

132. Chakravarthi, B. R., Arcan, M., & McCrae, J. P. (2018, January). Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th global wordnet conference* (pp. 77-86).

133. Arora, G. (2020). inltk: Natural language toolkit for indic languages. *arXiv preprint arXiv:2009.12534*.

134. Kakwani, D., Kunchukuttan, A., Golla, S., NC, G., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020, November). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 4948-4961).

135. Ramesh, G., Doddapaneni, S., Bheemaraj, A., Jobanputra, M., Ak, R., Sharma, A., ...& Khapra, M. S. (2022). Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, *10*, 145-162.

136. Desai, N. P., & Dabhi, V. K. (2021). Taxonomic survey of Hindi Language NLP systems. *arXiv preprint arXiv:2102.00214*.

137. Harish, B. S., & Rangan, R. K. (2020). A comprehensive survey on Indian regional language processing. *SN Applied Sciences*, *2*(7), 1204.

138. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

139. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ...& Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

140. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

141. Agarwal, D., Gupta, S., & Baghel, N. (2020, December). ScAA: A dataset for automated short answer grading of children's free-text answers in Hindi and Marathi. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)* (pp. 430-436).

142. Sun, J., Song, T., Peng, W., & Song, J. (2025). A survey of automated essay scoring: Challenges, advances, and future. *Neurocomputing*, *650*, 130916.

143. Sanuvala, G., Fatima, S. S., Kambhampati, T., & Sanuvala, R. (2023, October). Automatic short answer scoring on an Indian dataset using transformer-based language models. In *International Conference on Computer & Communication Technologies* (pp. 287-295). Singapore: Springer Nature Singapore.

144. Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

145. Fleckenstein, J., Meyer, J., Jansen, T., Keller, S., & Köller, O. (2020). Is a long essay always a good essay? The effect of text length on writing assessment. *Frontiers in psychology*, *11*, 562462.

146. Singh, J., Joshi, N., & Mathur, I. (2013, August). Development of Marathi part of speech tagger using statistical approach. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1554-1559). IEEE.

147. Kumar, D., & Josan, G. S. (2010). Part of speech taggers for morphologically rich indian languages: a survey. *International Journal of Computer Applications*, *6*(5), 32-41.

148. Kumawat, D., & Jain, V. (2015). POS tagging approaches: A comparison. *International Journal of Computer Applications*, *118*(6).

149. Zhou, H., Zhang, Y., Li, Z., & Zhang, M. (2020, October). Is POS tagging necessary or even helpful for neural dependency parsing?. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 179-191). Cham: Springer International Publishing.

150. Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Cinková, S., Flickinger, D., ...& Urešová, Z. (2015, June). Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 915-926).

151. Nivre, J., Basirat, A., Dürlich, L., & Moss, A. (2022). Nucleus composition in transition-based dependency parsing. *Computational Linguistics*, *48*(4), 849-886.

152. Tratz, S., & Hovy, E. (2011, July). A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 1257-1268).

153. Nivre, J. (2010). Dependency parsing. *Language and Linguistics Compass*, *4*(3), 138-152.

154. Habib, M., Faris, M., Alomari, A., & Faris, H. (2021). Altibbivec: a word embedding model for medical and health applications in the Arabic language. *IEEE Access*, *9*, 133875-133888.

155. Yu, T., Hidey, C., Rambow, O., & McKeown, K. (2017). Leveraging sparse and dense feature combinations for sentiment classification. *arXiv preprint arXiv:1708.03940*.

156. Wang, Z., Liu, J., & Dong, R. (2018, November). Intelligent auto-grading system. In *2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS)* (pp. 430-435). IEEE.

157. Chen, Z., & Zhou, Y. (2019, May). Research on automatic essay scoring of composition based on CNN and OR. In *2019 2nd international conference on artificial intelligence and big data (ICAIBD)* (pp. 13-18). IEEE.

158. Hasanah, U., Astuti, T., Wahyudi, R., Rifai, Z., & Pambudi, R. A. (2018, November). An experimental study of text preprocessing techniques for automatic short answer grading in Indonesian. In *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)* (pp. 230-234). IEEE.

159. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

160. Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

161. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

162. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

163. Jin, C., He, B., Hui, K., & Sun, L. (2018, July). TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1088-1097).

164. Beseiso, M., & Alzahrani, S. (2020). An empirical analysis of BERT embedding for automated essay scoring. *International Journal of Advanced Computer Science and Applications*, *11*(10).

165. Fromkin, V., Rodman, R., Hyams, N. M., Amberber, M., Cox, F., & Thornton, R. (2017). *An Introduction to Language with Online Study Tools 12 Months*. Cengage AU.

166. James, H. R., & Heasley, B. (1983). Semantics: a coursebook. *Great Britain: Cambridge University*.

167. Oaks, D. (2010). Understanding Ambiguity in English. Palgrave Macmillan.

168. Evangelopoulos, N. E. (2013). Latent semantic analysis. *Wiley Interdisciplinary Reviews: Cognitive Science*, *4*(6), 683-692.

169. Singh, s. A. Topic-stemming technology.

170. Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6), 1930-1938.

171. Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, *9*.

172. Chandaka, S., Chatterjee, A., & Munshi, S. (2009). Cross-correlation aided support vector machine classifier for classification of EEG signals. *Expert systems with applications*, *36*(2), 1329-1336.

173. Awad, M., & Khanna, R. (2015). *Efficient learning machines: theories, concepts, and applications for engineers and system designers* (p. 268). Springer nature.

174. Kou, L., Sysyn, M., Liu, J., Fischer, S., Nabochenko, O., & He, W. (2023). Prediction system of rolling contact fatigue on crossing nose based on support vector regression. *Measurement*, *210*, 112579.

175. Breiman L (2001) Random Forests Mach Learn 45:5–32

176. Palczewska, A., Palczewski, J., Marchese Robinson, R., & Neagu, D. (2014). Interpreting random forest classification models using a feature contribution method. *Integration of reusable systems*, 193-218.

177. Demir, S., & Şahin, E. K. (2022). Liquefaction prediction with robust machine learning algorithms (SVM, RF, and XGBoost) supported by genetic algorithm-based feature selection and parameter optimization from the perspective of data processing. *Environmental Earth Sciences*, *81*(18), 459.

178. Quinto, B. (2020). *Next-generation machine learning with spark: Covers XGBoost, LightGBM, Spark NLP, distributed deep learning with keras, and more*. Apress.

179. Demir, S., & Sahin, E. K. (2023). An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost. *Neural Computing and Applications*, *35*(4), 3173-3190.

180. Limna, P., Jakwatanatham, S., Siripipattanakul, S., Kaewpuang, P., & Sriboonruang, P. (2022). A review of artificial intelligence (AI) in education during the digital era. *Advance Knowledge for Executives*, *1*(1), 1-9.

181. Del Gobbo, E., Guarino, A., Cafarelli, B., Grilli, L., & Limone, P. (2023). Automatic evaluation of open-ended questions for online learning. A systematic mapping. *Studies in Educational Evaluation*, *77*, 101258.

182. Dong, F., Zhang, Y., & Yang, J. (2017, August). Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)* (pp. 153-162).

183. O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

184. Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388-427.

185. Hibat-Allah, M., Ganahl, M., Hayward, L. E., Melko, R. G., & Carrasquilla, J. (2020). Recurrent neural network wave functions. *Physical Review Research*, *2*(2), 023358.

186. Kou, X., Yang, Z., & Wang, Y. (2023). Research on English teaching reading quality assessment based on cognitive diagnostic assessment. *International Journal of Continuing Engineering Education and Life Long Learning*, *33*(4-5), 388-402.

187. Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

188. Shang, K., Chen, Z., Liu, Z., Song, L., Zheng, W., Yang, B., ...& Yin, L. (2021). Haze prediction model using deep recurrent neural network. *Atmosphere*, *12*(12), 1625.

189. Khan, M. A. (2021). HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes*, *9*(5), 834.

190. Ramesh, D., & Sanampudi, S. K. (2022, April). An Improved Approach for Automated Essay Scoring with LSTM and Word Embedding. In *Evolution in Computational Intelligence: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)* (pp. 35-41). Singapore: Springer Nature Singapore.

191. Li, Z., Huang, J., Zhou, Z., Zhang, H., Chang, S., & Huang, Z. (2016, June). LSTM-based deep learning models for answer ranking. In *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)* (pp. 90-97). IEEE.

192. Shah, S. R. B., Chadha, G. S., Schwung, A., & Ding, S. X. (2021). A sequence-to-sequence approach for remaining useful lifetime estimation using attention-augmented bidirectional LSTM. *Intelligent Systems with Applications*, *10*, 200049.

193. Azhari, A., Santoso, A., Ratna, A. A. P., & Prestiliano, J. (2024). Optimization of AES using BERT and BiLSTM for Grading the Online Exams. *International Journal of Intelligent Engineering & Systems*, *17*(5).

PUBLICATIONS

1. Deepender & Walia, T. S. (2025). Optimizing Automated Scoring for Hindi Responses: A Hybrid Framework with Advanced Features Integration. International Journal of High Speed Electronics and Systems, (World Scientific) ISSN: 0129-1564. DOI: 10.1142/S0129156425408071 (Scopus)

2. Deepender & Walia, T. S. (2025). A Holistic Framework for Automated Answer Scoring: Unifying Syntactic and Semantic Analysis. International Journal of Basic and Applied Sciences, 14(2), ISSN: 2227-5053. DOI: 10.14419/f18ev204 (Scopus)

3. Deepender & Walia, T. S. (2024). Hybrid Approach for Automated Answer Scoring Using Semantic Analysis in Long Hindi Text. *Revue d'Intelligence Artificielle*, *38*(1). DOI: 10.18280/ria.380122 (Scopus)

4. Deepender, & Walia, T. S. (2022, November). Investigating the Role of Semantic Analysis in Automated Answer Scoring. In *International Conference on Innovations in Computational Intelligence and Computer Vision* (pp. 559-571). Singapore: Springer Nature Singapore. DOI: 10.1201/9781003405573-58 (Scopus)

5. Deepender & Walia, T. S. (2023). Investigating the scope of semantic analysis in natural language processing considering accuracy and performance. In *Recent Advances in Computing Sciences* (pp. 323-328). CRC Press. (Taylor & Francis Group) DOI: 10.1007/978-981-99-2602-2_42

CONFERENCES

Presented a research paper entitled "Investigating the Role of Semantic Analysis in Automated Answer Scoring at 3rd International Conference on Innovations in Computational Intelligence & Computer Vision (ICICV-2022) organized by Department of Computer & Communication Engineering, Manipal University Jaipur, Rajasthan, India during November 24-25, 2022.

MANIPAL UNIVERSITY JAIPUR

NAAC A+ GRADE WITH 3.28 SCORE

ICICV

Springer

No. ICICV-2022/171_1

3rd International Conference on

Innovations in Computational Intelligence and Computer Vision (ICICV-2022)

November 24-25, 2022

CERTIFICATE

This is to certify that ***Deepender Singh*** from ***School of Computer Applications, Lovely Professional University,Punjab, India*** has presented a research paper entitled, ***"Investigating the Role of Semantic Analysis in Automated Answer Scoring"*** authored by ***Deepender Singh and Tarandeep Singh Walia*** at the ***3rd International Conference on Innovations in Computational Intelligence and Computer Vision (ICICV-2022)*** organized by Department of Computer and Communication Engineering, Manipal University Jaipur, Rajasthan, India during November 24-25, 2022.

Dr. Deepak Sinwar
(Convener)

Dr. Satyabrata Roy
(Convener)

Prof. Vijaypal Singh Dhaka
(Program Chair)

Presented a research paper entitled "Investigating the Scope of Semantic Analysis in Natural Language Processing Considering Accuracy and Performance" in the 1st International Conference on Recent Advances in Computing Advances (RACS-2022) organized by School of Computer Applications, Lovely Professional University, Punjab, India during November 04-05, 2022.

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

RACS

SCHOOL OF COMPUTER APPLICATION

Certificate No. 256888

Certificate of Paper Presentation

This is to certify that **Dr./Mr./Ms. Deepender Singh** of **Lovely Professional University, Punjab** has presented a paper entitled "**Investigating the Scope of Semantic Analysis in Natural Language Processing Considering Accuracy and Performance**" in the **1st International Conference on Recent Advances in Computing Sciences (RACS-2022)** held from 4th to 5th November 2022, organized by School of Computer Application at Lovely Professional University, Punjab.

Date of Issue : 30-11-2022
Place : Phagwara (Punjab), India

Prepared by
(Administrative Officer-Records)

Convener
(RACS-2022)

Conference Chair
(RACS-2022)

Session Chair
(RACS-2022)