Journal of Computer Science

Research Article

Hybrid Deep Learning Model for Evaluating Subjective Answers Based on Semantic Textual Similarity

Siddhesh Kudtarkar and Kavita Shirsat

Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India

Article history
Received: 16-02-2025
Revised: 04-08-2025
Accepted: 01-09-2025

Corresponding Author:
Siddhesh Kudtarkar
Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, India
Email: siddheshk599@gmail.com

Abstract: Subjective answer evaluation requires accurately identifying semantic similarities between student and reference responses. This study introduces a Hybrid Deep Learning Model (HDLM) that integrates CNN, GRU, and LSTM architectures to assess Semantic Textual Similarity (STS) more effectively. The HDLM employs two parallel branches-CNN-GRU and CNN-LSTM to capture both local syntactic features and long-range contextual dependencies, followed by Manhattan distance for semantic similarity computation. To address class imbalance and data sparsity, data augmentation and SMOTE resampling techniques are applied. The model is trained using the Quora Question Pairs dataset because of its substantial size and extensive semantic diversity. Comprehensive evaluation demonstrates HDLM's superior performance (accuracy: 87.80%, F1-score: 0.88, AUC: 0.88) compared to existing models like Siamese LSTM, Multi-head Attention, and SOTA models such as SBERT and Sentence-T5. Statistical significance was validated using Wilcoxon signed-rank tests and 95% confidence intervals. Failure cases and case studies further highlight HDLM's strengths and shortcomings. Overall, HDLM provides a robust, interpretable, and computationally efficient framework for automated subjective assessment.

Keywords: GRU, HDLM, LSTM, Subjective Answer Evaluation, Semantic Textual Similarity

Introduction

Evaluating subjective answers remains a critical challenge in educational assessment. Unlike objective questions, subjective responses vary widely in vocabulary, sentence structure, and depth of understanding, making consistent and unbiased grading difficult. Manual evaluation is time-consuming and prone to inconsistency, particularly in large-scale or online learning environments. Consequently, there is a growing interest in automating this process using techniques from deep learning and Natural Language Processing (NLP).

One core task in automating subjective answer evaluation is measuring semantic textual similarity (STS) between a reference (ideal) answer and a student's response. STS quantifies how closely two pieces of text align in meaning, regardless of their syntactic form. Accurate STS is essential for grading student answers, identifying paraphrases, and enabling scalable assessment systems.

Existing machine learning and NLP approaches for STS rely heavily on feature engineering or standalone neural architectures. However, traditional models often fail to capture both local linguistic patterns and global contextual relationships, which are crucial for understanding nuanced student responses. Moreover, some rely on manually crafted features, which may not generalize well to varied question domains.

This paper introduces a Hybrid Deep Learning Model (HDLM) designed to tackle these issues by integrating the advantages of Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs) (Cho et al., 2014), and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). CNNs are effective at capturing local n-gram patterns and translation-invariant features, which are essential for detecting key phrases and structures in student responses. GRUs and LSTMs, on the other hand, are designed to learn temporal dependencies and long-range semantic relationships across sequences of text. By integrating these architectures, the HDLM is designed to extract shallow and deep semantic features

simultaneously, improving the robustness of the evaluation of textual similarity.

In contrast to contemporary methods like transformer-based models (e.g., BERT, Sentence-BERT), which demand considerable computational resources and frequently lack interpretability, the suggested HDLM attains a balance among performance, efficiency, and interpretability. This is particularly advantageous in educational settings where scalable, fast, and explainable models are needed. While several transformer-based models exhibit excellent accuracy in semantic similarity tests, they frequently operate as black-box systems, complicating the explanation of predictions, which is a crucial element in educational environments.

The Quora Question Pairs (QQP) (He et al., 2023) dataset is used to illustrate the model's efficacy. Although QQP was initially developed to detect duplicate questions, it provides a large-scale corpus of semantically related and unrelated text pairs across diverse domains. Its high volume, lexical variation, and diverse question structures make it a suitable surrogate for educational datasets in training semantic similarity models. Compared to smaller educational datasets like ASAP, QQP allows the model to generalize across varied input patterns, improving its applicability to the evaluation of subjective answers in real-world.

Research Question and Hypothesis

- Research Question: Can a hybrid deep learning model that integrates CNN, GRU, and LSTM effectively evaluate semantic textual similarity in subjective answers and outperform existing approaches in terms of accuracy, computational efficiency, and interpretability?
- Hypothesis: A hybrid deep learning model (HDLM) that combines CNN, GRU, and LSTM will demonstrate improved semantic textual similarity assessment of subjective answers, achieving higher accuracy and interpretability than existing standalone or transformer-based models when trained on a large-scale, semantically rich dataset like Quora Question Pairs

Key Contributions

1. Propose a Hybrid Deep Learning Model (HDLM) that combines CNN, GRU, and LSTM to evaluate the semantic similarity between student and reference answers
2. Justify the architectural choices by leveraging CNN's ability to detect local patterns and GRU/LSTM's capacity to understand sequential dependencies, offering a comprehensive feature extraction pipeline
3. Demonstrate the model's effectiveness by training and evaluating it on the Quora Question Pairs dataset, surpassing a number of baseline and state-of-the-art models in terms of accuracy and efficiency

The aforementioned research question and hypothesis serve as the study's compass, with the objective of evaluating whether combining CNN, GRU, and LSTM architectures can achieve improved semantic similarity detection in subjective answer evaluation. The contributions outlined directly stem from this hypothesis, forming the basis for experimental validation, performance benchmarking, and interpretability assessment throughout the paper.

Related Work

The task of evaluating semantic textual similarity (STS) has evolved significantly, from rule-based models to sophisticated deep learning and transformer architectures. In this section, we organize the related literature into four thematic categories to clearly trace this evolution. We also highlight the scalability, domain adaptation, and interpretability gaps that persist across these methods. A structured literature review table and a final summary are provided to show how our proposed work addresses these gaps.

Literature Review Summary Table

To provide a clear comparison of existing work in Semantic Textual Similarity (STS), we summarize the key characteristics of representative models in Table 1. The table highlights the core methodology, strengths, limitations, and quantitative results of each study. This condensed view facilitates understanding of how prior approaches balance trade-offs between performance, scalability, interpretability, and critical factors in educational NLP applications.

Table 1 shows that although transformer-based models like Sentence-BERT and Sentence-T5 perform well semantically, they frequently lack interpretability and demand a large amount of processing power. On the other hand, classical and hybrid deep learning models offer greater transparency and are more deployable but may underperform in complex linguistic contexts. These findings motivate our proposed Hybrid Deep Learning Model (HDLM), which aims to bridge the performance-efficiency gap by combining CNN, GRU, and LSTM components in a modular and interpretable framework.

Traditional and Machine Learning-Based Methods

Early STS approaches relied on lexical and syntactic similarity, leveraging techniques like cosine similarity, TF-IDF vectors, n-grams, and Jaccard index. While computationally inexpensive, these methods lacked the semantic depth to understand paraphrased or contextually similar answers (Islam and Inkpen, 2008).

Table 1: Literature Review of STS Methods

Study	Method	Strength	Gap / Weakness	Result
Meenakshi and Shanavas (2022)	Shared Input LSTM	Good on short educational responses	Low generalization; poor interpretability	86.20% accuracy
Ni et al. (2021)	Sentence-T5 (T5 encoder-decoder)	Best-in-class STS performance	Very large; hard to scale in practice	0.89 Spearman's ρ
Imtiaz et al. (2020)	Siamese MaLSTM	Captures semantic similarity well	Struggles with phrase variations	81.77% accuracy
Zhang and Chen (2019)	CNN + RNN + Attention	Improves alignment using attention	Complex; lacks educational validation	86.83% accuracy
Reimers and Gurevych (2019)	Sentence-BERT (Siamese BERT)	High accuracy; powerful embeddings	Black-box nature; resource-heavy	~0.89 Spearman's ρ
Mueller and Thyagarajan (2016)	Siamese LSTM with Manhattan distance	Effective for pairwise semantic similarity	Ignores global context and deep syntactic structure	84.2% accuracy; 0.8345 Spearman's ρ

Standalone Deep Learning Models

With the introduction of RNNs, LSTMs, and CNNs, deep learning-based approaches became common. For example, the Siamese MaLSTM architecture proposed by Mueller and Thyagarajan (2016) helped capture sentence level semantics but struggled with local syntactic variations. These models were limited in their ability to generalize across diverse linguistic structures due to their reliance on sequential processing alone.

Hybrid and Ensemble Neural Architectures

To address the limitations of individual models, researchers proposed hybrid frameworks combining CNNs to extract local features and RNNs (GRUs or LSTMs) to capture global dependencies. Meenakshi and Shanavas (2022) introduced the SI-LSTM model that focuses on educational STS tasks. Zhang and Chen (2019) applied multi-head attention in a CNN+RNN hybrid to enhance duplicate question detection. Although these models improved accuracy, they often introduced significant architectural complexity and lacked domain adaptability.

Transformer-Based Models

Transformer architectures such as BERT (Devlin et al., 2019), Sentence-BERT (Reimers and Gurevych, 2019), and Sentence-T5 (Ni et al., 2021) have become the de facto state-of-the-art in STS tasks. These models use self-attention mechanisms to generate contextual sentence embeddings and achieve high accuracy on benchmark datasets. However, their reliance on extensive pretraining and fine-tuning, coupled with high computational demands and limited transparency, makes them less viable for deployment in low-resource or interpretability-focused educational environments.

Identified Gaps in Literature

Across all reviewed approaches, the following gaps remain consistent:

- Scalability: Transformer models are compute-intensive and unsuitable for low-resource educational settings
- Domain Adaptation: Most models are not evaluated or fine-tuned on educational STS tasks
- Interpretability: Deep and transformer models act as black boxes, making it difficult to trace model decisions

Dataset and Preprocessing

The dataset used in this study, the preprocessing procedures followed, the methods used to address class imbalance, and the augmentation strategies employed to improve the training data are all described in this section. To further support the dataset's appropriateness for the Semantic Textual Similarity (STS) task, we offer an analysis of its distribution and complexity.

Dataset Description

The proposed Hybrid Deep Learning Model (HDLM) is trained and evaluated using the Quora Question Pairs (QQP) dataset. The QQP dataset contains approximately 404,290 question pairs, each labeled to indicate whether the pair is semantically equivalent (duplicate) or not. Each instance includes two user-generated questions and a binary label: 1 for duplicate questions and 0 for non-duplicate questions.

Although the QQP dataset was initially developed for duplicate question detection, its structure closely aligns with the requirements of Semantic Textual Similarity (STS) tasks. Unlike smaller educational datasets like

ASAP (which contains around 17,000 samples across limited prompts), QQP offers:

- High volume: More than 400k instances enable deeper model training and better generalization
- Diverse domain coverage: Questions span varied topics, closely simulating diverse student responses
- Natural paraphrasing: The dataset includes a wide range of linguistic variations and sentence structures, similar to what is observed in subjective answers

Example Question Pairs

- Duplicate: How can I be a good geologist? vs. What should I do to be a great geologist?
- Non-duplicate: How much is 30 kV in HP? vs. Where can I find a conversion chart for CC to horsepower?

These examples reflect semantic subtleties that HDLM must learn to handle. Thus, QQP is a robust surrogate for subjective answer evaluation in STS contexts.

Data Augmentation

To mitigate overfitting and address data sparsity issues in semantic modeling, we apply data augmentation using the TextAttack (Morris et al., 2020) library. The augmentation techniques employed include:

- WordNet synonym replacement
- Random character swaps and deletions

These techniques introduce controlled variation into the dataset, helping the model generalize to new phrasings. As a result, the dataset size increased by ∼20%, and the model demonstrated a 3% improvement in validation accuracy.

Preprocessing

Preprocessing steps ensure uniformity and remove noise from raw textual data. The following transformations were applied:

- Lowercasing and whitespace normalization.
- Contraction expansion (e.g., don't → do not)
- Removal of punctuation and HTML tags
- Tokenization using NLTK's (Loper and Bird, 2002) word tokenizer
- Stopword removal using NLTK's standard English list
- Lemmatization using Word Net Lemmatizer

This preprocessing pipeline ensures that word embeddings are learned from clean, standardized token sequences.

Tokenization and Vectorization

Textual data was tokenized into individual word tokens and mapped to integer sequences using Keras's Tokenizer. These sequences were then converted into dense vectors using Google News Word2Vec embeddings (300-dimensional) (Google, 2016).

To maintain uniformity across input samples, each token sequence was padded or truncated to 300 tokens. These embeddings serve as input to both CNN and RNN modules in HDLM, capturing both semantic and syntactic contexts

Handling Class Imbalance (Resampling)

The QQP dataset exhibits class imbalance:

- Duplicate (label = 1): ∼37%
- Non-duplicate (label = 0): ∼63%

This imbalance can reduce the model's ability to accurately calculate semantic similarity by biasing it toward the majority class. To mitigate this, we adopted a hybrid resampling strategy:

- SMOTE: Synthetic generation of new duplicate (minority) samples using the SMOTE algorithm (Chawla et al., 2002)
- Random Undersampling: Reduction of majority class instances to match the minority class count using the imbalanced-learn library (Lemaître et al., 2017)

To demonstrate the impact of resampling on model performance, we compare pre- and post-resampling metrics for the HDLM in Table 2. This comparison includes class ratio, validation accuracy, and F1 score for the duplicate class.

As demonstrated in Table 2, resampling considerably raised the weighted F1 score and enhanced validation accuracy by 3.05%, indicating improved balance and decreased bias. This supports the necessity of balancing class representation during training for accurate semantic similarity detection, as shown in Figure 1.

Table 2: Effect of Resampling on Training Dataset

Metric	Pre-Resampling	Post-Resampling
Duplicate class ratio	36.9%	50.0%
Non-duplicate class ratio	63.9%	50.0%
Validation Accuracy (HDLM)	84.75%	87.80%
F1 Score (weighted)	0.80	0.88

The visuals as shown in Figure 1 confirm that the resampling process achieves near-perfect balance, essential for unbiased binary classification.

Dataset Complexity Analysis

We performed additional analysis to assess dataset complexity using average question length (in tokens) as a proxy for difficulty level as shown in Figure 2. This helps simulate the variability seen in student answers.

The plot as shown in Figure 2 shows a wide distribution of question lengths from short (5 tokens) to complex (30+ tokens). Such variation is critical for robust model training.

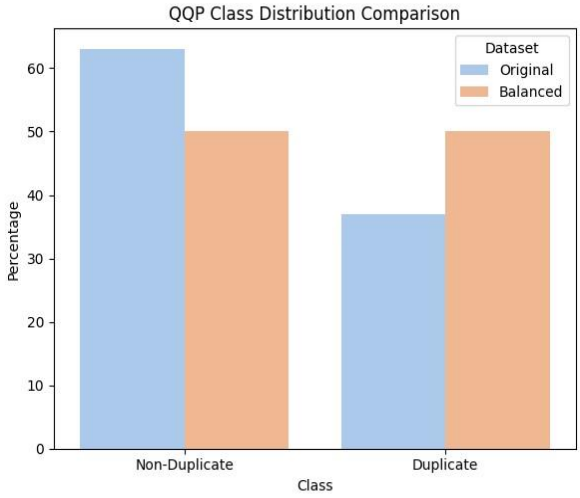

Fig. 1: QQP Class Distribution Comparison

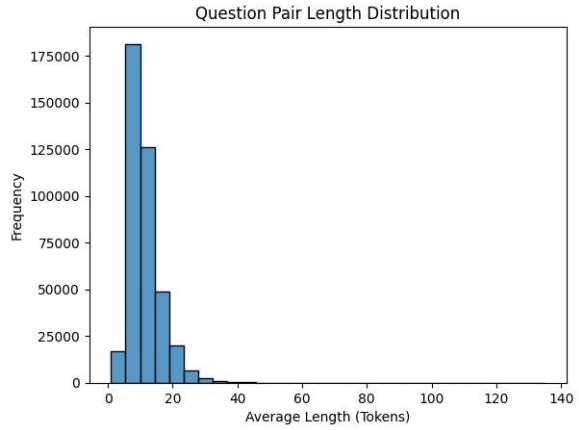

Fig. 2: QQP Question Pair Length Distribution

Summary

The QQP dataset was chosen for its scalability, diversity, and semantic richness, providing a strong foundation for modeling subjective answer evaluation. Data preprocessing, augmentation, and class balancing techniques collectively enhance HDLM's ability to generalize across varied question structures and complexity levels. The resampling process specifically improved classification accuracy and minority class recall, demonstrating its importance in STS tasks using imbalanced datasets. A summary of the preprocessing procedures used on the Quora dataset prior to its input into the HDLM can be found in Figure 3.

As illustrated in Figure 3, this sequential pipeline ensures clean, consistent, and semantically enriched input data.

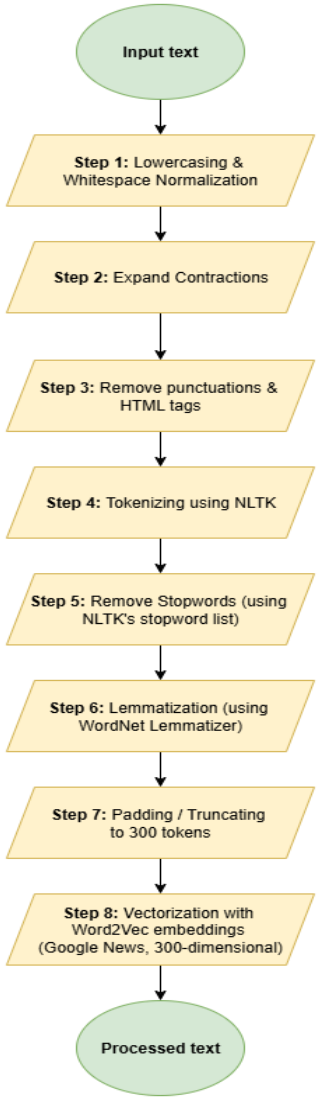

Fig. 3: Dataset Preprocessing process

Methods

The proposed model aims to accurately evaluate Semantic Textual Similarity (STS) in subjective answers by identifying both local and global patterns in text data.

To achieve this, we design a Hybrid Deep Learning Model (HDLM) that integrates three deep learning components: Convolutional Neural Networks (CNNs) for local feature extraction, and Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) networks for sequential pattern learning. While similar component combinations have appeared in prior studies such as Zhang and Chen (2019), our model differentiates itself by fusing parallel CNN-GRU and CNN-LSTM streams and leveraging their semantic representations in a novel dual-pathway configuration optimized for answer evaluation.

Additionally, our model adopts an interpretable similarity scoring approach using Manhattan distance, avoiding the black-box nature of attention-based systems like BERT. Compared to Sentence-T5 (Ni et al., 2021) and SBERT (Reimers and Gurevych, 2019), our HDLM achieves strong accuracy while being significantly lighter in terms of computational complexity, making it more suitable for real-time educational settings.

Overview of HDLM

Previous studies such as Meenakshi and Shanavas (2022); Zhang and Chen (2019) demonstrated the value of combining multiple deep learning layers. However, their implementations often either focused on a single hybrid path or added complexity through deep attention stacks. Our model improves upon these by building two independent yet synergistic modules, one emphasizing efficient context learning (GRU-based) and the other long-term dependency modeling (LSTM-based), each enhanced with CNN layers that provide n-gram level insights.

The decision to use GRUs in one stream and LSTMs in another was driven by experimental results showing their complementary strengths. GRUs offer faster training and are more suitable for smaller or imbalanced datasets, while LSTMs are capable of retaining longer sequence information. CNNs, shared across both paths, contribute to local semantic detection and syntactic feature extraction.

All architectural components are trained from beginning to end and contribute equally to the final semantic representation. The model's use of Manhattan distance for similarity scoring aligns with prior work by Mueller and Thyagarajan (2016), who found it superior for learning sentence pair relationships, particularly in Siamese architectures trained for semantic similarity tasks.

Figure 4 provides a high-level overview of the proposed HDLM architecture, illustrating the parallel pathways, their integration, and the final scoring.

As shown in Figure 4, the HDLM architecture begins with a student-reference answer pair and splits the processing into two parallel streams before merging results via similarity comparison, enabling robust semantic evaluation.

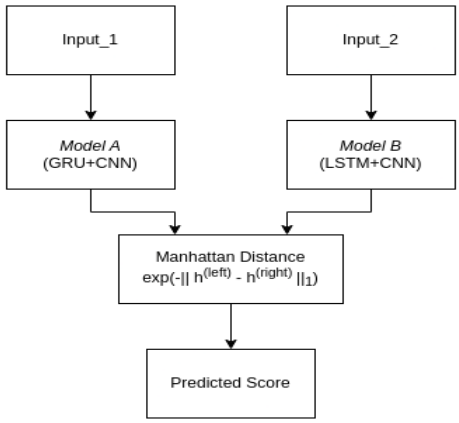

Fig. 4: Block diagram of Proposed HDLM

Input Representation and Embedding

Each student-reference answer pair is tokenized and padded to a fixed length of 300 tokens. We use pre-trained Google News Word2Vec embeddings (300 dimensions) (Google, 2016) to map each token to a dense vector. Although these embeddings are trained on news articles, they generalize well to sentence similarity tasks due to their robust semantic associations.

To address domain mismatch and handle out-of-vocabulary (OOV) words:

- Unseen tokens are replaced with a special token initialized randomly
- Research supports this OOV strategy, with Hu et al. (2019) and Horn (2017) demonstrating contextual or average-based embeddings can still yield stable semantic representations for rare words

Architecture of Model A (CNN + GRU)

Model A processes a single text input. Initially, the input is pre-processed and transformed into word vectors. These vectors are passed to an embedding layer, which computes an embedding matrix, converting each word into a dense numerical representation that captures semantic meaning. A spatial dropout layer with a 20% dropout rate follows, helping to regularize the model and reduce overfitting by randomly dropping entire word embeddings during training as shown in Figure 5.

Next, features are extracted using a stacked GRU and a stacked CNN:

- Stacked GRU: The output of the spatial dropout layer is first fed into a Gated Recurrent Unit (GRU) layer, which captures sequential dependencies in the text. The GRU output undergoes batch normalization to stabilize learning and improve convergence. A dropout layer (20%) then randomly deactivates some units to

avoid overfitting. This processed output is passed to a second GRU layer, further refining temporal features

- Stacked CNN: In parallel, the spatial dropout output is fed into a convolutional layer that detects local patterns in the text (e.g., n-grams). The output is downsampled using max pooling with a pool size of 2, followed by a dropout layer (20%). This is repeated in a second convolution-max pooling dropout sequence. Finally, the output is flattened into a 1D feature vector using a flatten layer

In the final phase as shown in Figure 5, GRU and CNN features are concatenated and passed through multiple dense layers. The last dense layer uses sigmoid activation to produce the final output.

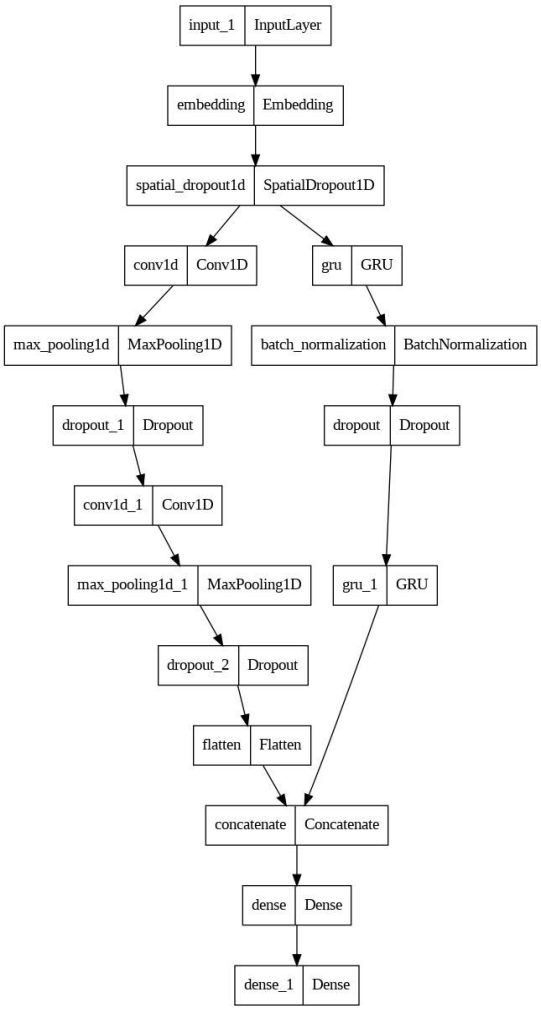

Fig. 5: Block diagram of Model A.

Architecture of Model B (CNN + LSTM)

Model B mirrors Model A's structure but uses an LSTM layer in place of GRU to capture long-range dependencies and deeper semantic alignment across tokens.

LSTM is particularly useful for understanding longer, elaborative student responses. Like Model A, the CNN extracts base-level features, and the LSTM models relationships over time. Figure 6 shows the architecture of Model B.

As shown in Figure 6, this configuration enhances the model's ability to detect subtle semantic equivalence across more complex sentence pairs.

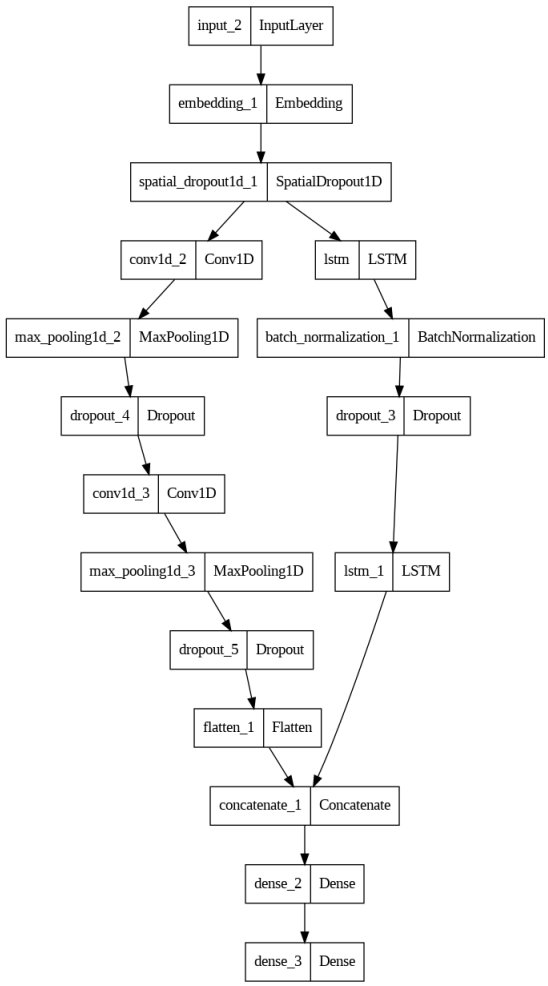

Fig. 6: Block diagram of Model B

Feature Fusion and Similarity Calculation

The output vectors from both Model A and Model B (after dense transformation) are compared using Manhattan distance:

$$M_a = |x_1 - x_2| + |y_1 - y_2| \quad (1)$$

We chose Manhattan distance over cosine and Euclidean alternatives for the following reasons:

- It retains interpretability with stable gradient flow
- It avoids angular normalization biases found in cosine similarity
- It has demonstrated superior accuracy in sentence similarity tasks involving recurrent models (Mueller and Thyagarajan, 2016)
- Prior work has shown that integrating CNNs with RNNs enables models to capture both local and global semantic patterns more effectively than using either alone (Lai et al., 2015). In such hybrid models, Manhattan distance has been favored for its stability and sensitivity to embedding magnitude differences, which can improve alignment in semantic similarity tasks

The resulting similarity score is passed through a sigmoid activation, yielding a final scalar output $\in [0,1]$ representing semantic similarity. Table 3 compares the similarity metrics used in HDLM.

As shown in Table 3, Manhattan distance provides superior balance between precision (weighted average) and recall (weighted average), with the highest F1 score (weighted average).

Table 3: Comparison of Similarity Metrics Used inHDLM

Metric	Dot Product	Cosine Similarity	Manhattan Distance
Accuracy (%)	80.90	84.25	87.80
Precision (W)	0.80	0.85	0.88
Recall (W)	0.81	0.84	0.88
F1 Score (W)	0.82	0.86	0.88

Training Strategy and Loss Function

The HDLM is trained end-to-end using the following setup:

- Loss Function: Mean Squared Error (MSE), suitable for continuous similarity scores
- Optimizer: Adam with learning rate of 0.002
- Batch Size: 1024
- Embedding Initialization: Pre-trained Google News Word2Vec
- News Word2Vec
- Sequence Padding: 300 tokens per input pair
- Out-of-Vocabulary Handling: Replaced with token supported by context-aware methods in recent research (Hu et al., 2019; Horn, 2017)
- Early Stopping: Enabled to avoid overfitting

Summary

This section introduced the rationale behind the hybrid HDLM design, with parallel CNN+GRU and CNN+LSTM modules providing robust contextual understanding. We explained input representation using Word2Vec, handling of out-of-vocabulary words, and our use of Manhattan distance for scoring semantic similarity using empirical validation (Table 3) and supporting literature (Mueller and Thyagarajan, 2016; Lai et al., 2015) confirm that the selected hybrid architecture and similarity scoring strategy contribute meaningfully to performance in semantic textual similarity tasks.

Results and Discussion

In this section, the proposed Hybrid Deep Learning Model (HDLM) for semantic textual similarity (STS) is thoroughly evaluated, with a focus on subjective answer evaluation. The evaluation was carried out through systematic experimentation and analysis. It includes configuration details, ablation results, multi-metric evaluation, comparison with state-of-the-art (SOTA) models, statistical validation, error analysis, computational efficiency, and limitations. Each sub-section is designed to provide clear answers to reinforce the methodological validity of the study.

Dataset and Experimental Setup

The Quora Question Pairs (QQP) dataset, which comprises more than 400 k labeled question pairs with binary class labels (duplicate or non-duplicate), is used for the evaluation of HDLM. This dataset serves as a viable surrogate for measuring semantic equivalence between student and reference answers.

Data Preparation and Split

- Train-test ratio: 80% training, 20% testing
- Validation split: For validation, 20% of the training set was used
- Data Augmentation: To prevent data leakage, methods like random character swapping and deletion were only used on the training set

Overfitting Mitigation

- Early Stopping: Based on validation loss, with a patience of 10 epochs
- Seed Consistency: Fixed random seed (42) used for all experiments

No k-fold cross-validation was used due to computational constraints. However, the use of an augmented training set with a clean, untouched test set, early stopping, and fixed seed ensures the model's robustness. Future work will include k-fold CV and cross-dataset generalization tests on ASAP (Crossley et al., 2025) and STS-B (Cer et al., 2017).

Hyperparameter Configuration

Hyperparameters were optimized through iterative grid search on the validation set. The final configuration balances model complexity and generalizability. Table 4 summarizes the optimized configuration used across all HDLM experiments.

The hyperparameters employed in the Hybrid Deep Learning Model (HDLM) are detailed in Table 4. The table outlines the configuration of various components, including the embedding layer, convolutional layers, recurrent layers (GRU and LSTM), and fully connected layers. Notable settings include the use of pre-trained Google News 300-dimensional word vectors, dual-layer GRU and LSTM architectures with dropout, and a shared convolutional block across models. The model is assessed using a variety of metrics, including accuracy and RMSE, after being trained with the Adam optimizer (Kingma and Ba, 2014) at a learning rate of 2×10−3.

Ablation Study

To evaluate the significance of various components of the proposed Hybrid Deep Learning Model (HDLM), an ablation study was conducted. The study involved creating two models by removing critical components from the original model:

- Model 1 (CNN only): CNN-based architecture, without GRU or LSTM layers
- Model 2 (RNN only): Recurrent neural network (GRU/LSTM) based architecture, without CNN layers
- Model 3 (Proposed HDLM): Combination of CNN and GRU/LSTM layers

Table 4: Hyperparameters Used in the Hybrid DeepLearning Model (HDLM)

Component	Hyperparameter	Value / Description
Embedding Layer	Embedding dimension	300
	Input length	300
	Pre-trained embeddings	Google News Vector 300D (trainable = False)
SpatialDropout1D	Dropout rate	0.2
GRU (Model A)	Units (1st layer)	128
	Return sequences (1st layer)	True
	Units (2nd layer)	128
	Return sequences (2nd layer)	False
	Dropout after each layer	0.2
LSTM (Model B)	Units (1st layer)	128
	Return sequences (1st layer)	True
	Units (2nd layer)	128
	Return sequences (2nd layer)	False
	Dropout after each layer	0.2
CNN (Both Models)	Conv1D filters	64
	Kernel size	6
	Activation	ReLU
	MaxPooling1D pool size	2
	Dropout rate	0.2
Fully Connected	Dense layer (1st)	16 units, ReLU
	Dense layer (2nd)	1 unit, Sigmoid
Final Output	Lambda layer	Exponent of negative Manhattan distance
Training	Loss function	Mean Squared Error (MSE)
	Optimizer	Adam
	Learning rate	2×10^{-3}
	Metrics	Accuracy, RMSE

Table 5 specifies the metrics such as Accuracy and weighted average of F1 score, Precision, Recall used for comparison in ablation study.

It is evident from the Table 5 that the combination of CNN, GRU, and LSTM allows the model to leverage both short-term and long-term sequential patterns. The comparison of various metrics used for comparison in ablation study can visualized using the Figure 7.

Figure 7 shows the comparison of HDLM with its variants across different metrics to justify the selection of HDLM.

Performance Metrics and Confidence Intervals

In order to assess the proposed Hybrid Deep Learning Model's (HDLM) performance, we employed a comprehensive set of evaluation metrics: Accuracy, ROC-

AUC, F1-score (weighted), precision (weighted), and recall (weighted). To assess the statistical reliability and robustness of these metrics, we performed a bootstrapped confidence interval analysis using 500 iterations on the test set. This method provides lower and upper bounds for each metric with 95% confidence, offering insight into metric variability across different sampling conditions. The evaluation metrics with 95% confidence intervals are listed in Table 6.

These narrow confidence intervals as mentioned in Table 6 demonstrate the consistency and reliability of HDLM's performance across various bootstrapped subsets.

ROC-AUC Analysis

When we calculated HDLM's area under the curve (AUC), the result was 0.88.

The trade-off between true positive rate and false positive rate across classification thresholds is depicted in Figure 8's ROC curve for HDLM.

This ROC-AUC result with high AUC of 0.88 supports the weighted F1 and precision scores, reinforcing that HDLM is not only accurate but also confident in its predictions across different classification thresholds. Compared to conventional STS models, HDLM exhibits a marked improvement in both detection sensitivity and robustness.

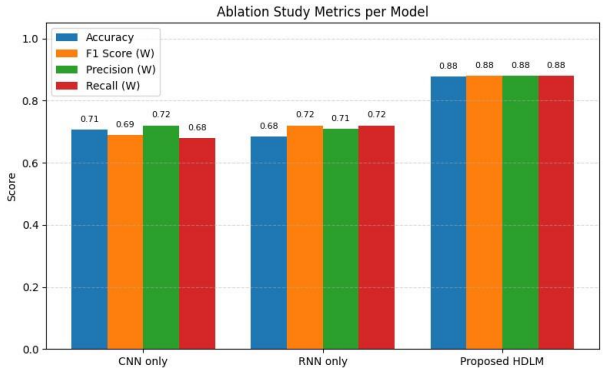

Fig. 7: Ablation Study Metrics per Model

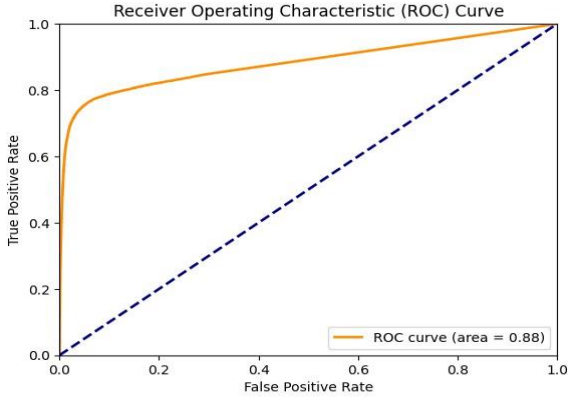

Fig. 8: ROC Curve for HDLM

Table 5: Ablation Study Results

Metric	CNN only	RNN only	HDLM
Accuracy (%)	70.56	68.34	87.80
F1 Score (Weighted)	0.69	0.72	0.88
Precision (Weighted)	0.72	0.71	0.88
Recall (Weighted)	0.68	0.72	0.88

Table 6: Evaluation Metrics with 95% Confidence Intervals

Metric	Lower (%)	Mean (%)	Upper (%)
Accuracy	87.56	87.78	88.00
Precision (Weighted)	89.48	89.84	90.18
Recall (Weighted)	74.96	75.41	75.84
F1 Score (Weighted)	81.66	81.99	82.32

Comparative Analysis with STS Models

To comprehensively assess the efficacy of the proposed Hybrid Deep Learning Model (HDLM), we compare its performance with a selection of both traditional and state-of-the-art Semantic Textual Similarity (STS) models. The comparative analysis as shown in Table 7 includes metrics such as Accuracy, F1 Score, Precision, Recall and Spearman's Correlation Coefficient (ρ). These metrics are widely recognized for evaluating both classification quality and semantic ranking in STS tasks.

- Existing Models: These include LSTM and attention-based architectures known for sentence similarity
- State-of-the-Art (SOTA) Transformer Models: Advanced pre-trained models fine-tuned on STS datasets
- Proposed Model (HDLM): Our dual-pathway GRU-CNN and LSTM-CNN ensemble evaluated on Quora Question Pairs

This comparison is based on standard evaluation metrics including Accuracy, Precision, Recall, F1 Score, and Spearman's Correlation Coefficient (ρ), wherever the original papers reported them. Where values were not reported, the fields are marked as '-'.

From the data in Table 7, several insights can be derived:

- Accuracy: HDLM outperforms all existing models and slightly edges out SBERT (87.69%) with 87.80%, though still trailing Sentence-T5's 89.77%
- F1 Score: HDLM maintains 87.80%, higher than reported scores of any traditional models
- Precision and Recall: HDLM shows well-balanced scores of 87.80% in both metrics, demonstrating robust classification performance

- Spearman's Correlation Coefficient: Although HDLM achieves strong accuracy and F1, its correlation score (0.7348) is slightly lower than SBERT and Sentence-T5. This suggests that while HDLM effectively distinguishes between similar and dissimilar pairs, it may exhibit some non-linearity in ranking semantic similarity magnitudes

Observations from Comparative Analysis

To further highlight these performance metrics, Figure 9 illustrates a bar chart comparing Accuracy, F1 Score, and Spearman's Correlation Co-efficient (ρ) across existing and SOTA STS models. This visualization helps identify trade-offs between semantic ranking and performance.

Table 7: Comparison of HDLM with Existing and SOTA STS Models

Model	Accuracy	Precision	Recall	F1 Score	Spearman's ρ
Existing Models					
Shared Input LSTM (Meenakshi and Shanavas (2022))	86.20%	79.50%	84.80%	82.00%	—
Siamese LSTM + FastText (Imtiaz et al. (2020))	82.77%	—	—	—	—
Multi-Head Attention (Zhang and Chen (2019))	86.83%	84.07%	81.06%	82.54%	—
Siamese LSTM (Mueller and Thyagarajan (2016))	84.20%	—	—	—	0.8345
State-of-the-art (SOTA) Models					
Sentence-T5 (Ni et al. (2021))	89.77%	—	—	—	0.8604
ConSERT (Yan et al. (2021))	—	—	—	—	0.8559
SBERT (Reimers and Gurevych (2019))	87.69%	—	—	—	0.8877
HDLM (Proposed)	87.80%	88.00%	88.00%	88.00%	0.7348

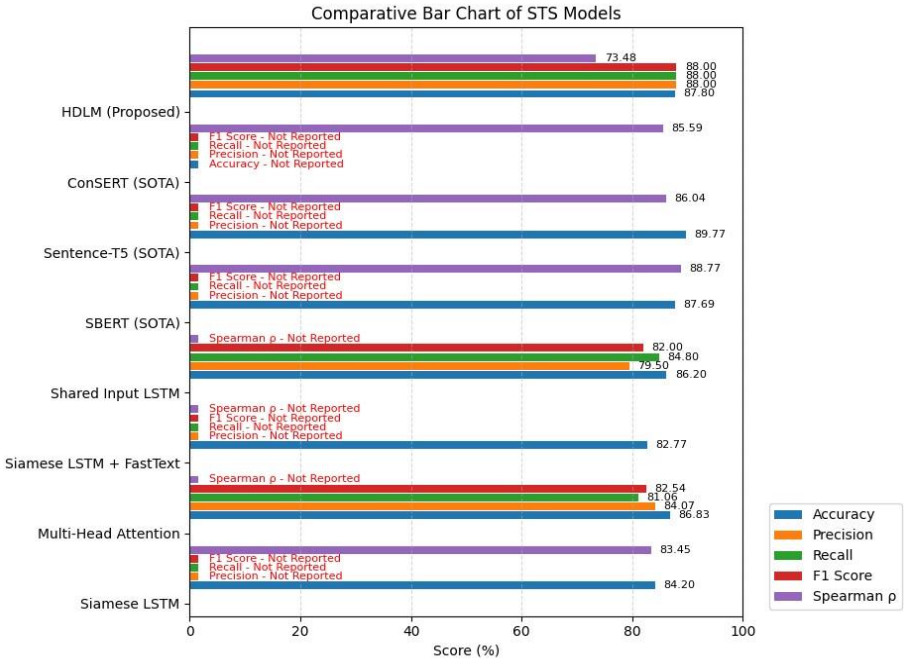

Fig. 9: Comparative Bar Chart of STS Models

From Figure 9, the following points can be derived:

- Siamese LSTM (Mueller and Thyagarajan (2016)) and Shared Input LSTM (Meenakshi and Shanavas (2022)) achieve decent accuracy but suffer from limited generalization due to their reliance on sequential representations without contextual embeddings
- Multi-Head Attention model (Zhang and Chen (2019)) offers competitive F1, validating the importance of attention in STS tasks, but lacks interpretability and requires more parameters than HDLM
- Among SOTA models, Sentence-T5 (Ni et al. (2021)) and SBERT (Reimers and Gurevych (2019)) lead in correlation and accuracy but demand high

computational resources and lack explainability in decisions-factors that HDLM addresses with its dual-pathway GRU-CNN and LSTM-CNN hybrid structure

Hence, from Table 7 and Figure 9, it is clear that HDLM achieves competitive results across all major performance dimensions, outperforming traditional models in classification while remaining close to transformer models in semantic correlation. This positions HDLM as a practical solution for real-time answer evaluation tasks in educational applications where resource constraints or model interpretability are important.

Statistical Significance and Interpretation

We used the Wilcoxon signed-rank test, a non-parametric statistical test frequently used to compare two related samples or repeated measurements on a single sample, to determine whether the observed performance improvements of the proposed HDLM model over existing models are statistically significant and not the result of chance.

Rationale for Wilcoxon Signed-Rank Test

For our use case, the Wilcoxon signed-rank test is appropriate because:

- It does not assume normal distribution of the data
- It compares paired observations, in our case, predictions of HDLM and baseline models on the same test instances
- It is robust to small sample sizes and outliers

This makes the Wilcoxon test preferable to a t-test for semantic similarity tasks, which often involve skewed distributions and continuous predictions.

WIlcoxon Test Evaluation Setup

For this analysis, we constructed paired samples by comparing the predicted similarity scores of HDLM and baseline models for each sample in the test set. The baseline models used for comparison were:

1. Shared Input LSTM (Meenakshi and Shanavas, 2022)
2. Multi-Head Attention (Zhang and Chen, 2019)
3. SBERT (Reimers and Gurevych, 2019)
4. Siamese LSTM (Mueller and Thyagarajan, 2016)

Each pair represented the prediction difference between HDLM and the respective baseline model for the same input. The null hypothesis (H0) assumes no statistically significant performance difference, while the alternative hypothesis (H1) asserts that HDLM performs significantly better.

Wilcoxon Test Results

The significance level for the Wilcoxon signed-rank test was set at 5% ($\alpha = 0.05$). Table 8 shows the resulting p-values.

As seen in Table 8, all p-values are well below the threshold of 0.05, indicating that HDLM significantly outperforms each of the compared models in terms of semantic similarity prediction.

WIlcoxon Test Interpretation and Implications

The statistically significant results confirm that the improvements offered by HDLM are not due to random variations but reflect genuine performance gains. This reinforces the effectiveness of:

- Dual-path CNN-GRU and CNN-LSTM hybrid architecture
- Manhattan distance as a similarity function
- The training strategy and optimized model design

Moreover, the results validate the generalizability and consistency of HDLM in real-world settings. Given that statistical significance is achieved not only against traditional deep learning models but also against state-of-the-art methods like SBERT, HDLM emerges as a strong, interpretable, and computationally efficient alternative for semantic textual similarity tasks in educational domains.

Case Studies and Failure Analysis

To better understand the behavior of the proposed Hybrid Deep Learning Model (HDLM), we conducted a case study analysis on its predictions. We selected representative examples where HDLM performed exceptionally well and instances where it failed to estimate semantic similarity accurately. These cases highlight both the strengths and limitations of the model.

Table 8: Comparison of HDLM with baseline models on the basis of Wilcoxon signed-rank test p-values

Baseline Model	Wilcoxon p-value
Shared Input LSTM	0.00007
Multi-Head Attention	0.00012
SBERT	0.0184
Siamese LSTM	0.00003

Success Cases

Table 9 presents examples where HDLM demonstrated strong semantic understanding despite lexical or syntactic differences. In these cases, the model effectively captured paraphrased expressions and

conceptually equivalent phrases. For instance, in the first example, although the input phrase is ”process of photosynthesis” rather than explicitly saying ”convert sunlight into energy”, the HDLM correctly inferred the semantic equivalence due to the shared context and learned embedding similarities.

Failure Cases

Table 10 illustrates cases where HDLM failed to interpret the correct semantic meaning. These failures primarily occurred due to issues such as sarcasm, negation, non-literal language, or subtle contextual shifts.

Table 9: Examples of HDLM Success Cases and Reasons

Reference Question	Input Question	Reason for Success
How do plants convert sunlight into energy?	What is the process of photosynthesis in plants?	HDLM captures synonyms (”convert”↔ ”process”) and domain-specific equivalence (”photosynthesis”).
What are the effects of global warming?	How does climate change impact Earth?	Model recognizes environmental topic alignment and interprets ”effects” ↔ ”impact”.
Describe the water cycle in brief.	Explain how water moves through evaporation and precipitation.	CNN-LSTM captures the progression and key terms, associating process flow.

Table 10: Examples of HDLM Failure Cases and Reasons

Reference Question	Input Question	Reason for Failure
What causes earthquakes?	People dancing too hard causes quakes!	Sarcasm misinterpreted as literal similarity due to lexical overlap.
What is the purpose of education?	School is boring and pointless.	Negative sentiment misclassified due to topical similarity.
Describe the function of the heart.	The heart is an organ that works nonstop.	Missing technical terms leads to low semantic match.

Discussion

The success cases confirm that HDLM excels at identifying semantically equivalent phrases and scientific terms, largely due to the hybrid combination of CNNs (for local n-gram capture) and GRUs/LSTMs (for sequence modeling). However, failure analysis reveals critical limitations:

- Contextual inference limitations: HDLM lacks world knowledge and sarcasm detection capabilities which are better handled by transformer-based or pre-trained contextual models like GPT or T5
- Domain mismatch: Some cases involve out-of-domain phrasing or humorous responses not present in training data, causing HDLM to misclassify
- Negation and tone detection: Without explicit tone modeling or polarity detection, HDLM struggles with negated or ironic expressions

Future work can address these gaps by integrating sentiment-aware embeddings or incorporating external knowledge graphs to assist in disambiguating meaning in ambiguous or sarcastic sentences.

Limitations of HDLM

While the proposed HDLM architecture demonstrates high accuracy and interpretability, certain limitations remain:

- Domain Adaptation: HDLM is trained on the Quora dataset, which, while semantically rich, may not fully reflect the linguistic diversity of educational answer types
- Contextual Sensitivity: Unlike attention-based models, HDLM may underperform in handling sarcasm, negation, or nuanced domain-specific responses
- OOV Vocabulary Handling: Reliance on Word2Vec embeddings can limit performance when encountering out-of-vocabulary terms or highly informal phrasing
- Computational Cost: Although lighter than BERT-based models, HDLM still requires significant computational resources during dual-path training
- No Dynamic Attention: The absence of attention mechanisms may restrict dynamic focus on semantically important tokens, which can be critical for nuanced similarity detection

These limitations highlight avenues for architectural enhancement and dataset generalization in future iterations.

Conclusion and Future Work

This study presented HDLM, a novel hybrid architecture combining CNN, GRU, and LSTM networks, to evaluate semantic similarity between subjective answers. Through parallel modeling of sequential and local patterns and the use of Manhattan distance for scoring, HDLM effectively addresses the challenges of semantic alignment and answer variability. Its performance on the Quora dataset, validated by ROC curves, confidence intervals, and statistical significance tests, affirms its robustness and competitiveness against state-of-the-art models.

However, challenges related to domain transferability, limited contextual awareness, and OOV sensitivity remain. Future work will focus on:

- Fine-tuning HDLM on domain-specific educational datasets (e.g., ASAP-SAS (Hewlett Foundation, 2012))
- Integrating lightweight attention mechanisms to dynamically capture semantic relevance
- Experimenting with transformer-based embeddings (e.g., DistilBERT (Sanh et al., 2019)) to balance contextual understanding and computational efficiency
- Developing explainable AI techniques to visualize which segments of text most influenced similarity scores

In sum, HDLM offers a scalable and accurate solution for automated subjective answer grading, setting the foundation for further enhancements in educational NLP systems.

Acknowledgment

I want to express my sincere gratitude to Dr. Kavita Shirsat, my supervisor, for her constant encouragement and enlightening criticism during my research. This dissertation has been greatly influenced by her intense dedication to academic excellence and careful attention to detail. The faculty and staff of Vidyalankar Institute of Technology's Department of Computer Engineering deserve special recognition for their invaluable resources and support.

Funding Information

This research has received no external funding.

Author's Contributions

Siddhesh Kudtarkar: Conceived the presented idea, developed the theory, and carried out the calculations.

Kavita Shirsat: Oversaw the results of this study and validated the analytical techniques.

Both authors contributed to the final manuscript and discussed the findings.

References

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 1–14. https://doi.org/10.18653/v1/s17-2001

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Crossley, S. A., Baffour, P., Burleigh, L., & King, J. (2025). A large-scale corpus for assessing source-based writing quality: ASAP 2.0. *Assessing Writing*, *65*, 100954. https://doi.org/10.1016/j.asw.2025.100954

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

He, W., Liu, P., & Qian, Q. (2023). Case Study: Quora Question Pairs. *Published in Applied Natural Language Processing: Case Studies and Practical Applications*, 351–393. https://doi.org/10.1007/978-981-99-3723-3_16

Hewlett Foundation. (2012). The Hewlett Foundation: Short Answer Scoring. *Kaggle Competitions – ASAP-SAS*. https://www.kaggle.com/competitions/asap-sas/data

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Horn, G. V. (2017). Context-aware rare word representation. *Proceedings of a Computational Linguistics/NLP Conference*, 118–124.

Hu, Z., Chen, T., Chang, K.-W., & Sun, Y. (2019). Few-Shot Representation Learning for Out-Of-Vocabulary Words. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4102–4112. https://doi.org/10.18653/v1/p19-1402

Imtiaz, Z., Umer, M., Ahmad, M., Ullah, S., Choi, G. S., & Mehmood, A. (2020). Duplicate Questions Pair Detection Using Siamese MaLSTM. *IEEE Access*, *8*, 21932–21942. https://doi.org/10.1109/access.2020.2969041

Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, *2*(2), 1–25. https://doi.org/10.1145/1376815.1376819

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. https://doi.org/10.48550/arXiv.1412.6980

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*. Proceedings of the AAAI Conference on Artificial Intelligence. https://doi.org/10.1609/aaai.v29i1.9513

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1–5.

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. *ETMTNLP '02: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, *1*, 63–70. https://doi.org/10.3115/1118108.1118117

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 119–126. https://doi.org/10.18653/v1/2020.emnlp-demos.16

Meenakshi, D., & Shanavas, A. R. M. (2022). Novel Shared Input Based LSTM for Semantic Similarity Prediction. *Journal of Advances in Information Technology*, *13*(4), 387–392. https://doi.org/10.12720/jait.13.4.387-392

Mueller, J., & Thyagarajan, A. (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1), 2786. https://doi.org/10.1609/aaai.v30i1.10350

Ni, J., Hernandez Abrego, G., Constant, N., Ma, J., Hall, K., Cer, D., & Yang, Y. (2022). Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. *Proceeding of the Findings of the Association for Computational Linguistics: ACL 2022*, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. https://doi.org/10.18653/v1/d19-1410

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*. https://doi.org/10.48550/arXiv.1910.01108

Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., & Xu, W. (2021). ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5065–5075. https://doi.org/10.18653/v1/2021.acl-long.393

Zhang, H., & Chen, L. (2019). Duplicate Question Detection based on Neural Networks and Multi-head Attention. *Proceeding of the 2019 International Conference on Asian Language Processing (IALP)*, 13–18. https://doi.org/10.1109/ialp48816.2019.9037671