Journal of Techniques, ISSN: 2708-8383, Vol. 8, No. 2, 2026, Pages 58-70
https://doi.org/10.51173/jt.v8i2.2840

JOURNAL OF TECHNIQUES

Journal homepage: http://journal.mtu.edu.iq

RESEARCH ARTICLE - ENGINEERING (MISCELLANEOUS)

Context-Aware Hybrid Text Generation Integrating Bidirectional Long Short-Term Memory Sequencing with Semantic Clustering

Mustafa Abbas Hussein[1*]

[1]Electronics and Computer Engineering, Çankiri Karatekin University, Çankırı, Türkiye

[*] Corresponding author E-mail: mustafaabbas67@yahoo.com

Article Info.

Article history:

Received
03 January 2026

Revised
06 June 2026

Accepted
15 June 2026

Published
30 June 2026

Abstract

Natural language generation systems sometimes struggle to model long-range semantic connections and maintain contextual consistency, especially when applied to linguistically sophisticated literary corpora. Traditional recurrent neural architectures are good at modeling sequential patterns but typically fail to preserve higher-level thematic and stylistic information in text production. The current paper proposes a semantic-aware hybrid framework based on Word2Vec embedding representations, ++K-Means semantic clustering, and Bidirectional Long Short-Term Memory (Bi-LSTM) sequence learning to improve contextual coherence and next-word prediction performance. In the proposed architecture, the model learns to obtain semantic context vectors from clustered embedding spaces and to fuse them with sequential hidden representations for better language modeling. This study examines the system on three benchmark datasets from English corpora of both literary and general domains: the Nietzsche corpus, Shakespeare plays and WikiText-2. Experimental results show that the proposed semantic-aware recurrent architecture consistently outperforms the standard statistical and neural baseline models. The model achieves prediction accuracies of 67.4%, 61.3%, and 63.1% on the Nietzsche, Shakespeare, and WikiText-2 datasets, respectively, while reducing perplexity values and enhancing linguistic coherence. A more detailed analysis of the robustness test, semantic error evaluation, and ablation experiments confirm that semantic clustering effectively can improve contextual consistency, stylistic preservation, and semantic continuity. The results demonstrate that combining clustering-based semantic abstractions with recurrent sequence modeling is an effective, computationally lightweight approach to context-aware text synthesis for both literary and general-domain applications.

This is an open-access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/)

Publisher: Middle Technical University

Keywords: Semantic Clustering; Word2Vec; Context-Aware; Natural Language Generation.

1. Introduction

The field of Natural Language Generation (NLG) has become increasingly relevant in Artificial Intelligence with the emergence of deep learning, particularly for chatbots and automatic summarization [1]. However, it remains a significant challenge to generate text that reflects the styles of diverse authors, maintains thematic coherence, and demonstrates semantic richness. Neural sequence modeling has replaced traditional language modeling techniques to address the problems of data scarcity and difficulty in capturing nuanced meanings [2].

Recurrent neural networks (RNNs) extended the interpretation of sequential data and enhanced language modeling by propagating context information over time through recurrent hidden states. However, vanilla RNNs suffer from problems storing information due to vanishing and exploding gradients [3]. For this purpose, extended Long Short-Term Memory (LSTM) networks have been constructed that use gating techniques to retain relevant context across extended sequences [4]. Bidirectional LSTM (Bi-LSTM) models improve sequence encoding by considering both past and future contexts. This led to the invention of early neural NLG systems based on LSTM, which were very successful in several areas [5].

LSTM-basedLSTM-based models have made some advancements, but still struggle with linguistically demanding texts. They tend to focus on surface patterns without identifying salient traits, and their outputs, while technically correct, are semantically inconsistent, stylistically unfaithful, and thematically shallow. This is particularly evident in the realm of literary datasets, where language is often used in abstract thinking, metaphors, and antiquated phrases [6, 7].

Distributed word representations, such as Word2Vec and other embedding models, have been created that considerably improve semantic modeling. Words are represented as vectors in a continuous space that captures lexical and semantic commonalities [8]. Word embeddings help to overcome the sparsity problem, generalize better and give more informative lexical representation than the usual one-hot encodings. This is why embeddings have become a common feature of neural language models. However, such embeddings are naturally static, context-free, and thus limited in their ability to encode changes in meaning across diverse settings [9].

Nomenclature and Symbols			
f_t	Fused Feature Representation	$\mathbf{c_k}$	Semantic Cluster Centroid
$\mathcal{L}$	Cross-Entropy Loss	C	Set of Semantic Clusters
$\oplus$	Vector Concatenation	$\mathrm{sim}(\cdot)$	Cosine Similarity Function
η	Learning Rate	θ	Trainable Model Parameters
PPL	Perplexity	K	Number of Semantic Clusters
ACC	Prediction Accuracy	T	Sequence Length
BLEU	Bilingual Evaluation Understudy	ROUGE-L	Longest Common Subsequence Metric
METEOR	Semantic Evaluation Metric	SPI	Style Preservation Index
Bi-LSTM	Bi-Directional Long Short-Term Memory	e_i	Word Embedding Vector

Clustering in embedding space is one of the most important techniques for extracting latent semantic structure in high-dimensional vector representations [10]. Clustering techniques are used to discover hidden conceptual information, such as thematic relevance, contextual affinity, emotional traits, and semantic dependencies among words, by grouping semantically related keywords [11]. Such higher-level semantic representations are an efficient means of eliminating redundant variants and improving feature expressiveness for downstream language processing applications. Clustering approaches have been found useful in several applications such as subject identification, semantic retrieval, conversational modeling, and intelligent text analysis systems [12]. However, in most existing natural language production systems, clustering is used only in the initial steps of data preparation. The generation of adaptive, sequence-aware semantic context vectors from clustering structures and their direct integration into recurrent deep learning systems have rarely been addressed. Therefore, the potential of clustering-driven contextual guidance for improved dynamic sequence modeling remains largely unexplored in present NLG systems [13].

The constraint of sequential models is that neural sequence modeling hybrid strategies incorporate the contextual or semantic information [14]. Two ways that can selectively attend to relevant tokens are attention mechanisms and syntactic features [15]. Transformer-based systems with self-attention have achieved outstanding performance on natural language generation challenges but require large datasets and substantial computational resources. They are less appropriate for controlled investigations aiming at mild semantic enrichment in recurrent frameworks [16]. They decline in data-poor or specialized-language domains.

There is a growing demand for research on context-aware NLG designs that effectively integrate semantic abstractions with sequential modeling. Embedding methods do not provide systematic semantic guidance, and standard LSTMs are unable to emphasize semantically relevant parts of sequences [17]. The dynamic application of clustering methods combined with interpretable semantic abstractions in text generation is rarely employed [18]. Furthermore, the performance of semantically enriched recurrent models on complex datasets of literary texts, as well as their robustness to noise and diverse data sizes, remains an open question [19].

RNNs are useful for deep learning language production systems, since they can model sequential dependencies and employ shifting hidden states to improve context and fluency [20]. But at larger distances, they are destroyed due to gradient instability. LSTM networks' gating strategies better capture long-term dependencies, govern the flow of information, and perform better on tasks like text synthesis and machine translation [21].

Bi-LSTM architectures analyze sequences in both directions, which boosts their performance on challenging language tasks and helps them better absorb contextual information. Sequential techniques still cannot capture deeper semantic patterns [22]. This can result in work that is neither cognitively sophisticated nor thematically coherent.

Distributed word representations are among the most significant breakthroughs in language modeling. Methods such as Word2Vec embed words into dense vectors in multi-dimensional spaces and enable models to learn fine-grained syntactic and semantic links, thereby alleviating the sparsity problem of classical representations [23]. But these embeddings are static and hence cannot reflect changes in meaning depending on context [24].

Many research efforts are moving towards hybrid strategies that combine sequential modeling with additional semantic components to address the limitations of neural models [25-27]. Such approaches take into account not just the sequence of words but also other factors, such as grammatical structures and contextual cues [28, 29]. This is due to the greater efficiency and interpretability of the attention processes. Also, clustering-based algorithms have been proposed to detect latent semantic structures in large text corpora, uncovering patterns that improve the coherence, consistency, and intelligibility of the output text [30].

Clustering is widely used for exploratory analysis and topic recognition, but it is poorly integrated into neural prediction pipelines. The benefits of integrating this semantic information, obtained through clustering, with recurrent neural models for next-word prediction have not been examined in detail [31, 32].

In this paper, a hybrid context-aware text generation system that generates semantic context vectors using Word2Vec embeddings and ++K-Means clustering is proposed. This study injects these vectors into a Bi-LSTM model. This is an attempt to improve sequential modeling through explicit semantic abstraction, enabling the model to learn high-level themes and local interactions in text production. This study offers a lightweight, context-aware NLG framework by directly embedding semantic abstractions into recurrent modeling pipelines, helping bridge the gap between sequential learning and semantic understanding.

2. Materials and Methods

This study proposes a unique framework that extends the semantic context extraction by using recurrent neural architectures to increase the coherence, contextual consistency and accuracy of the resulting text sequences. The proposed method comprises five crucial phases: data collection and preparation, text preprocessing and development of text embeddings, extraction of a semantic context vector via clustering, construction of an RNN-LSTM network, and model training and performance evaluation. The textual data is cleaned and turned into distributed vector representations to capture the semantic links among words. The study then uses clustering algorithms to obtain semantic context vectors that capture high-level contextual information for sequence learning. The recovered semantic representations are then incorporated into the

RNN-LSTM architecture to improve contextual dependency modeling in text production. Finally, the performance of the proposed system is evaluated by several assessment measures for prediction performance and generation quality. Fig. 1 displays the overall workflow of the suggested methodology.

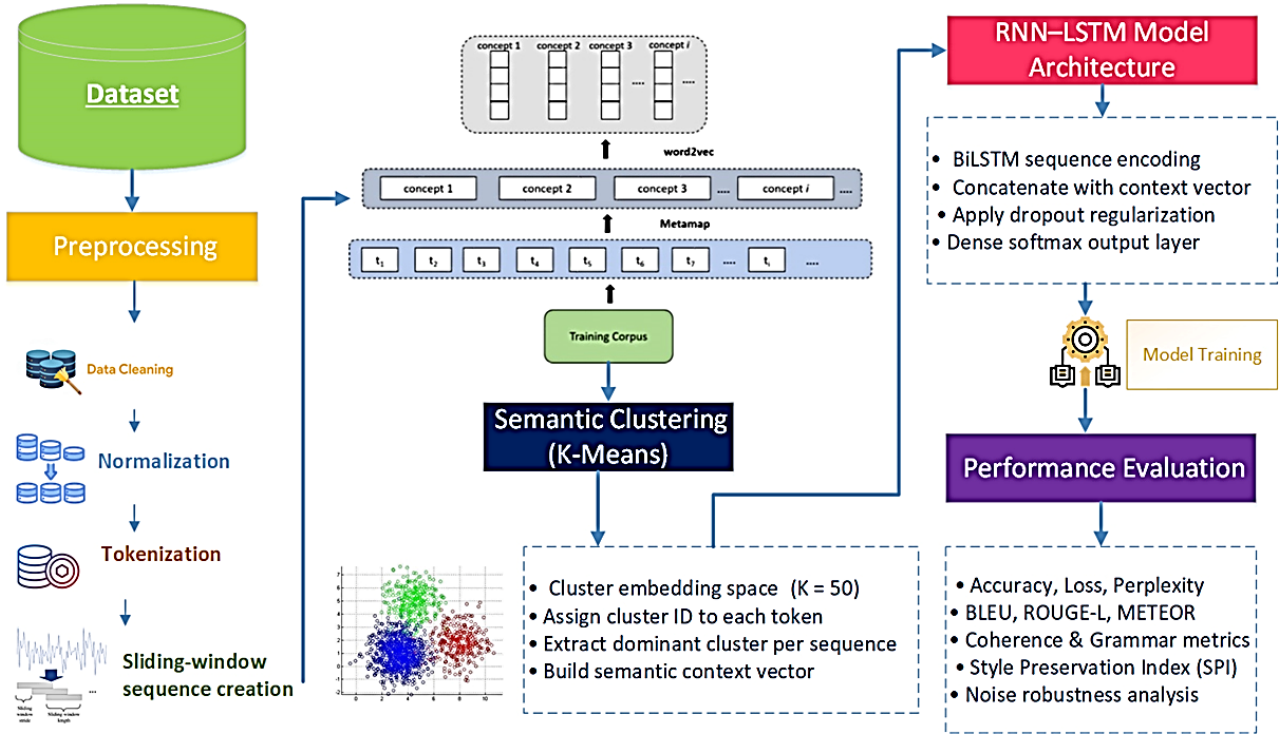

Fig. 1. Overall workflow of the proposed context-aware text generation framework

2.1. Datasets description

The experimental datasets required for this investigation were obtained from Kaggle public datasets to ensure repeatability, transparency, and availability for subsequent research. To thoroughly evaluate the proposed context-enhanced RNN-LSTM architecture, the study selected three benchmark textual corpora: the Friedrich Nietzsche corpus [33], the Shakespeare Plays corpus [34], and the WikiText-2 dataset [35]. The study carefully selected these datasets to include a diverse mix of linguistic features, literary forms, and contextual challenges, to evaluate the model's ability to handle semantic dependencies, stylistic variation, and long-range context in text generation tasks.

The corpus comprises approximately 2.1 million textual tokens and almost 18,742 unique vocabulary entries from Friedrich Nietzsche's philosophical writings. The corpus is characterized by intricate intellectual phrasing, metaphorical language and abstract ideas. Thus, it provides a useful benchmark for evaluating the ability of language generation algorithms to detect complex semantic relations and maintain logical coherence across long sequences of text. On the other hand, the Shakespeare Plays dataset comprises over 24,000 lexical types and approximately 3.4 million tokens from renowned theatrical works such as Othello, Macbeth, and Hamlet. The collection is described with emotionally loaded language patterns, dramatic dialogue styles and lyrical idioms. Hence, it provides a suitable setting to study the possibility of deep learning architectures for reproducing style, emotional context, and sequential consistency in the generation of literary works.

The third dataset, WikiText-2, contains 33,278 distinct words and approximately 2 million tokens, collected from diverse Wikipedia articles across different expertise areas. WikiText-2 contains a large variety of topics and informative, generic language patterns compared to the literary corpora. The dataset is useful for examining the potential generalization of the proposed framework to additional textual domains, to ensure the contextual continuity and semantic purity of the sequence development.

2.2. pre-processing

Before training the proposed RNN-LSTM architecture, a robust multistage preparation pipeline was established to ensure the quality and reliability of the textual input required for sequential learning. The main aim of this preprocessing system is to convert unstructured textual corpora into semantically coherent, ordered sequences for neural language modeling applications. The preprocessing strategies not only reduce noise and redundancy but also improve the model's contextual consistency and next-word prediction accuracy during training. The initial phase of preprocessing comprises text cleaning and normalization. In this step, the corpora were cleaned by removing unnecessary textual elements such as punctuation marks, numerical values, special characters, HTML elements and non-alphabetic symbols. These noisy parts are removed, leaving only meaningful textual information for subsequent processing phases. This cleaning process helps to reduce the training data mismatches and the superfluous vocabulary complexity

After cleaning, all texts are converted to lowercase for better embedding uniformity and vocabulary homogeneity. This normalization method significantly reduces vocabulary sparsity by treating words with different capitalization patterns, e.g., "King" and "king", as the same lexical elements. Thus, the semantic representation of words is more robust and computationally more efficient for buildingfor building embeddings and learning sequences. Then, the text is normalized and segmented into discrete lexical units using word-level tokenization methods. Tokenization converts unstructured streams of text into well-structured sequences of tokens that are easily understood by deep neural

architectures. At this level, the vocabulary is a bit extensive, but the sequential linkages between the words remain intact. The resulting token sequences are then used as the primary input representation for the proposed recurrent neural network model, enabling effective contextual learning and sequential dependency extraction.

2.3. Word embedding and vector representation

After the preparation stage, the Word2Vec embedding approach was used to convert the cleaned text sequences to dense numerical representations for deep sequential learning. Word embeddings are vector representations of words such that related words in terms of contextual meaning are close to each other in the vector space. This representation allows neural circuits to encode grammatical links, contextual associations, and semantic similarities more effectively than earlier sparse encoding schemes.

The study trained several Word2Vec models independently for each dataset to retain the semantic traits and stylistic trends peculiar to each corpus. The proposed framework allows embeddings to be trained directly on the Friedrich Nietzsche and Shakespeare corpora, thereby enabling the learning of domain-specific linguistic structures, philosophical terminology, literary expressions, and contextual writing styles that are not well represented by generalized pre-trained embeddings. This dataset-specific embedding strategy enhances contextual learning for text generation and increases the model's capacity to capture complex semantic information.

To maintain trial consistency and a fair comparison of performance, the study trained Word2Vec with a fixed set of hyperparameter combinations across all datasets. The obtained embedding vectors were used as the input representation layer in the proposed RNN–LSTM architecture to model sequences efficiently and incorporate semantic context. The hyperparameters used to generate the embeddings are detailed in Table 1.

Table 1. Configuration settings for training custom word embedding models

Hyperparameter Category	Configuration Value
Embedding Vector Size	200 Dimensions
Context Window Radius	5 Tokens
Minimum Token Occurrence	3 Instances
Embedding Architecture	Skip-Gram Model
Negative Sampling Rate	10 Negative Samples
Training Iterations	15 Epochs

The study adopted the distributed embedding strategy, a powerful approach for generating compact semantic representations that is computationally efficient for sequential language modeling tasks. The selected embedding method is an effective way to capture semantic regularities and lexical relations with relatively lightweight architectures, as opposed to transformer-based contextual encoders (BERT or ELMo), which require significant computational resources and much larger training corpora. This makes it extremely useful for controlled experimental situations including recurrent neural frameworks and semantic augmentation methods. Also essential is the interoperability of the embedding technology with clustering-based semantic analysis. Clustering algorithms can reveal coherent semantic structures with high stability, as the generated vector spaces preserve crucial geometric and semantic relations between words. These properties allow us to derive representative semantic context vectors that can effectively help later neural sequence modeling techniques. The clustering operations and contextual feature learning are further enhanced in robustness by preserving semantic continuity in the embedding space. Furthermore, the selected skip-gram training mechanism is well-suited for learning informative representations for uncommon, stylistically distinct and context-sensitive vocabulary items that are often present in literary collections such as Shakespearean dramatic works and Nietzsche's philosophical writings. This property is vital for retaining the semantic richness in complex literary-language production tasks. However, highly contextualized transformer models tend to be more sophisticated in construction, which may not be compatible with lightweight recurrent systems for interpretable semantic augmentation.

After the embedding training phase, the resulting vector spaces displayed clearly defined semantic clusters and contextually significant separations of lexical elements. In the recommended approach, the following clustering-based context-extraction step relied heavily on semantic structures. Table 2 presents the statistical properties of the final embedding models obtained during training.

Table 2. Quantitative analysis of the trained word embedding representations

Text Corpus	Unique Vocabulary Terms	Mean Embedding Magnitude	Average Semantic Neighbor Similarity	Embedding Outlier Ratio
Nietzsche Corpus	18,742	3.91	0.68	2.4%
Shakespeare Corpus	24,615	3.88	0.72	3.1%
WikiText-2 Corpus	33,278	3.95	0.70	2.8%

Semantic outliers correspond to embedding vectors whose magnitudes differ substantially from the global statistical distribution, generally associated with low-frequency lexical items.

This study describes the main reasons for selecting the Word2Vec framework as the main embedding approach. First, the embedding approach allows for computationally efficient distributed vector representations of lexical entities while preserving important semantic relations among lexical elements. Deep contextual representations can be generated by architectures such as BERT and ELMo, though they often require larger datasets, greater processing capacity, and more advanced training methods. In contrast, Word2Vec is better suited to recurrent neural architectures for controlled semantic enhancement experiments and provides a lean, efficient technique for learning semantic representations from large textual corpora.

Second, the learned embedding space greatly enhances the effectiveness of clustering-based semantic analysis with consistent geometric and linear semantic links between words. The stable nature of these vector representations enables the reliable discovery of coherent semantic groupings using clustering algorithms such as k-means.

Third, the skip-gram training technique for learning embeddings is well-suited to acquiring informative representations of rare, stylistically unique, and context-sensitive lexical items, which are common in literary corpora such as the works of Friedrich Nietzsche and Shakespeare. There is domain-specific vocabulary, metaphorical structures and unusual formulations in literary collections that require consistent semantic representation learning. For recurrent systems that require efficient semantic enrichment with low computational overhead, lightweight embedding approaches are a more feasible alternative than transformer-based contextual methods.

The embedding training approach resulted in vector spaces with stable lexical groupings and clear semantic distinctions. The proposed methodology included an additional clustering-based context extraction stage built on top of these semantic structures. Table 3 presents the exact parameters of the setting of the ++K-Means clustering procedure.

Table 3. Configuration parameters of the semantic clustering process

Clustering Parameter	Selected Configuration
Number of Semantic Clusters	50
Centroid Initialization Strategy	++K-Means
Maximum Optimization Iterations	300
Similarity Distance Metric	Euclidean Distance
Random Initialization Seed	42

The extracted semantic organization in the embedding space is summarized quantitatively in Table 4.

Table 4. Representative semantic clusters extracted from the embedding space

Cluster Label	Dominant Semantic Category	Representative Lexical Terms	Intra-Cluster Cohesion Score
SC-12	Emotional and Expressive Language	“love”, “grief”, “tears”, “joy”	0.83
SC-27	Philosophical and Abstract Concepts	“truth”, “power”, “will”, “spirit”	0.79
SC-33	Human Communication and Interaction	“speak”, “answer”, “hear”, “call”	0.81
SC-41	Temporal and Sequential References	“before”, “after”, “now”, “then”	0.77
SC-49	Motion and Dynamic Actions	“rise”, “fall”, “stand”, “move”	0.84

The study employed cosine similarity on Word2Vec embeddings, together with ++K-Means clustering to identify words near cluster centers, and semi-automatically generated cluster titles. This manual review approach is often employed in semantic clustering studies. Furthermore, the language was clearer and more uniform in style, and the LSTM model was better able to identify deeper meanings through clustering-based context extraction.

2.4. RNN-LSTM architecture

The proposed text-generating approach is based on a multi-layer RNN-LSTM architecture that includes an embedding input layer, a bidirectional LSTM for sequence learning, a dropout layer for regularization, a dense output layer for prediction, and a context concatenation mechanism. It relies on initial word embeddings derived from Word2Vec representations of Shakespeare's and Nietzsche's works to provide greater contextual flexibility. The context concatenation layer produces better writing across genres by combining the Bi-LSTM outputs with a semantic context vector. The bidirectional LSTM captures data from both directions. It enhances the contextual understanding (Fig. 2).

It is a modest fusion process but it works. This allows the network to consider not just "what is next" in the sequence but also "what is most important". In fact, this tends to yield more accurate, contextually appropriate forecasts. Eq. 1 presents a mathematical expression for this chemical.

$$H' = [H_{\text{LSTM}} \| C_{\text{context}}] \tag{1}$$

where the total feature representation is improved, and the cluster-averaged vector enriches semantics.

The concatenated embeddings are then fed via a dropout regularization layer with a dropout rate of 0.20 to improve generalization and reduce overfitting especially for thematically packed corpora. It randomly drops neurons during training to form robust, transportable patterns.

The proposed architecture terminates with a fully connected dense layer with a softmax activation function, converting the final integrated feature representation into a normalized probability distribution over the vocabulary. This layer computes the probability of each candidate token being the next word in the generated sequence. This prediction process can be mathematically formulated as Eq. 2.

$$P(w_{t+1} \mid \text{ sequence }) = \text{Softmax}(W \cdot H' + c) \tag{2}$$

W denotes the trainable weight matrix, H' represents the fused hidden representation obtained from the recurrent and semantic context integration stages, and c corresponds to the bias vector.

The main architectural and training settings of the proposed framework are summarized in Table 5, including the embedding dimensions, Bi-LSTM structural parameters, the size of the semantic context fusion vector, dropout regularization settings, and the dimensional characteristics of the output prediction layer.

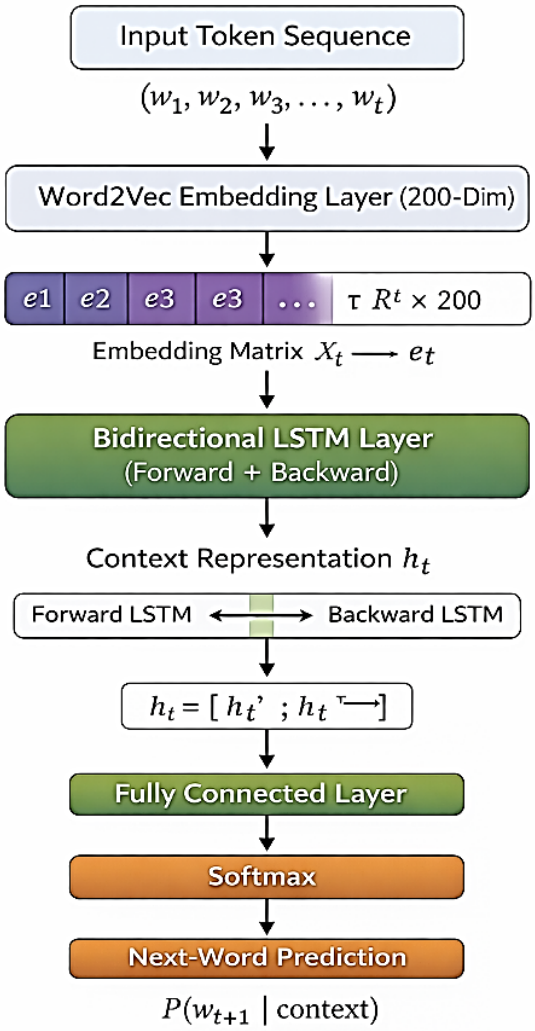

Fig. 2. Proposed multi-layer RNN-LSTM architecture

Table 5. Hyperparameter specification for neural model optimization

Training Component	Assigned Setting
Objective Function	Categorical Cross-Entropy
Optimization Algorithm	Adam Optimizer
Initial Learning Rate	0.001
Mini-Batch Size	128 Samples
Maximum Training Epochs	100 Epochs
Early Stopping Strategy	Enabled
Dropout Regularization Rate	0.20

2.5. Evaluation metrics

Context-sensitive Evaluation Metrics. The design of RNN-LSTM was based on accuracy and category cross-entropy loss. Accuracy measures the model's ability to forecast future tokens. This reflects the model's dependence on semantic context and sequential patterns. On the other hand, minimizing cross-entropy loss leads to better learning, faster convergence, improved generalization, and a holistic evaluation framework.

The predictive capability of the proposed architecture was quantitatively evaluated using classification accuracy, as defined in Eq. 3:

$$\text{ACC} = \frac{1}{M}\sum_{j=1}^{M} \delta(\grave{t}_j, t_j) \tag{3}$$

$\grave{t}_j$ denotes the predicted token at position j, t_j represents the corresponding ground-truth token and $\delta(\cdot)$ is the indicator function that returns 1 when the prediction is correct and 0 otherwise.

To further evaluate the confidence and the uncertainty of the language model, the study used the cross-entropy objective function provided in Eq. 4 and the perplexity metric in Eq. 5. The cross-entropy loss is given by:

$$\mathcal{J} = -\sum_{k=1}^{C} q_k \log(\grave{q}_k) \tag{4}$$

q_k denotes the reference probability distribution for class k, $\grave{q}_k$ indicates the predicted probability distribution and C corresponds to the vocabulary size.

The perplexity score was then obtained through exponential transformation of the loss function according to Eq. 5:

$$\mathrm{PPX} = e^{\mathcal{J}} \tag{5}$$

To evaluate semantic continuity and contextual consistency within the generated sequences, a coherence metric was calculated using adjacent embedding similarity as expressed in Eq. 6:

$$\mathrm{COH} = \frac{1}{T-1}\sum_{r=1}^{T-1} \mathrm{sim}(\mathrm{u}_r, \mathrm{u}_{r+1}) \tag{6}$$

u_r and u_{r+1} denote embedding representations of consecutive words and $\mathrm{sim}(\cdot)$ refers to cosine similarity.

In addition to semantic coherence, stylistic fidelity was quantified using the Style Preservation Measure (SPM) defined in Eq. 7:

$$\mathrm{SPM} = \mathrm{sim}\left(\mathrm{z}_{\mathrm{syn}}, \mathrm{z}_{\mathrm{ref}}\right) \tag{7}$$

$\mathrm{z}_{\mathrm{syn}}$ represents the embedding vector of the synthesized text and $\mathrm{z}_{\mathrm{ref}}$ corresponds to the reference authorial representation.

Lexical overlap and semantic similarity were assessed between generated and reference texts using common Natural Language Generation (NLG) metrics such as BLEU, ROUGE-L and METEOR. The BLEU formulation is given in Eq. 8:

$$\mathrm{BLEU} = \gamma \cdot \exp\left(\sum_{m=1}^{4} \alpha_m \ln s_m\right) \tag{8}$$

γ denotes the brevity penalty, α_m represents the weighting coefficient, and s_m indicates modified n-gram precision.

The ROUGE-L metric was computed according to Eq. 9:

$$\mathrm{ROUGE} - L = \frac{\mathrm{LCS}(A,B)}{|A|} \tag{9}$$

$\mathrm{LCS}(A, B)$ is the longest common subsequence between the generated sequence A and the reference sequence B.

Similarly, the METEOR score was calculated using Eq. 10:

$$\mathrm{METEOR} = \Phi \times (1 - \lambda) \tag{10}$$

Φ denotes the harmonic alignment score and λ corresponds to the fragmentation penalty factor.

3. Results and Discussion

The results indicate that as the proportion of training data increases, the proposed architecture improves prediction accuracy and reduces loss and confusion values. On the full dataset, the model achieved 67.4% accuracy on the Nietzsche corpus, 61.3% on the Shakespeare corpus, and 63.1% on the WikiText-2 dataset. Across all datasets, performance improved from 25% to 100% training consumption, demonstrating good scalability of learning without early saturation. The Nietzsche corpus shows the strongest prediction performance due to the structural coherence of the writing style. Shakespeare's works have a lengthier convergence time due to the grammatical complexity and archaic language. Meanwhile, the competitive performance on WikiText-2 suggests that the proposed framework can be used to solve a range of problems beyond the generation of literary texts (Fig. 3).

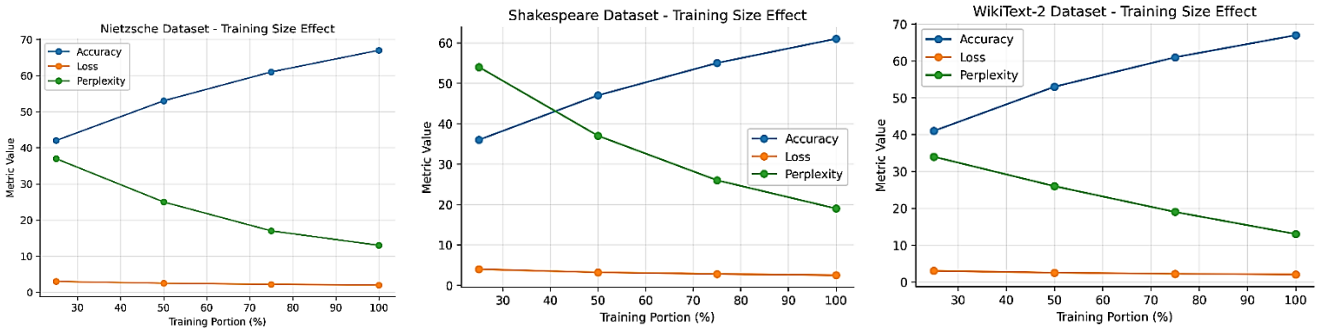

Fig. 3. Prediction accuracy and loss behavior across progressive training data utilization levels

Fig. 4 illustrates the robustness analysis of the proposed design under various noise situations. As the injected noise level increased from 10% to 30%, the prediction accuracy gradually decreased, and perplexity values increased across all investigated datasets. The results reveal the detrimental effect of noise on the model's capacity to maintain semantic consistency and to predict sequences. At the highest perturbation level (30%), the accuracy loss for the Shakespeare corpus was around 13.5%, and for the Nietzsche dataset, somewhat greater at 14.8%. The suggested framework shows reasonably consistent predictive behaviour and therefore robustness in noisy textual conditions, despite the observed performance drops.

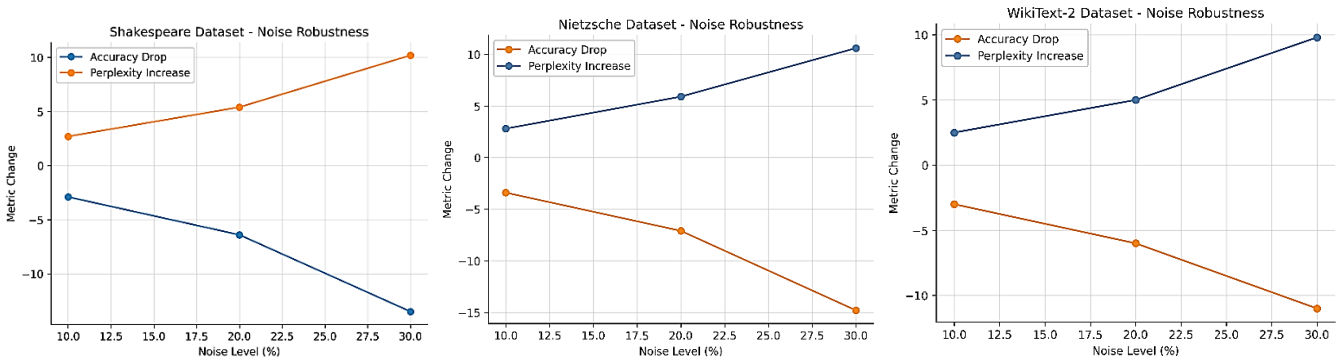

Fig. 4. Performance degradation under progressive noise perturbation levels

Fig. 5 shows the effect of embedding dimensionality on the predictive performance of the proposed context-aware RNN-LSTM framework across the datasets under study. The experiments demonstrate that the prediction accuracy across all corpora improves as the embedding dimension increases from 50 to 300. Higher-dimensional embeddings are more effective at capturing semantic information because they encode more lexical and contextual information in the vector space. Ultimately, this enhances the model's ability to learn complex semantic linkages, contextual associations, and long-range word dependencies.

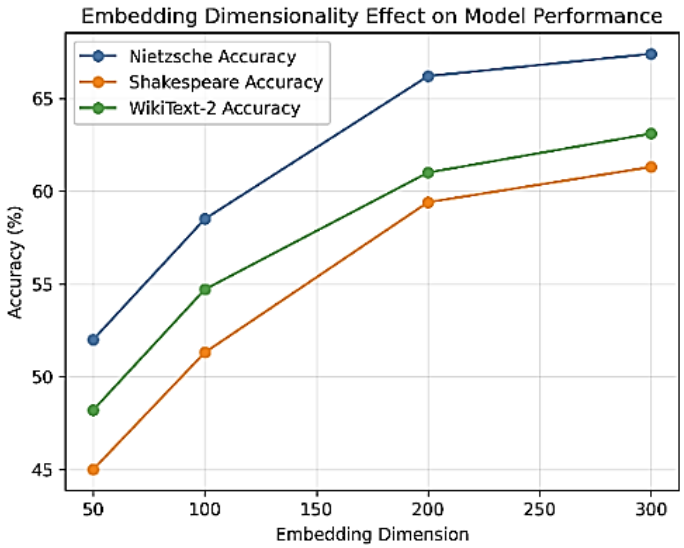

Fig. 5. Effect of embedding vector dimension on semantic learning performance

The speed improvements are particularly obvious when going from micro to medium-sized embeddings, e.g., up to 200 dimensions. However, the rate of performance growth immediately diminishes at this point, indicating that the main semantic features of the datasets are already well represented by the medium-dimensional embedding spaces. This suggests that large embedding dimensions may contribute to the computational complexity without improving the quality of language modeling

Among the tested corpora, the best prediction quality was achieved in the Nietzsche dataset. This is owing to the structured semantic patterns, philosophical consistency, and relatively consistent writing style visible in Nietzsche's writings, which encourage more steady contextual learning. Conversely, the Shakespeare corpus has somewhat poor accuracy due to its very passionate literary style, difficult grammatical structures, and antiquated language. These linguistic features also make sequential modeling and next word prediction more challenging.

Despite these challenges, particularly the diversity of general-domain material in the WikiText-2 corpus, the suggested architecture achieved competitive results across all datasets. The experimental results reveal that larger embedding representations can significantly improve semantic understanding and contextual learning within the proposed clustering-enhanced RNN–LSTM text generation paradigm.

Table 6 presents the experimental results of the proposed context-aware RNN-LSTM framework, demonstrating consistent linguistic quality across all evaluation datasets. Experiments show that the model effectively preserves contextual coherence, grammatical consistency, and semantic continuity in text generation across both literary and general-domain corpora. The Friedrich Nietzsche corpus received the highest ratings for both local and global coherence among the datasets investigated. This enhanced performance is mainly due to the greater semantic consistency of Nietzsche's writings and the more orderly style of philosophical writing. The repetition of design, together with the repetition of theme patterns and logically connected phrase structures, helps the model learn the context more successfully, enabling it to keep the consistency of long-range semantic linkages in the output sequences. In contrast, the coherence scores for the Shakespeare data set were far worse. Shakespearean literature is characterized by syntactically complex phrase patterns, highly emotive dramatic exchanges, and archaic vocabulary, which lead to inferior performance. These linguistic features also impose additional difficulties on the neural language generation process by impeding sequence prediction and context modeling. Even as the complexity of the literary corpora increased, the proposed architecture maintained constant performance on the WikiText-2 dataset, which contains a variety of general-domain textual content from several knowledge

domains. The sequences produced demonstrated sufficient semantic coherence and grammatical consistency across a range of topics and writing styles. The above results, with grammar evaluation exceeding 84%, further indicate the effectiveness of the recommended semantic context integration technique in preserving the linguistic stability and syntactic correctness of the final text across a wide range of textual domains.

Table 6. Linguistic coherence and syntactic quality assessment of the generated text

Evaluation Criterion	Nietzsche Corpus	Shakespeare Corpus	WikiText-2 Corpus
Sentence-Level Coherence	0.83	0.79	0.81
Document-Level Semantic Consistency	0.71	0.67	0.69
Grammatical Accuracy Rate	86.2%	82.4%	84.7%

The Style Preservation Index (SPI) evaluates the degree of preservation of the original author's writing style in the generated text. Shakespeare scored approximately 0.66, and Nietzsche came in at 0.71. These results indicate that many of the author's stylistic traits are preserved in the context-aware model. But the machine is less skilled at Shakespearean works, which feature more complex stylistic traits, such as iambic meter and archaic language. Semantic error analysis shows that the most common faults in the model are semantic substitution errors (33%) and wrong verb tenses (19%). These issues usually stem from long-range dependencies, which RNN-based models are often unable to capture accurately. The rare word misprediction (11%) implies a problem with vocabulary sparsity, notably in the Shakespeare corpus. Fig. 6 demonstrates that the least common error is context drift (7%). However, it suggests that models may be unable to preserve topic continuity across longer text sequences.

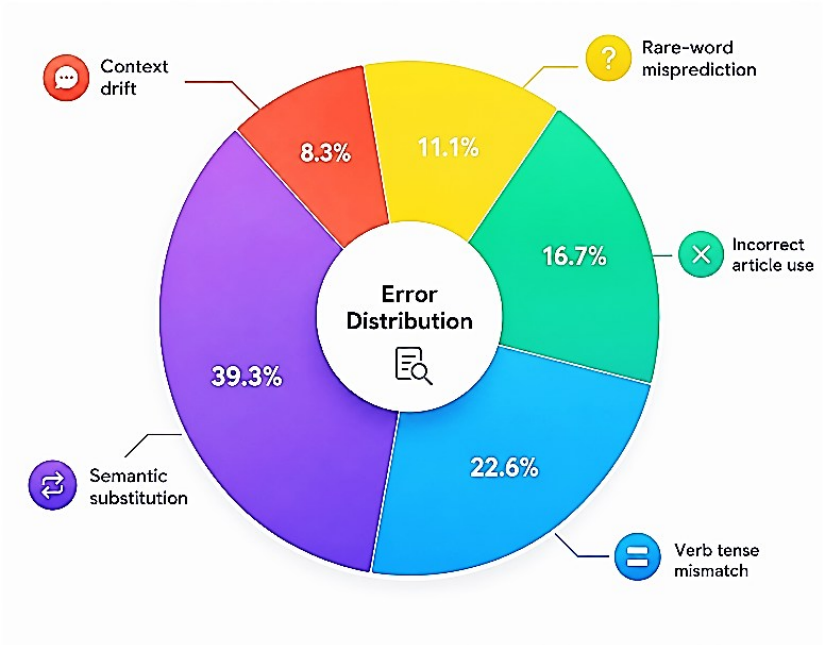

Fig. 6. Analysis of semantic drift verb inconsistency and lexical prediction errors

Table 7 shows the average similarity between the generated text and reference texts throughout the evaluated datasets. The scores are a bit lower for the Shakespeare dataset. Shakespeare's language is more difficult to predict due to his complex dialogue, antiquated vocabulary, and unusual phrasing. Average findings on the WikiText-2 dataset. This indicates that the context-aware RNN-LSTM model can still exhibit considerable lexical and semantic similarity when used in larger general text datasets. Finally, the results suggest that combining semantic clustering with sequential modeling improves the quality of the generated text and maintains consistency of context across different text types.

Table 7. Comparative evaluation of text generation quality across benchmark datasets

Evaluation Metric	Nietzsche Corpus	Shakespeare Corpus	WikiText-2 Corpus
BLEU Score	0.42	0.39	0.41
ROUGE-L Similarity	0.57	0.54	0.56
METEOR Performance	0.29	0.26	0.28

The findings of ++K-Means clustering in Fig. 7 show clearly separated semantic regions in the trained Word2Vec embedding space with cluster centroids in discrete conceptual domains. The scatter projection based on PCA reveals that the embedding training method successfully learned significant lexical and semantic links and semantically related words formed cohesive cluster structures. Strong coherence within clusters, as reflected in dense agglomeration around numerous centroids, especially for dominant literary themes such as emotion, abstract reasoning, and interpersonal interaction, is crucial for reliable semantic context extraction. These findings confirm the fundamental notion that literary corpora abound with identifiable semantic markers that can be fruitfully exploited to improve the performance of next-word prediction and contextual language modeling.

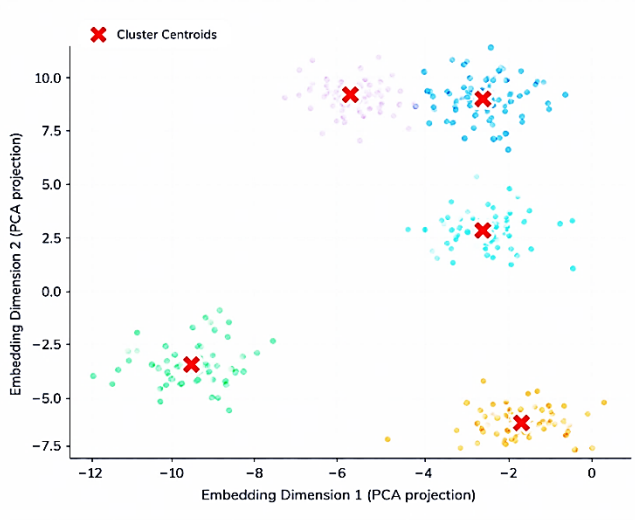

Fig. 7. PCA scatter distribution of semantically coherent word clusters

This interpretation is further reinforced by the cluster distance matrix in Fig. 8, which shows inter-cluster interactions in the embedding space. In the Nietzsche corpus, the clusters of philosophical reasoning are fairly close to the clusters of abstract conceptual words. In the Shakespeare-related clusters, those associated with action verbs, conversation markers, and archaic idioms are closer together.

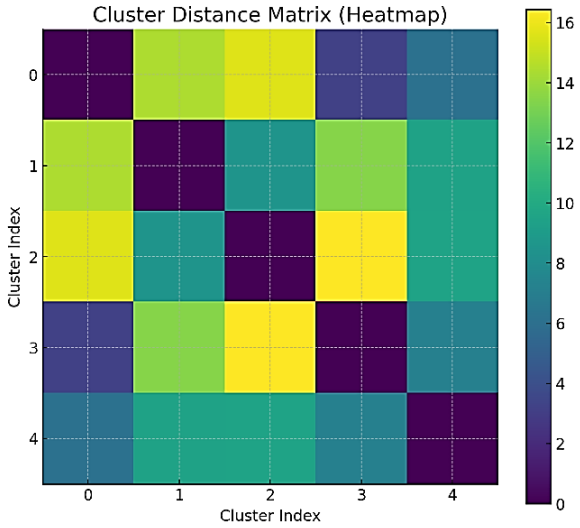

Fig. 8. Distance-based relationship mapping of semantic clusters

Table 8 proposes that the model may achieve better performance than the conventional and neural baseline models on all datasets. Neural structures introduce sequential dependencies, which significantly improve performance, whereas standard statistical models, e.g., the 5-gram technique, exhibit high confusion and low prediction accuracy.

Table 8. Comparative performance evaluation between baseline architectures and the proposed semantic-aware framework

Architecture Type	Nietzsche (Acc / PPL)	Shakespeare (Acc / PPL)	WikiText-2 (Acc / PPL)
Statistical 5-Gram Model	24.5% / 210.4	21.7% / 243.5	23.6% / 198.7
Conventional LSTM	49.1% / 44.3	43.5% / 57.9	46.8% / 48.5
GRU Sequential Network	54.6% / 36.8	48.2% / 49.3	51.3% / 41.7
Compact Transformer Architecture	61.9% / 24.7	56.1% / 36.4	58.7% / 29.8
Proposed Context-Aware Semantic RNN–LSTM	67.4% / 21.5	61.3% / 32.8	63.1% / 26.4

The Transformer-Small model surpassed baseline neural language production models thanks to its self-attention mechanism, which effectively captures contextual dependencies and long-range word connections in textual sequences. This transformer design enables efficient description of semantic relationships. It dynamically assigns attention weights to relevant tokens. However, the proposed context-enhanced strategy consistently improves the overall performance, especially in prediction accuracy, semantic consistency, and reduced contextual ambiguity during text production. The study reports continuous improvements across both literary and general-purpose corpora, including Nietzsche, Shakespeare, and WikiText-2. These results show that the proposed methodology can capture higher-level semantic structures, theme continuity, and sequential language linkages. The design combines semantic context vectors derived from clustering and sequential learning with a Bi-LSTM to preserve contextual coherence across a range of writing styles and language domains. Moreover, to study the individual contribution of each architectural component, the study performs a thorough ablation analysis. The purpose of this experiment is to evaluate

and quantify the influence of the main components of the proposed architecture, including the distributed embedding representation technique, the Bi-LSTM sequential modelling layer, and the clustering-based semantic context extraction mechanism. This can be done by gradually deleting or simplifying individual parts. The relative value of each module can be assessed in a methodical manner

In the ablation experiment four different architectural configurations have been explored. The first setting is a simple Bi-LSTM architecture, trained only on sequential token input with no additional semantic clusters. The second one is an enhancement of lexical-unit semantic encoding using distributed embedding representations. The third configuration includes the semantic context vectors from clustering but not the full semantic fusion technique used in the final architecture. The final configuration is the entire suggested system, combining Bi-LSTM sequential learning, clustering-based semantic context fusion, and Word2Vec embeddings into a unified text generation model.

Table 9 shows the contribution of each architectural module to the system's overall predictive performance. The results confirm the significance of clustering-based representation learning and semantic context fusion in improving contextual consistency, semantic understanding, and sequence prediction accuracy of the proposed language generation system. The comparative analysis indicates that the Transformer-Small architecture performs well among the baseline neural language production models, thanks to its self-attention mechanism, which learns contextual dependencies and long-range word interactions within textual sequences. The transformer architecture efficiently encodes semantic relationships by dynamically allocating attention weights to relevant tokens. However, the proposed context-enhanced strategy consistently improves the overall performance, especially in prediction accuracy, semantic consistency, and reduced contextual ambiguity during text production.

Table 9. Contribution analysis of sequential and semantic components in the proposed framework

Experimental Architecture	Nietzsche Accuracy	Shakespeare Accuracy	WikiText-2 Accuracy
Standalone Bi-LSTM Architecture	52.1%	46.3%	49.2%
Bi-LSTM Enhanced with Word2Vec Representation	57.8%	50.2%	54.6%
Bi-LSTM Integrated with Semantic Cluster Features	61.2%	55.0%	58.4%
Complete Context-Aware Semantic RNN–LSTM Model	67.4%	61.3%	63.1%

The study continues to improve on both literary and general-purpose corpora, including Nietzsche, Shakespeare, and WikiText-2. These results show that the suggested methodology can capture higher-level semantic structures, theme continuity, and sequential language linkages. The design combines semantic context vectors derived from clustering and sequential learning with a Bi-LSTM to preserve contextual coherence across a range of writing styles and language domains. Moreover, to study the individual contribution of each architectural component, the study performs a thorough ablation analysis. The purpose of this experiment is to evaluate and quantify the influence of the main components of the proposed architecture, including the distributed embedding representation technique, the Bi-LSTM sequential modelling layer, and the clustering-based semantic context extraction mechanism. This can be done by gradually deleting or simplifying individual parts. The relative value of each module can be assessed in a methodical manner

In the ablation experiment four different architectural configurations have been explored. The first setting is a simple BiLSTM architecture, trained only on sequential token input with no additional semantic clusters. The second one is an enhancement of lexical-unit semantic encoding using distributed embedding representations. The third configuration includes the semantic context vectors from clustering but not the full semantic fusion technique used in the final architecture. The final configuration is the full proposed system, which integrates BiLSTM sequential learning, clustering-based semantic context fusion, and Word2Vec embeddings into a single text generation model.

4. Conclusions

This study proposed a novel neural language generation approach by integrating bidirectional recurrent sequence learning with clustering-based semantic augmentation to enhance contextual dependency learning and next-token prediction performance. The proposed method employed distributed lexical representations, semantic grouping strategies, and deep recurrent modeling to jointly model higher-level semantic compositions within text sequences and short-term syntactic relations. Experiments on the Nietzsche, Shakespeare, and WikiText-2 corpora showed that the proposed approach outperforms many traditional benchmark methods, including probabilistic n-grams, classical LSTMs, GRUs, and small transformer models. The results showed that the generated texts had higher semantic continuity and lower perplexity scores, with prediction accuracies of 67.4%, 61.3%, and 63.1%, respectively. Moreover, the results indicated that semantic cluster integration can significantly enhance contextual interpretation, theme coherence, and the preservation of writing style in literary and open-domain text generation tasks. In particular, the framework maintained more cohesive language transitions and improved the overall quality of the sequence by using semantic-guiding vectors. Also, further demonstrated, through ablation testing, that the integration of semantic context is a major driver of the observed performance gains. These results are promising, but there are several limitations in handling very rare vocabulary items and maintaining long contextual dependencies throughout generated sequences. Future work should include effective attention-based or transformer-based semantic augmentation modules to improve scalability, contextual sensitivity, and stylistic adaptability in sophisticated neural text generation systems.

Acknowledgment

The author appreciates the editor and the anonymous reviewers for insightful comments and helpful suggestions. Their recommendations were very useful in improving and strengthening the manuscript. This research received no external funding and was conducted without financial support from any public, commercial or non-profit funding agency. Also, the author states that there is no conflict of interest regarding the publishing of this paper.

References

[1] W. H. Bisen and A. J. Agrawal, "Review on Natural Language Generation," Int. J. Health Sci., pp. 10365–10376, May 2022, https://doi.org/10.53730/ijhs.v6nS1.7489.

[2] R. Arabelli, S. Gupta, N. Prakash, and Z. Ali, "Natural Language Generation in AI: Developing Human-Like Text Through Deep Learning," in 2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT), Bhimtal, Nainital, India: IEEE, Feb. 2025, pp. 1411–1415 https://doi.org/10.1109/CE2CT64011.2025.10939615.

[3] D. Shan, K. Yao, and X. Zhang, "Sequential Learning Network with Residual Blocks: Incorporating Temporal Convolutional Information into Recurrent Neural Networks," IEEE Trans. Cogn. Dev. Syst., vol. 16, no. 1, pp. 396–401, Feb. 2024, https://doi.org/10.1109/TCDS.2023.3325358.

[4] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, and A. Muneer, "LSTM Inefficiency in Long-Term Dependencies Regression Problems," J. Adv. Res. Appl. Sci. Eng. Technol., vol. 30, no. 3, pp. 16–31, May 2023, https://doi.org/10.37934/araset.30.3.1631.

[5] M. Kanmani, S. H S, A. Mergin, I. T. Joseph S, and V. V, "Enhancing Sentence Prediction through Bidirectional Long Short-Term Memory Networks," Int. J. Electron. Commun. Eng., vol. 13, no. 3, pp. 292–300, Mar. 2026, https://doi.org/10.14445/23488549/IJECE-V13I3P123.

[6] I. van Heerden and A. Bas, "A Perspective on Literary Metaphor in the Context of Generative AI," 2024. [Online]. Available: https://arxiv.org/pdf/2409.01053

[7] L. Pathak, K. Lochab, and V. Gidwani, "Character-Level Text Generation for Shakespearean Style with LSTMs," Int. J. Innov. Sci. Res. Technol., vol. X, no. Y, pp. 1425–1431, Sep. 2024, https://doi.org/110.38124/ijisrt/IJISRT24AUG1043.

[8] M. Th, S. Sahu, and A. Anand, "Evaluating distributed word representations for capturing semantics of biomedical concepts," in Proceedings of BioNLP 15, Beijing, China: Association for Computational Linguistics, 2015, pp. 158–163. https://doi.org/10.18653/v1/W15-3820.

[9] [9] C. Zhang *et al*., "From Word Vectors to Multimodal Embeddings: Techniques, Applications, and Future Directions for Large Language Models," *arXiv preprint* arXiv:2411.05036, 2024, https://doi.org/110.48550/ARXIV.2411.05036.

[10] R. Choudhary, O. Alsayed, S. Doboli, and A. A. Minai, "Building Semantic Cognitive Maps with Text Embedding and Clustering," in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy: IEEE, Jul. 2022, pp. 01–08. https://doi.org/10.1109/IJCNN55064.2022.9892429.

[11] S. Jung, "Semantic Vector Learning and Visualization with Semantic Cluster Using Transformers in Natural Language Understanding," J. Comput. Sci. Eng., vol. 16, no. 2, pp. 63–78, Jun. 2022, https://doi.org/10.5626/JCSE.2022.16.2.63.

[12] F. Viegas, L. Rocha, and M. A. Gonçalves, "On the Role of Semantic Word Clusters — CluWords — in Natural Language Processing (NLP) Tasks," in Anais do XXXVII Concurso de Teses e Dissertações (CTD 2024), Brasil: Sociedade Brasileira de Computação - SBC, Jul. 2024, pp. 38–47. https://doi.org/10.5753/ctd.2024.2036.

[13] S. Jung and S. Lim, "Cluster-aware Semantic Vector Learning Using BERT in Natural Language Understanding," in 2021 IEEE Int. Conf. Big Data Smart Comput. (BigComp), Jeju Island, South Korea, Jan. 2021, pp. 91–98, https://doi.org/10.1109/BigComp51126.2021.00026.

[14] Q. Guo, X. Qiu, X. Xue, and Z. Zhang, "Low-Rank and Locality Constrained Self-Attention for Sequence Modeling," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 12, pp. 2213–2222, Dec. 2019, doi: 10.1109/TASLP.2019.2944078. https://doi.org/10.1109/TASLP.2019.2944078.

[15] E. Shirazi and A. H. Ardakani, "How Much Attention to Pay? Attention-Enhanced Sequential Learning Models," in 2025 IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT Europe), Valletta, Malta, Oct. 2025, pp. 1–5, https://doi.org/10.1109/ISGTEurope64741.2025.11305453.

[16] J. Armengol-Estapé and M. R. Costa-Jussà, "Semantic and syntactic information for neural machine translation: Injecting Features to the Transformer," Mach. Transl., vol. 35, no. 1, pp. 3–17, Apr. 2021, https://doi.org/10.1007/s10590-021-09264-2.

[17] R. Katrix, Q. Carroway, R. Hawkesbury, and M. Heathfield, "Context-Aware Semantic Recomposition Mechanism for Large Language Models," arXiv:2501.17386, 2025, https://doi.org/10.48550/ARXIV.2501.17386.

[18] J. K. Miller and T. J. Alexander, "Human-interpretable clustering of short text using large language models," R. Soc. Open Sci., vol. 12, no. 1, Art. no. 241692, 2025, https://doi.org/10.1098/rsos.241692.

[19] N. Fulda, "You Are What You Read: The Effect of Corpus and Training Task on Semantic Absorption in Recurrent Neural Architectures," in 2020 IEEE 18th World Symp. Appl. Mach. Intell. Informat. (SAMI), Herlany, Slovakia, Jan. 2020, pp. 201–206, https://doi.org/10.1109/SAMI48414.2020.9108757.

[20] P. Le and W. Zuidema, "Quantifying the Vanishing Gradient and Long Distance Dependency Problem in Recursive Neural Networks and Recursive LSTMs," in Proc. 1st Workshop Represent. Learn. NLP, Berlin, Germany, 2016, pp. 87–93, https://doi.org/10.18653/v1/W16-1610.

[21] H. Okut, "Deep Learning for Subtyping and Prediction of Diseases: Long-Short Term Memory," in Deep Learning Applications, P. L. Mazzeo and P. Spagnolo, Eds. London, U.K.: IntechOpen, 2021, https://doi.org/10.5772/intechopen.96180.

[22] Q. U. Ain, S. U. Nisa, Aamana, M. Hilal, H. Kabeer, and F. Subhan, "Bidirectional LSTM for Context-Rich Abstractive Summarization: A Step Beyond Sequence-to-Sequence and Applied to Speech Impaired Transcriptions," in 2025 IEEE 22nd Int. Conf. Smart Communities: Improving Quality of Life Using AI, Robotics and IoT (HONET), Topi, Pakistan, Dec. 2025, pp. 92–97, https://doi.org/10.1109/HONET67928.2025.11318471.

[23] V. Hofmann, J. Pierrehumbert, and H. Schütze, "Dynamic Contextualized Word Embeddings," in Proc. 59th Annu. Meeting Assoc. Comput. Linguistics and 11th Int. Joint Conf. Natural Language Processing (Vol. 1), Online, 2021, pp. 6970–6984, https://doi.org/10.18653/v1/2021.acl-long.542.

[24] M. Apidianaki, "From Word Types to Tokens and Back: A Survey of Approaches to Word Meaning Representation and Interpretation," Comput. Linguist., vol. 49, no. 2, pp. 1–59, Mar. 2023, https://doi.org/10.1162/coli_a_00474.

[25] P. Tsvilodub, R. D. Hawkins, and M. Franke, "Integrating Neural and Symbolic Components in a Model of Pragmatic Question-Answering," arXiv:2506.01474, 2025, https://doi.org/10.48550/ARXIV.2506.01474.

[26] G. Neubig and C. Dyer, "Generalizing and Hybridizing Count-based and Neural Language Models," in Proc. 2016 Conf. Empirical Methods Natural Language Process. (EMNLP), Austin, TX, USA, 2016, pp. 1163–1172, https://doi.org/10.18653/v1/D16-1124.

[27] J. Björklund, A. Dahlgren Lindström, and F. Drewes, "Bridging Perception, Memory, and Inference through Semantic Relations," in Proc.

2021 Conf. Empirical Methods Natural Language Process. (EMNLP), Online and Punta Cana, Dominican Republic, 2021, pp. 9136–9142, https://doi.org/10.18653/v1/2021.emnlp-main.719.

[28] Supriyono, A. P. Wibawa, Suyono, and F. Kurniawan, "A survey of text summarization: Techniques, evaluation and challenges," Nat. Lang. Process. J., vol. 7, Art. no. 100070, 2024, https://doi.org/10.1016/j.nlp.2024.100070.

[29] P. Contreras Kallens and M. H. Christiansen, "Models of Language and Multiword Expressions," Front. Artif. Intell., vol. 5, Art. no. 781962, 2022, https://doi.org/10.3389/frai.2022.781962.

[30] A. Thielmann, C. Weisser, T. Kneib, and B. Säfken, "Coherence-Based Document Clustering," in 2023 IEEE 17th Int. Conf. Semantic Comput. (ICSC), Laguna Hills, CA, USA, Feb. 2023, pp. 9–16, https://doi.org/10.1109/ICSC56153.2023.00009.

[31] J. Wood, B. Li, J. Lee, C. Arnold, and W. Wang, "On the Utility of Combining Topic Models and Recurrent Neural Networks," in Recent Advances in Information and Communication Technology 2021, P. Meesad, S. Sodsee, W. Jitsakul, and S. Tangwannawit, Eds. Cham, Switzerland: Springer, 2021, pp. 66–76, https://doi.org/10.1007/978-3-030-79757-7_7.

[32] L. George and P. Sumathy, "An integrated clustering and BERT framework for improved topic modeling," Int. J. Inf. Technol., vol. 15, no. 4, pp. 2187–2195, Apr. 2023, https://doi.org/10.1007/s41870-023-01268-w.

[33] Kris. (2018, Aug. 30). *Nietzsche texts* [Online]. Available: https://www.kaggle.com/datasets/pankrzysiu/nietzsche-texts

[34] L. Larsen. (2024, Jan. 15). *Shakespeare plays* [Online]. Available: https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays/data

[35] V. Mettu. (2021, Jul. 12). *WikiText-2 data* [Online]. Available: https://www.kaggle.com/datasets/vivekmettu/wikitext2-data