
An Intelligent Domain-Specific Content Categorization Framework Using a Deep Learning-Based Hybrid Approach

V.K. Janani Sivapriya^{1,2}, R. Saravana Moorthy², S. Angel³

¹Department of Data Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India

²Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, India

³Department of Cybersecurity with CS, PSGR Krishnammal College for Women, Coimbatore, India

Abstract.

Automatic content classification is becoming increasingly important as the volume of multi-domain digital information produced by business, healthcare, education, sports, and technology grows exponentially. Traditional text categorisation methods are unable to capture semantic relations within domain-specific documents, resulting in low classification performance when applied across domains. To solve this problem, this paper proposes an intelligent multi-domain content categorization system. First, Multi-Level Contextual Text Refinement Filtering (MCTRF) performs denoising, text normalisation, domain-term preservation, and context-consistency enhancement to obtain a high-quality text representation through a comprehensive text refinement process. Thereafter, a Graph Attention Network (GAT) is used to capture and learn the semantic correlations among domain-specific concepts by assigning different attention weights to neighbouring nodes, thereby yielding informative domain-relationship features. Lastly, a hybrid Convolutional Neural Network with a Bidirectional Long Short-Term Memory (CNN-BiLSTM) classifier is applied to capture both local semantic patterns and long-term contextual information for precise content classification. The evaluation is performed on a public multi-domain text dataset using the parameters are accuracy, precision, recall, and F1-score. It is anticipated that the presented framework will provide better results than traditional approaches, and deliver a computationally efficient, scalable approach for multi-domain document categorisation.

Keywords. Deep Learning (DL), Domain based content categorisation, relationship, hybrid approach, local features, global features.

1. INTRODUCTION

The boom in digital text across business, healthcare, education, sports, entertainment, and technology presents the challenge of developing smart, domain-specific content classification systems. Automatic domain label extraction is an important task for information retrieval, recommender systems, knowledge management, and decision support, facilitating the organisation and analysis of large-scale textual data (Yuyuan Zhang et al., 2025). Conventionally, domain classification is based on expert manual annotation, a process in which a set of predefined labels is assigned to a document based on its semantics. However, such a process is time-consuming, labour-intensive, costly and inconsistent, and

thus it is not feasible for large-scale and real-time applications. Therefore, automatic domain classification is an efficient way to handle diverse textual data. Recent progress in DL has led to dramatic improvements in the performance of automatic text classification by learning hierarchical semantic representations from raw document input. DL-based document classification methods also effectively capture sequential context and have been shown to outperform traditional machine learning methods for document classification in various domains such as Technology, Business, Sports, Entertainment, and Politics D. Tarun et al., (2023).

However, current DL techniques are prone to producing redundant document representations, failing to exploit the semantic relations within domain knowledge fully, and struggling to model both local and global contextual patterns. These constraints degrade the ability to classify domain-specific data accurately. To overcome these challenges, we introduce an intelligent domain-specific content categorisation system based on a hybrid DL model in this paper. The framework utilises MCTRF for contextual denoising, lexical normalisation, and the retention of domain-specific keywords, and then employs a GAT model to capture semantic relations among domain terms. Finally, a Hybrid CNN–BiLSTM classifier is proposed to combine local feature extraction with bidirectional contextual learning to achieve a more accurate and robust multi-domain content classifier. The primary contributions of this study are as follows:

- A novel MCTRF framework is proposed for textual denoising, lexical normalisation, contextual refinement and domain-specific keyword preservation.
- To produce relation-aware feature representations, a GAT-based semantic learning model is proposed that adaptively attends to different parts of the input data.
- A Hybrid CNN–BiLSTM classifier is proposed to simultaneously capture local semantic features and long-range contextual dependency for effective multi-domain text classification.
- The proposed framework is tested on a public multi-domain text dataset and compared in terms of accuracy, precision, recall, and F1 Score.

2. LITERATURE SURVEY

The research in L. Liu *et al.*, (2024) proposed an Improved Text Label Optimization (ITLO) framework for Specific Domain Prompt Learning (SDPL) to optimize prompt representations for domain-specific text classification. The research in D. K. Meghana *et al.*, (2025) introduced an Auto-Encoder-Based Structural Correspondence Learning with Pivot Augmentation (AESCL-PA) model for multilingual text classification. The work in Emanuel, R. H *et al.*, (2024) evaluated different activation functions in Convolutional Neural Network (CNN) models for text classification. The approach in Sun, Z *et al.*, (2024) developed a Digital Twin with Unsupervised Domain Adaptive Learning (DT-UDAL) framework for fault diagnosis using Proton Exchange Membrane Fuel Cell (PEMFC) systems. The methodology in Zhu, J *et al.*, (2024) proposed Causal Prototype-Inspired Contrast Adaptation (CPCA) for Unsupervised Domain Adaptive Semantic Segmentation (UDASS). The model reduced domain discrepancies using contrastive adaptation and achieved superior segmentation accuracy and cross-domain generalization. The study in Nair, A. R *et al.*, (2024) investigated text data augmentation techniques using DistilBidirectional Encoder Representations from Transformers (DistilBERT) for text classification.

The framework in Yin, K *et al.*, (2026) introduced CrisisSense-Large Language Model (CrisisSense-LLM) for multi-label disaster-related social media text classification. The instruction fine-tuned model effectively captured contextual semantics and achieved high precision, recall, and F1-score. The research in Sy, C. Y *et al.*, (2024) compared Robustly Optimized Bidirectional Encoder Representations from Transformers Approach (RoBERTa) for supervised text classification. The method in Zheng, T *et al.*, (2024) proposed Character Distance Network (CDistNet) for robust multi-domain text recognition. The framework learned character-distance relationships across domains and improved recognition accuracy under diverse conditions. The study in I. Bagwan *et al.*, (2026) presented Domain-Adaptive Transfer Learning (DATL) for scam call detection. The proposed model enhanced transferability across domains while maintaining privacy and achieved higher detection accuracy.

3. PROPOSED METHODOLOGY

Our proposed framework operates on a domain-based dataset of Business, Technology, Sports, Entertainment, and Education, as presented in figure 1. The MCTRF module first preprocesses the documents by dropping URLs, punctuation, and other noise, and then performs stop word removal and lemmatisation to obtain a polished textual representation.

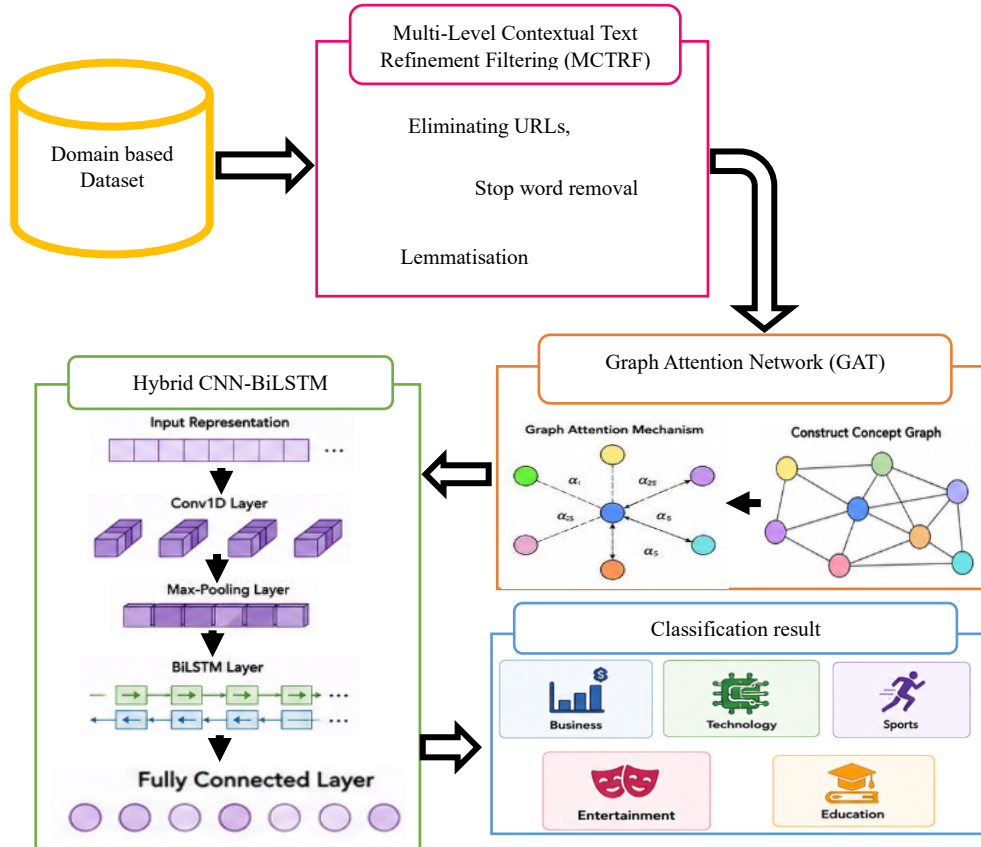


Figure .1. Architecture diagram

The refined documents are then passed to the GAT to establish semantic relations between domain-related keywords by assigning attention weights to neighbouring concepts. As a result, they produce a relation-aware feature representation that encodes both local and contextual semantic information. The relationship-aware features, in turn, are fed into the Hybrid CNN–BiLSTM classifier, where the CNN learns local semantic features and the BiLSTM learns bidirectional contextual relations. The final SoftMax layer predicts the domain category.

3.1 Multi-Level Contextual Text Refinement Filtering (MCTRF)

The presented MCTRF approach enhances the semantic quality of the multi-domain news dataset, including articles on Business, Technology, Sports, Education, and Entertainment. Each news item has a headline, a description, and the article content, which are initially combined into a single document to preserve the full context. The integrated document is represented as equation 3.1.

$$T_i = h_i \oplus s_i \oplus c_i \quad (3.1)$$

Here, T_i refers to integrated textual document, h_i lies the headline, s_i express the description, c_i denotes the article content, and $\oplus$ denotes Concatenation operator. The merged document is then preprocessed by eliminating URLs, punctuation, special characters, numbers, and other irrelevant content to denoise the data and enhance semantic uniformity. The refinement process is given by equation 3.2.

$$T_i^r = \phi(T_i) \quad (3.2)$$

Where, T_i^r refined document and $\phi(\cdot)$ is the Noise removal function. Then the processed text is tokenised into a sequence of words, and the words are lowercased to reduce lexical discrepancies across documents.

$$W_i = \{w_1, w_2, \dots, w_m\} \quad (3.3)$$

Herein, W_i is the token sequence, w_j individual token, and m refers to number of tokens. The normalised tokens are then tokenised on whitespace to obtain a semantically meaningful sequence of tokens, after eliminating domain-irrelevant stop words.

$$W_i^s = W_i - S \quad (3.4)$$

Let us assume, W_i^s is the filtered token sequence, S is the stop word vocabulary. Following lemmatisation, several inflected forms are mapped to their canonical root forms, which helps reduce the effect of vocabulary sparsity while preserving the implicit meaning.

$$L_i = \psi(W_i^s) \quad (3.5)$$

Where, L_i is the Lemmatized token sequence, and $\psi(\cdot)$ lies the Lemmatization function. Lastly, keep domain-related keywords to strengthen and preserve the semantic representation for better domain discrimination before performing graph-based relationship learning.

$$R_i = L_i \cup K \quad (3.6)$$

where R_i denotes the refined document, K is the domain specific keyword set, and $\cup$ is the union operator. The resultant refined document R_i , which contains semantically enriched domain-specific information after redundancy removal, is then sent to the GAT for learning semantic relationships.

3.2 Graph Attention Network (GAT)

After preprocessing, our proposed GAT captures the semantic correlations among domain-specific concepts by mapping each processed document to a semantic graph, in which keywords serve as nodes, and their contextual co-occurrences define edges. Each

node is initialised with a contextual feature vector and projected into a learnable latent space. The graph attention network now computes adaptive attention coefficients to weight information from adjacent nodes based on semantic relevance. Finally, the new node embeddings are integrated to form a relation-aware feature representation that captures not only local semantic interactions but also global contextual dependencies.

$$G = (V, E) \quad (3.7)$$

Here, G is the semantic graph, $V = \{v_1, v_2, \dots, v_n\}$ is the set of graph nodes representing domain specific keywords, $E \subseteq V \times V$ is the set of semantic edges representing contextual relationships. Above equation transforms the refined document into a semantic graph, where each node is a domain-specific keyword and edges represent contextual relationships among keywords. Using its contextual feature vector, each graph node is initialized

$$H = \{h_1, h_2, \dots, h_n\} \quad (3.8)$$

Herein, H denotes the node feature matrix, h_i is the Contextual feature vector of node v_i , and n is the feature dimension. Equation 3.8 computes each graph node is initialized with its contextual feature vector, which is the input to graph attention learning. Before attention computation, each node feature is projected into a learnable latent feature space.

$$z_i = Wh_i \quad (3.9)$$

Where, z_i is the transformed node feature, W is the trainable weight matrix, and h_i refers to original node feature. Each node feature is transformed into a higher-dimensional latent representation using Equation 3.9, thereby helping the network learn more discriminative semantic information for computing attention weights. The semantic importance of neighbouring nodes is calculated as

$$e_{ij} = \text{ReLU}(a^T [z_i \parallel z_j]) \quad (3.10)$$

where, e_{ij} is the unnormalized attention coefficient, a is the learnable attention vector, $[z_i \parallel z_j]$ is the concatenated neighboring node features, and ReLU is the activation function. The semantic relevance of two connected nodes is computed by equation 3.10. Its attention scores for two neighbours who have a stronger contextual relation are greater than those for neighbours with a weaker tone. The attention coefficients are normalised through the SoftMax function;

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} \quad (3.11)$$

Equation 3.11 allows us to normalise the attention scores between all semantically connected nodes, so that their sum is 1, and it also enables us to emphasise the furthest relevant semantic neighbours in the network adaptively. Here, α_{ij} refers to normalized attention coefficient, and $N(i)$ is the neighbour set of nodes i . The normalised attention coefficients are used to obtain the new node embedding by aggregating neighbourhood information.

$$h'_i = \sigma \left(\sum_{j \in N(i)} \alpha_{ij} Wh_j \right) \quad (3.12)$$

Where, h'_i is the Updated node representation, $\sigma(\cdot)$ activation function, α_{ij} is the normalized attention coefficient, Wh_j denotes the transformed neighbouring feature. Semantic information is then accumulated from neighbouring nodes according to the attention weights learned by Equation 3.12. A node can retain its own semantic features and acquire contextual information from linked domain concepts. The updated node embeddings are then combined to form the final relation-aware feature matrix.

$$F_R = \{h'_1, h'_2, \dots, h'_n\} \quad (3.13)$$

Where, F_R is the relationship aware feature matrix and h'_i is the updated embedding of node i . The updated node embeddings are concatenated and multiplied by the matrix to

produce the final relation-aware feature representation. This feature matrix contains both local semantic interactions and global contextual dependencies among domain-specific concepts. It will be fed to a Hybrid CNN–BiLSTM classifier for high-precision multi-domain content classification.

3.3 Hybrid CNN–BiLSTM approach

Based on the feature representation, the proposed model employs a Hybrid CNN–BiLSTM architecture for domain classification. The relationship-aware feature matrix F_R is fed into the classifier, where the CNN learns local semantic patterns with 1D convolution, followed by ReLU and max-pooling, as shown in Fig.1. The pooled features are then fed into a BiLSTM to model the long-range contextual correlations. In the end, the contextualised representation passes through a fully connected layer, and the Softmax classifier predicts the probabilities of the domain classes; the one with the highest probability is selected as the output label. The output of GAT, the relationship-aware feature matrix, is denoted as

$$X = F_R = \{x_1, x_2, \dots, x_m\} \in \mathbb{R}^{m \times d} \quad (3.14)$$

Where, X is the feature matrix, x_i is the feature vector of the i^{th} relationship-aware feature, m denotes the number of selected relationship aware features, and F_R is the Relationship-aware feature representation. The relationship-aware feature matrix obtained from the GAT is represented as in Equation 3.14. These compacted semantic features are then fed into the Hybrid CNN–BiLSTM classifier.

$$C_i = \sigma \left(\sum_{j=0}^{k-1} W_j x_{i+j} + b \right) \quad (3.15)$$

where, C_i is the convolution feature map, W_j is the convolutional kernel, x_{i+j} is the input feature vector, k is the kernel size, b is the bias, and $\sigma(\cdot)$ activation function. Based on the neighboring relationship-aware vectors, local semantic features are extracted by the convolution layer. The 2D convolution of 2D input blocks with 2D filters and the 1D convolution of 1D input sequences with 1D filters are popular in many fields. The convolution feature maps are sub-sampled via Max-Pooling.

$$P_i = \max_{j \in \Omega_i} (C_j) \quad (3.16)$$

Here, P_i is the pooled feature, and Ω_i is the pooling window. The max activation within each pooling window is taken in Equation 3.16, which maintains the prominent semantic information while decreasing the feature size. The Sequence of pooled features is cursorily processed in the forward direction.

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \quad (3.17)$$

Equation 3.17 captures contextual dependencies by running a sequence from the left to the right. The reverse is true for the same sequence. Herein, x_t is the input feature at time step t , $\vec{h}_t$ is the forward hidden state, and $\vec{h}_{t-1}$ is the previous hidden state. The reverse direction also learns contextual information in Equation 18, where future semantic information can guide feature learning.

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}) \quad (3.18)$$

Herein, $\overleftarrow{h}_t$ is the backward hidden state, and $\overleftarrow{h}_{t+1}$ is the next hidden state. The hidden states of the two directions are concatenated.

$$H_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3.19)$$

The forward and backward context representations are concatenated to form rich semantic features in 19. Herein, H_t is the Bidirectional contextual representation, and $[\cdot; \cdot]$ is the Concatenation operator. The contextual representation is then turned into a discriminative feature vector.

$$Z = W_d H + b_d \quad (3.20)$$

Equation 3.20 is used to map the contextual features to a higher-level representation for multi-class classification. Here, Z is the dense layer output, W_d is the weight matrix, H lies the BiLSTM output, and b_d is the bias vector. The probability of every domain class is calculated as

$$P(y = c | Z) = \frac{\exp(W_c Z + b_c)}{\sum_{i=1}^C \exp(W_i Z + b_i)} \quad (3.21)$$

Where, $P(y = c | Z)$ is the probability of every domain class, W_c is the weight matrix of class c , b_c is the bias, and c is the total number domain class. The target class for prediction is

$$\hat{y} = \arg \max_c P(y = c | Z) \quad (3.22)$$

The posterior probability for each content-category is calculated by Equation (3.21), and then d is classified into the category with the maximum probability score by Equation (3.22). The ultimate output of the proposed framework is the predicted domain label along with its classification probability, where each news article is classified into one among five classes i.e. Business, Technology, Sports, Education, and Entertainment.

4. RESULT AND DISCUSSION

The framework was developed in Python using Visual Studio Code (VS Code). The experiments are performed on a public Kaggle multi-domain news dataset with 5 categories: Business, Technology, Sports, Entertainment, and Education. The dataset is split 70% for training and 30% for testing, ensuring consistent model evaluation. The presented solution Hybrid CNN–BiLSTM classifier is evaluated using accuracy, precision, recall, and F1-score. Two state-of-the-art methods, i.e., DATL and DT-UDAL, are used to evaluate the proposed method's performance and demonstrate its excellent domain-dependent content classification.

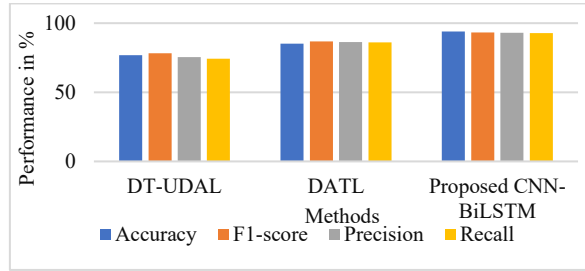


Figure.2. Comparison of accuracy and recall performance

Figure 2 presents a comparison of the accuracy, precision, recall, and F1-score between our CNN–BiLSTM model and DT-UDAL and DATL. DT-UDAL achieved 76.90%, 75.40%, 74.21%, and 78.31% in accuracy, precision, recall, and F1-score, respectively, and DATL further improved these results to 85.24%, 86.39%, 86.09%, and 86.90%. Our proposed CNN–BiLSTM achieved even better results, with an accuracy of 94.03%, a precision of 93.06%, a recall of 92.80%, and an F1-score of 93.27%, beating both methods.

5. CONCLUSION

In this paper, an intelligent domain-specific content categorisation using a DL-based Hybrid Approach is developed to classify multi-domain news articles into the Business, Technology, Sports, Entertainment and Education categories. In general, the proposed framework comprises three parts: The MCTRF module for contextual text pre-processing, GAT to capture semantic relations among domain-specific concepts, and a hybrid CNN-BiLSTM classifier to extract local semantic features and long-range contextual information. Results on a public Kaggle dataset demonstrate that the proposed framework attained the best performance, i.e., 94.03% for accuracy, 93.06% for precision, 92.8% for recall, and 93.27% for the F1 score, which significantly exceeded the state-of-the-art DT-UDAL and DATL approaches.

6. REFERENCE

- [1] Yuyuan Zhang, Xiaolan Wang, Feiyan Liu, Yibin Jin, Zechen Liu, Kun Shi, Using domain-specific keyword features to enhance deep learning-based pressure vessel inspection problem identification, *Engineering Applications of Artificial Intelligence*, Volume 160, Part C, 2025.
- [2] D. Tarun et al. "Domain Classification of Text Conversation using Deep Learning", *International Research Journal*, Volume 10, Issue 7, pp.368-373, 2023.
- [3] L. Liu, N. Wang, D. Liu, X. Yang, X. Gao and T. Liu, "Towards Specific Domain Prompt Learning via Improved Text Label Optimization," in *IEEE Transactions on Multimedia*, vol. 26, pp. 10805-10815, 2024, doi: 10.1109/TMM.2024.3413318.
- [4] D. K. Meghana, K. Kiran, P. Deepa Shenoy and K. R. Venugopal, "Auto-Encoder-Based Structural Correspondence Learning With Pivot Augmentation for Indian Languages," in *IEEE Access*, vol. 13, pp. 201577-201591, 2025, doi: 10.1109/ACCESS.2025.3637085.
- [5] Emanuel, R. H., Docherty, P. D., Lunt, H., & Möller, K. (2024). The effect of activation functions on accuracy, convergence speed, and misclassification confidence in CNN text classification: a comprehensive exploration: RHK Emanuel et al. *The Journal of Supercomputing*, 80(1), 292-312.
- [6] Sun, Z., Wang, Y., & Chen, Z. (2024). Fault diagnosis method for proton exchange membrane fuel cell system based on digital twin and unsupervised domain adaptive learning. *International Journal of Hydrogen Energy*, 50, 1207-1219.
- [7] Zhu, J., Guo, Y., Sun, G., Hong, L., & Chen, J. (2024). Causal prototype-inspired contrast adaptation for unsupervised domain adaptive semantic segmentation of high-resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1-17.
- [8] Nair, A. R., Singh, R. P., Gupta, D., & Kumar, P. (2024). Evaluating the impact of text data augmentation on text classification tasks using DistilBERT. *Procedia Computer Science*, 235, 102-111.
- [9] Yin, K., Li, B., Liu, C., Mostafavi, A., & Hu, X. (2026). Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics. *Advanced Engineering Informatics*, 75, 104836.
- [10] Sy, C. Y., Maceda, L. L., Canon, M. J. P., & Flores, N. M. (2024). Beyond BERT: Exploring the Efficacy of RoBERTa and ALBERT in Supervised Multiclass Text Classification. *International Journal of Advanced Computer Science & Applications*, 15(3).