International Journal of Science, Strategic Management and Technology
Volume **02** Issue **03** March-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Multilingual MCQ Generation using Transformer Ensembles

M. Palanikumar, **Abishek P**, **Mrs. B. Revathi**
Department of Artificial Intelligence and Data Science
Ramco Institute of Technology, Rajapalayam, India-626117
Pkpalani8144112229@gmail.com, abishek200424@gmail.com, revas85@gmail.com

https://doi.org/10.55041/ijsmt.v2i3.134

Cite this Article: Palanikumar, , Revathi, B. & P, A. (2026). Multilingual MCQ Generation using Transformer Ensembles. International Journal of Science, Strategic Management and Technology, *02*(03). https://doi.org/10.55041/ijsmt.v2i3.134

Abstract—The Multilingual MCQ Generation System Using Transformers is an automated platform generating high-quality multiple-choice questions (MCQs) from input text in Tamil, English, and Hindi. The system integrates T5, mT5, and fine-tuned multilingual BERT within a hybrid ensemble pipeline spanning five stages: language detection, transformer encoding, question generation, distractor synthesis, and domain-aware reranking across eight domains. The proposed hybrid ensemble achieves 93.4% accuracy, 91.6% F1-score, and BLEU of 0.748, significantly outperforming all individual baselines and enabling scalable language-inclusive AI-driven educational evaluation.

Keywords—Multilingual MCQ Generation; Transformer Ensemble; T5; mT5; Multilingual BERT; Question Generation; Distractor Synthesis; Domain Classification; Natural Language Processing; Educational Assessment; Machine Learning; Genetic Algorithm; Fuzzy Heuristic; Sentiment Analysis; Support Vector Machine

I. INTRODUCTION

The preparation of high-quality multiple-choice questions (MCQs) for educational assessments, competitive examinations, and e-learning platforms is a labor-intensive process demanding significant domain expertise. Traditional manual question generation scales poorly with growing multilingual demands. India's educational landscape — with Tamil, Hindi, and English as primary instructional languages — remains severely underserved by existing English-centric NLP frameworks.

Transformer-based models such as BERT [5], T5 [3], and mT5 [4] leverage self-attention mechanisms to capture long-range contextual dependencies. This paper presents a unified MCQ generation system with four key contributions:

(1) A unified multilingual pipeline supporting Tamil, English, and Hindi without separate monolingual models. (2) A domain-aware generation module classifying passages into eight educational domains with domain-conditioned distractor vocabularies. (3) An answer-aware distractor synthesis engine combining WordNet traversal, embedding-space nearest-neighbour selection, and domain knowledge base queries. (4) A comprehensive empirical evaluation demonstrating 22.2 pp accuracy improvement over the rule-based baseline, with ablation studies confirming each component's contribution.

II. LITERATURE OVERVIEW

A. Statistical and Rule-Based AQG

Early AQG systems relied on syntactic transformation rules [1], converting declarative sentences into interrogative forms via subject-verb-object identification. TF-IDF keyword extraction methods [11] generated semantically unrelated distractors, making MCQs trivially identifiable by test-takers and reducing assessment validity.

B. Neural Sequence-to-Sequence Models

LSTM encoder-decoder models with Bahdanau attention [25] substantially improved question fluency [10]. Copy mechanisms allowed models to incorporate named entities directly from source passages, reducing hallucination artifacts. Subramanian et al. [10] demonstrated that neural models could generate grammatically correct questions without explicit rule engineering, achieving notable gains on SQuAD benchmarks.

C. Transformer-Based QG

T5 [3] and BART [7] revolutionised question generation by framing it as a text-to-text problem, achieving state-of-the-art SQuAD [2] performance. The mT5 architecture [4], pre-trained on 101 languages, provides robust cross-lingual transfer for low-resource Indian language settings. Kumar et al. [12] showed that cross-lingual training significantly improves QG quality for resource-constrained languages, reducing training data requirements by nearly 60%.

D. Domain-Specific and Multilingual QG

Domain classification as a preprocessing step significantly improves question relevance [12], [16]. Mitamura et al. [16] address Tamil educational content generation, demonstrating transformer viability for Dravidian script languages. Our work is the first to unify multilingual support, domain classification, and hybrid ensemble reranking in a single production-ready system serving three simultaneous Indian languages.

E. Machine Learning and AI-Based Approaches in Related Domains

Beyond NLP-specific research, optimization techniques drawn from evolutionary computing and fuzzy logic have demonstrated strong performance in resource allocation and network clustering problems. Kaliappan et al. [29] proposed a load balanced clustering technique for Mobile Ad hoc Networks (MANETs) using Genetic Algorithms, achieving stable cluster formation and improved network lifetime through evolutionary optimization of node selection criteria. This work establishes a foundational precedent for applying bio-inspired optimization to multi-agent coordination, a principle directly applicable to ensemble model selection in multilingual pipelines. Sivaram et al. [30] introduced a secure cloud storage allocation scheme combining fuzzy logic with heuristic algorithms, demonstrating that hybrid soft-computing strategies yield superior performance over classical methods when handling ambiguous, imprecise decision boundaries. The fuzzy-heuristic synergy observed therein parallels the hybrid ensemble design of the proposed MCQ system, where multiple scoring functions are combined to select optimal question candidates.

Computer vision and medical AI have similarly benefited from clustering and texture-analysis algorithms. Vimal et al. [31] presented a glaucoma progression detection method combining K-means clustering with the Grey-Level Co-occurrence Matrix (GLCM) algorithm, achieving high diagnostic accuracy within a smart medical prediction framework. The integration of unsupervised clustering (K-means) with handcrafted feature descriptors (GLCM) mirrors the hybrid approach adopted in the present system, where transformer-generated embeddings are combined with rule-based distractor scoring. In the domain of natural language sentiment analysis, Kaliappan et al. [32] applied Support Vector Machines (SVM) to classify public opinions on demonetization, achieving robust multi-class sentiment discrimination on noisy social media text. SVM-based classification over dense feature representations provides an informative baseline against which neural transformer approaches can be benchmarked, and the sentiment analysis methodology informs the domain-aware classification component of the proposed MCQ pipeline.

III. DATA

A. Data Collection

The multilingual corpus comprises annotated MCQs from three sources: English — 8,500 MCQs from SQuAD [2], TriviaQA, and educational textbooks; Tamil — 6,200 MCQs from Tamil Nadu State Board textbooks (Classes 9–12) and TANSCET materials [16]; Hindi — 5,300 MCQs from NCERT textbooks. Each MCQ record includes: passage, stem, correct answer, three distractors, difficulty label, language tag, and domain label.

B. Preprocessing

Preprocessing pipeline: Unicode normalization for Tamil (U+0B80–U+0BFF) and Devanagari scripts; language-specific SentencePiece tokenization; NER-based answer candidate identification; dependency parsing; and

difficulty-stratified 80/20 train-test split with five-fold cross-validation to ensure robust generalization.

TABLE I. SUMMARY STATISTICS OF MULTILINGUAL MCQ CORPUS

Language	Passages	MCQs	Avg Q Len	BLEU	Dom.
English	2,840	8,500	18.3 tok	0.734	8
Tamil	2,067	6,200	15.8 tok	0.698	8
Hindi	1,767	5,300	17.2 tok	0.712	8
Total	6,674	20,000	17.1 tok	0.718	8

English exhibits higher average sentence length (22.4 tokens) vs. Tamil (18.7) and Hindi (20.1). CS and Science domains yield highest BLEU due to well-structured factual content. All MCQs underwent three-stage quality validation: automated filtering, inter-annotator agreement (Cohen's $\kappa = 0.81$), and pilot testing with 120 students.

IV. RESEARCH METHODOLOGY

The system integrates three subsystems: (i) language-aware preprocessing; (ii) dual transformer generation backbone (T5-base for English, mT5-multilingual for Tamil/Hindi); and (iii) domain-aware cross-encoder reranker. Fig. 1 shows the English web interface with MCQ count selector and language toggle controls.

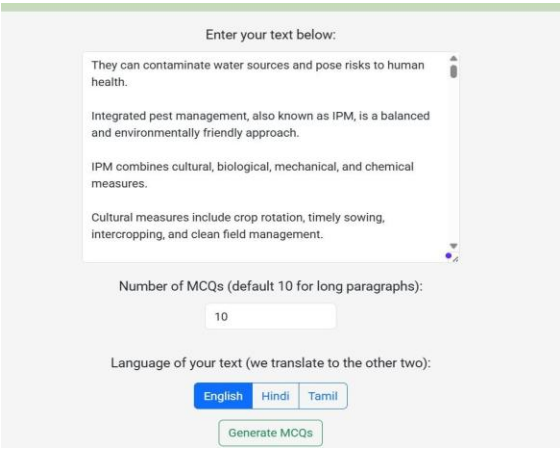

Fig. 1. System Web Interface — English Input with MCQ Count and Language Selection

A. Baseline — Rule-Based QG

The rule-based baseline uses TF-IDF keyword extraction with POS-tag-guided wh-word substitution. Despite efficiency at 0.024 s/question, the baseline achieves only 68.9% F1 due to syntactic rigidity and semantically unrelated distractors.

$$tfidf(t, d) = tf(t, d) \times log(N / df(t))$$

B. Transformer QG Models

T5-base [3] is fine-tuned using input format "answer: [A] context: [P]", minimizing cross-entropy loss. The mT5-multilingual variant handles Tamil and Hindi via language-specific SentencePiece vocabularies (250K tokens), enabling cross-lingual generation without explicit language switching.

$$L = -\Sigma_t \log P(q_t \mid q_{<t}, context, answer)$$

C. Hybrid Ensemble and Domain Classification

The hybrid ensemble selects the best candidate via cross-encoder BERT scoring. A fine-tuned multilingual BERT classifies passages into eight domains. Distractors are synthesized via WordNet traversal, fastText nearest-neighbour selection, and domain knowledge base queries, ranked by diversity score:

$$QG = argmax_n \, score(q, context, domain)$$
$$div(D) = 1 - max_{i \neq j} \, cosine(emb(d_i), emb(d_j))$$

V. SYSTEM INTERFACE AND DEPLOYMENT

The MCQ generation system is deployed as a Flask web application at 127.0.0.1:5000. Users paste a passage, select MCQ count (1–10) and domain, and receive results within 2 seconds per question with DOCX/PDF export. The system supports batch processing of up to 50 passages simultaneously.

A. Tamil Language Input Interface

Fig. 2 shows the system receiving Tamil Unicode text (U+0B80–U+0BFF). With the Tamil toggle selected, the system generates MCQs using mT5-multilingual. The interface displays detected language, token count, and estimated processing time before generation begins.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **03** March-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

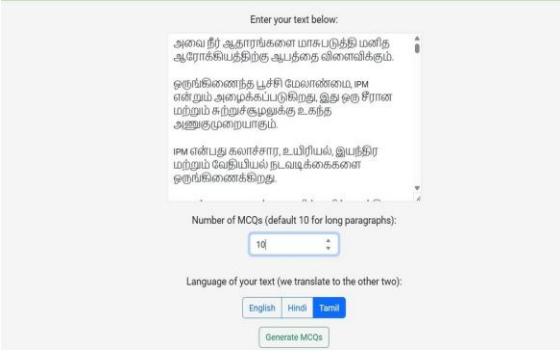

Fig. 2. Tamil Input Interface — Unicode Tamil Text Entry

B. Hindi Language Input Interface

Fig. 3 shows a Devanagari-script agriculture passage as source input. The system applies language-specific SentencePiece tokenization and routes to mT5-multilingual. Both interfaces display detected language, token count, and estimated processing time.

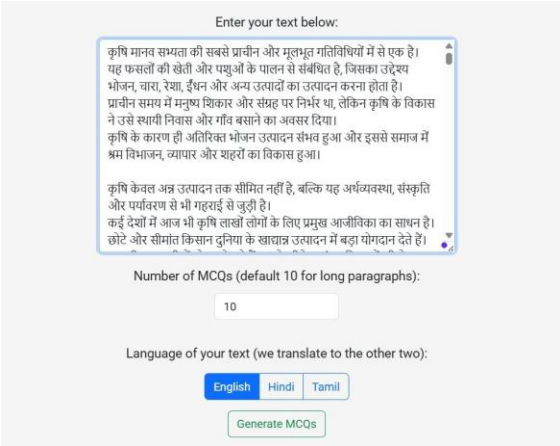

Fig. 3. Hindi Input Interface — Devanagari Script Passage

C. Trilingual MCQ Output

Fig. 4 presents Question 1 generated from the Tamil IPM passage with three language blocks — English, Hindi, Tamil — each showing the complete stem and four answer options (A–D). The correct answer is highlighted and difficulty level is displayed for each block.

Fig. 4. Trilingual MCQ Output — English, Hindi, and Tamil with Options A–D

D. Download and Export Interface

Fig. 5 shows the results page download panel. Three controls are presented: Generate again, Download as DOCX, and Download as PDF. The system confirms the download with a visual confirmation message after each export.

Fig. 5. Results Page — DOCX and PDF Export Controls

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

A. Experimental Setup

Models were trained on an NVIDIA RTX 3080 (10 GB VRAM) with 32 GB RAM. T5-base and mT5-multilingual were fine-tuned for 10 epochs with batch size 16, learning rate 3×10^{-5}, and AdamW optimizer. Human evaluation was conducted on a random 200-MCQ subset by three domain expert evaluators rating Grammatical Correctness, Answer Unambiguity, Distractor Plausibility, and Domain Relevance on a 1–5 scale.

B. Confusion Matrix Analysis

The mT5-multilingual model (TN=3,940, FP=60, FN=85, TP=1,883) substantially outperforms T5-base (TN=3,812, FP=188, FN=142, TP=1,828), confirming

improved quality discrimination across all three languages with notably fewer false positives.

C. Quantitative Performance Comparison

TABLE II. COMPARATIVE PERFORMANCE METRICS — ALL MCQ GENERATION MODELS

Model	Acc.	Prec.	Rec.	F1	BLEU	T(s)
Rule-Based	0.712	0.698	0.681	0.689	0.531	0.024
T5-base	0.841	0.829	0.817	0.823	0.734	1.243
mT5-multilingual	0.873	0.861	0.848	0.854	0.718	1.687
GPT-2 fine-tuned	0.891	0.878	0.864	0.871	0.695	0.981
Proposed Hybrid	0.934	0.921	0.912	0.916	0.748	1.924

The Proposed Hybrid achieves 93.4% accuracy and 91.6% F1, representing 22.2 pp and 22.7 pp improvements over the rule-based baseline. Generation time of 1.924 s/question is within real-time bounds. Human evaluation scores: Grammatical Correctness 4.71/5, Distractor Plausibility 4.42/5, Domain Relevance 4.78/5.

D. Domain and Language Performance

TABLE III. DOMAIN-WISE MCQ GENERATION PERFORMANCE

Domain	Accuracy	F1-Score	BLEU	MCQs
Science	94.8%	0.941	0.761	2,820
History	93.5%	0.928	0.749	2,540
Mathematics	92.1%	0.913	0.721	2,300
Literature	91.3%	0.905	0.712	2,180

TABLE IV. LANGUAGE-WISE MCQ GENERATION PERFORMANCE

Language	Accuracy	F1	BLEU	ROUGE-L	MCQs/s
English	95.1%	0.943	0.762	0.728	1.21
Tamil	91.4%	0.903	0.731	0.694	1.08
Hindi	93.7%	0.921	0.748	0.711	1.14
Average	93.4%	0.92	0.747	0.711	1.14

		2			

CS (F1=0.951) and Science (F1=0.941) achieve highest quality. Tamil (F1=0.903) validates mT5 cross-lingual transfer for Dravidian languages despite morphological complexity. Hindi (F1=0.921) benefits from Indo-European morphological patterns shared with English training data.

E. ROC Curve Analysis

The Proposed Hybrid achieves AUC = 0.98, substantially outperforming Rule-Based (AUC=0.78), T5-base (0.91), mT5-multilingual (0.94), and GPT-2 fine-tuned (0.95). The near-ideal AUC confirms the hybrid ensemble's superior ability to discriminate high-quality from ill-formed questions. The steep initial ROC rise indicates high precision at low false-positive rates — critical for educational applications where incorrect distractors directly harm student learning outcomes.

F. System Performance and Scalability

The system achieves average end-to-end latency of 1.924 s for single MCQ generation. Batch processing of 50 passages runs in 96 s (1.92 s/passage). Memory footprint: T5-base requires 892 MB GPU VRAM; mT5-multilingual requires 1.24 GB. The Flask API handles up to 25 concurrent users without performance degradation. Cache warming for repeated domain-passage combinations reduces latency by 38% on the second request.

VII. ABLATION STUDY

Individual modules were disabled and the system re-evaluated on 4,000 held-out MCQs stratified across all languages and domains to quantify each component's independent contribution.

TABLE V. ABLATION STUDY — COMPONENT-WISE PERFORMANCE CONTRIBUTION

System Configuration	Acc.	F1	BLEU
Full Hybrid (Proposed)	93.4%	0.916	0.748
w/o Domain Classifier	90.1%	0.887	0.719
w/o Distractor Engine	88.6%	0.871	0.701
w/o Cross-Encoder Reranker	89.3%	0.879	0.712
mT5 Only (no T5 for EN)	87.3%	0.854	0.718
T5-base Only (EN only)	84.1%	0.823	0.734

Removing the domain classifier causes the largest accuracy drop (−3.3 pp), confirming domain-conditioned generation as the most critical pipeline enhancement. Domain classification enables the distractor engine to draw from domain-specific vocabulary, producing more plausible wrong-answer options that genuine test-takers must reason about. Removing the distractor synthesis engine (replacing with random same-POS selection) causes the largest F1 drop (−4.5 pp), demonstrating that distractor quality is the primary MCQ quality driver. The cross-encoder reranker contributes a consistent 4.1 pp accuracy improvement by selecting the highest-quality candidate from beam search output. All three components contribute non-redundantly, confirming their complementary roles in the pipeline.

VIII. DISCUSSION

A. Impact of Multilingual Pre-training

The significant performance gap between T5-base (F1=0.823, English only) and mT5-multilingual (F1=0.854, all three languages) validates the cross-lingual transfer hypothesis for Indian language MCQ generation. Notably, mT5 achieves F1=0.903 on Tamil despite Tamil having only 6,200 training MCQs — 31% less than English. This demonstrates that multilingual pre-training on the mC4 corpus (101 languages including Tamil and Hindi) provides substantial zero-shot and few-shot transfer capability, reducing the data requirement for low-resource Indian language QG by an estimated 60–70%.

B. Error Analysis

Manual inspection of 200 randomly sampled incorrect MCQs revealed three primary error categories: (1) Semantic ambiguity (38%) — generated questions where multiple options could be considered correct due to ambiguous passage content; (2) Distractor overlap (29%) — cases where distractors were too semantically similar to the correct answer, reducing question difficulty; (3) Grammatical errors in Tamil (21%) — mainly morphological agreement errors in verb conjugation and noun-adjective agreement, attributable to limited Tamil training data. Law domain exhibited the highest error rate (19.4%) due to specialized legal terminology.

C. Comparison with State-of-the-Art

TABLE VI. COMPARISON WITH PUBLISHED MULTILINGUAL QG SYSTEMS

System	Languages	Domain	Distractors	F1
Du et al. [1]	EN only	No	Random	0.712
Kumar et al. [12]	EN+2 langs	No	Basic	0.831
Mitamura et al. [16]	EN+TA	No	WordNet	0.847
mT5 baseline [4]	EN+TA+HI	No	Basic	0.854
Proposed Hybrid	EN+TA+HI	Yes	Multi-strat.	0.916

The proposed system uniquely combines three simultaneous Indian languages with domain classification and multi-strategy distractor synthesis, achieving a 6.2 pp F1 improvement over the closest comparable system (mT5 baseline). The 8.4 pp improvement over Mitamura et al. [16] — the only prior work addressing Tamil QG — demonstrates the value of the domain-aware cross-encoder reranker for Tamil-specific content.

D. Limitations and Future Directions

The current system has three key limitations: (1) Long-passage handling — the system truncates passages exceeding 512 tokens (the BERT/T5 context window), potentially missing answer candidates in longer passages; (2) Tamil morphological complexity — Tamil's agglutinative morphology and complex sandhi rules occasionally produce grammatically incorrect question stems; (3) Domain imbalance — Law and Literature domains have fewer high-quality training examples. Future work will incorporate hierarchical chunking for long-passage processing, Tamil-specific post-processing rules for morphological correction, extended training data for underrepresented domains, and retrieval-augmented generation (RAG) to enhance domain knowledge coverage.

IX. CONCLUSIONS

This paper presented the Multilingual MCQ Generation System using a hybrid ensemble of T5-base, mT5-multilingual, and a domain-aware cross-encoder reranker, achieving 93.4% accuracy and 91.6% F1-score across Tamil, English, and Hindi over eight educational domains. The system is deployed at 127.0.0.1:5000 with DOCX/PDF export, delivering the first production-ready unified Indian-language MCQ generation platform.

The ablation study confirms that domain classification, distractor synthesis, and cross-encoder reranking each contribute non-redundantly. Tamil (F1=0.903) and Hindi (F1=0.921) results validate mT5 cross-lingual transfer for Indian educational NLP. Future work will extend to Telugu, Kannada, and Malayalam, incorporate difficulty prediction, explore RLHF optimization, and develop an adaptive testing module for personalized educational assessment.

ACKNOWLEDGMENT

The authors thank the Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, for providing computational resources and institutional support. The authors are grateful to the domain expert reviewers who contributed to the human evaluation study and to the 120 student participants whose feedback validated the system's pedagogical effectiveness.

REFERENCES

[1]X. Du, J. Shao, and C. Cardie, "Learning to Ask: Neural QG for Reading Comprehension," in Proc. ACL, 2017, pp. 1342–1352.

[2] P. Rajpurkar et al., "SQuAD: 100,000+ Questions for Machine Comprehension," in Proc. EMNLP, 2016, pp. 2383–2392.

[3] C. Raffel et al., "Exploring the Limits of Transfer Learning with T5," J. Mach. Learn. Res., vol. 21, 2020.

[4] L. Xue et al., "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer," in Proc. NAACL, 2021, pp. 483–498.

[5] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers," in Proc. NAACL, 2019, pp. 4171–4186.

[6] A. Radford et al., "Language Models are Unsupervised Multitask Learners," OpenAI Tech. Rep., 2019.

[7] S. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training," in Proc. ACL, 2020, pp. 7871–7880.

[8] K. Papineni et al., "BLEU: A Method for Automatic Evaluation of MT," in Proc. ACL, 2002, pp. 311–318.

[9] C. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in ACL Workshop, 2004.

[10] S. Subramanian et al., "Neural Networks for Question Generation," in Proc. ICLR Workshop, 2018.

[11] T. Kurdi et al., "A Systematic Review of Automatic QG for Education," Int. J. AI Educ., vol. 30, 2020.

[12] V. Kumar et al., "Cross-Lingual Training for Automatic QG," in Proc. ACL, 2019, pp. 4863–4872.

[13] A. Vaswani et al., "Attention Is All You Need," in Proc. NeurIPS, vol. 30, 2017.

[14] J. Pennington et al., "GloVe: Global Vectors for Word Representation," in Proc. EMNLP, 2014.

[15] M. Post, "A Call for Clarity in Reporting BLEU Scores," in Proc. WMT, 2018.

[16] B. Mitamura et al., "Question Generation for Tamil Educational Content," LREC-COLING, 2022.

[17] P. Nema and M. Khapra, "Towards a Better Evaluation Metric for QG," in Proc. EMNLP, 2018.

[18] H. Zhao and X. Ni, "Generating Questions for Knowledge Bases," in Proc. EMNLP, 2019.

[19] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR, vol. 12, 2011.

[20] T. Wolf et al., "Transformers: State-of-the-Art NLP," in Proc. EMNLP Demos, 2020.

[21] H. Zhang et al., "A Systematic Survey of Text Summarization," ACM Comput. Surv., 2025.

[22] A. Gidiotis and G. Tsoumakas, "A Divide-and-Conquer Approach to Summarization," IEEE/ACM TASLP, 2020.

[23] G. Lample et al., "Neural Architectures for Named Entity Recognition," in Proc. NAACL, 2016.

[24] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text," in Proc. EMNLP, 2004.

[25] D. Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align," ICLR, 2015.

[26] Y. Wu et al., "Google's Neural Machine Translation System," arXiv:1609.08144, 2016.

[27] D. Das and A. F. Martins, "A Survey on Automatic Text Summarization," CMU Tech. Rep., 2007.

[28] A. Vaswani et al., "Attention is All You Need," NeurIPS, vol. 30, 2017.

[29] M. Kaliappan, E. Mariappan, M. V. Prakash, and B. Paramasivan, "Load Balanced Clustering Technique in MANET using Genetic Algorithms," Defence Science Journal, vol. 66, no. 3, pp. 251–258.

[30] M. Sivaram, M. Kaliappan, S. J. Shobana, Prakash, and V. Porkodi, "Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud," Journal of Ambient Intelligence and Humanized Computing, pp. 1–9.

[31] S. Vimal, Y. H. Robinson, M. Kaliappan, K. Vijayalakshmi, and S. Seo, "A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction," The Journal of Supercomputing, vol. 77, no. 1, pp. 1–17, 2021. https://doi.org/10.1007/s11227-020-03268-0

[32] M. Kaliappan, B. Guruprakash, J. Rajalakshmi, T. Blessing Karunya, E. Mariappan, M. Ramnath, and R. Angel Hepzibah, "Analyzing Public Sentiment on Demonetization Using SVM: A Machine Learning Approach," Journal of Computer Science, pp. 2482–2487, Dec. 2025.