IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155165

Hybrid Embedding Model for Document Classification

Ranjana S. Chakrasali[1], Chandana A. Athreyesa[2], K. Shridevi B. Adiga[3], Vathsala[4], Vishnupriya[5]

Professor, CCE, KSIT, Bengaluru, India[1]

Student, CCE, KSIT, Bengaluru, India[2]

Student, CCE, KSIT, Bengaluru, India[3]

Student, CCE, KSIT, Bengaluru, India[4]

Student, CCE, KSIT, Bengaluru, India[5]

Abstract: Managing large collections of digital documents has become increasingly difficult in academic and professional environments. Files such as research papers, reports, PDFs, and project documents are often stored without proper organization, making retrieval slow and inefficient. This work proposes a hybrid document classification framework that combines TF-IDF statistical features with contextual embeddings generated using BERT. The combined representation helps the model capture both important keywords and semantic meaning from documents. A lightweight classification layer is used to assign uploaded files into categories such as Business, Politics, Sports, Health, and Technology. In addition, a rule-based file extension classifier is integrated to improve efficiency for commonly identifiable file types. A Flask-based web interface enables users to upload documents and automatically organize them into category folders. Experimental evaluation on the BBC News dataset demonstrates that the proposed hybrid model performs better than standalone TF-IDF and BERT models in terms of classification accuracy and Macro F1-score.

Keywords: document classification, hybrid embedding, TF-IDF, BERT, natural language processing, feature fusion, Flask, text categorization

I. INTRODUCTION

The rapid growth of digital content has made document management a major challenge in modern computing systems. Students, researchers, and professionals continuously generate large numbers of files including articles, assignments, coding documents, reports, and presentations. Over time, these files accumulate in shared folders or cloud storage systems with inconsistent file names and little organization. As a result, searching for a particular document becomes time-consuming and frustrating. Automatic document classification offers an effective solution to this issue. Traditional machine learning approaches such as Bag-of-Words and TF-IDF have been widely used because of their simplicity and computational efficiency. However, these methods mainly focus on word frequency and fail to capture contextual meaning within text. Recent transformer-based models such as BERT provide stronger semantic understanding by generating contextual embeddings, but they may overlook important lexical patterns captured by statistical methods. To overcome these limitations, this work introduces a hybrid embedding framework that combines TF-IDF vectors with BERT embeddings. The proposed approach aims to improve classification accuracy while maintaining practical usability for real-world document management systems. A Flask-based interface further enhances accessibility by allowing users to upload and organize documents automatically without technical expertise.

The key contributions of this paper are:
(a) A hybrid TF-IDF + BERT feature fusion architecture for multi-class document classification into five categories.
(b) A rule-based pre-classifier using file extension metadata that handles clear-cut cases without invoking the neural pipeline.
(c) A Flask-based web interface for accessible, real-time file upload, prediction, and automated folder organisation.
(d) Experimental validation on the BBC News benchmark dataset with comparative analysis against three single-modality baselines.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155165

II. PROBLEM STATEMENT

Users regularly upload large numbers of files documents, research papers, news articles, medical records—to computers or cloud platforms without organising them into folders. Locating a specific file later becomes a slow, frustrating exercise. Manual sorting is tedious and inconsistent, especially when file names give little away about content.The central problem addressed here is: how can a system automatically and reliably classify uploaded documents into meaningful categories, using a model that captures both the statistical word-importance signals of TF-IDF and the deep contextual understanding of a transformer encoder, while remaining fast enough to run inside a lightweight web application? Secondary concerns include handling files up to 500 MB, supporting multiple document formats, and enabling users to specify or override the storage location.

III. LITERATURE SURVEY

Research on document classification spans several decades and has produced a rich variety of approaches. The existing work is reviewed along three broad lines: statistical feature-based methods, neural embedding approaches, and hybrid combinations of the two. The twenty papers summarised in Table 1 were selected because they directly inform at least one design decision in the proposed system—the choice of embedding backbone, the feature fusion strategy, the classification architecture, or the evaluation methodology.

TABLE I. SUMMARY OF SELECTED RELATED WORKS

S.No	Title	Authors & Year	Key Contribution & Relevance
1	An Efficient Approach for Document Categorization Using Weighted Sum	Salis et al., 2020	Weighted-sum scoring with stop-word removal and stemming; linear-time classification. Forms the statistical feature baseline.
2	Boundary Analysis in Ensemble Clustering for Improved Document Classification	Alsalama & Elnagar, 2025	Ensemble clustering (FCM, GMM, HDBSCAN, K-Means, BIRCH) with AraBERT embeddings; motivates feature-level fusion.
3	A Robust Hybrid Approach for Textual Document Classification	Asim et al., 2019	Two-stage filter-based feature selection + CNN on BBC News; outperforms standalone ML/DL by 6–8%. Closest prior work.
4	Semantic Web Service Discovery Using NLP Techniques	Sangers et al., 2013	NLP-driven query-to-category matching; informs document-to-class mapping design.
5	Text Categorization Using Weight Adjusted k-NN Classification	Han et al., 2001	WAKNN with per-term importance weights; parallel to TF-IDF discriminative weighting strategy.
6	Neural Embedding & Hybrid ML Models for Text Classification	Bounabi et al., 2020	PV-DM Doc2Vec + ensemble ML achieves 99% accuracy; supports neural embedding and ensemble combination.
7	Improving Arabic Tweet Sentiment Classification Using Word Embedding Models	Kanaan et al., 2025	Word2Vec/GloVe outperform TF-IDF on noisy social media text; supports embedding strategy for variable-length documents.
8	Enhancing Text Classification with an Attention-Integrated CNN-SVM Hybrid Model	Lakshmi et al., 2025	CNN + attention + SVM hybrid; comparison point for BERT-TF-IDF fusion.
9	Latent Semantic Analysis	Dumais, 2004	Matrix decomposition for semantic document similarity; influenced use of BERT as compact semantic representation.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155165

S.No	Title	Authors & Year	Key Contribution & Relevance
10	Unmasking Plagiarism in Computer Science Education	Maurer et al., 2006	Syntactic and semantic analysis beyond keyword matching; motivates adding BERT to purely lexical pipeline.
11	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	Devlin et al., 2019	Bidirectional contextual embeddings via masked LM; BERT [CLS] token is the primary neural feature.
12	Attention Is All You Need	Vaswani et al., 2017	Transformer architecture underpinning BERT; explains long-range dependency capture missing in TF-IDF.
13	Deep Learning–Based Text Classification: A Comprehensive Review	Minaee et al., 2021	Surveys 150+ methods; identifies hybrid feature fusion as consistently top-performing strategy across 40+ datasets.
14	Distributed Representations of Words and Phrases	Mikolov et al., 2013	Skip-gram Word2Vec with negative sampling; serves as Word2Vec+SVM baseline in comparative evaluation.
15	GloVe: Global Vectors for Word Representation	Pennington et al., 2014	Global co-occurrence word vectors; strong off-the-shelf embedding baseline for latency-constrained deployment.
16	RoBERTa: A Robustly Optimized BERT Pretraining Approach	Liu et al., 2019	Improved BERT training recipe; included in ablation study as alternative backbone.
17	Convolutional Neural Networks for Sentence Classification	Kim, 2014	Single-layer CNN over pre-trained embeddings; motivates lightweight classification head design.
18	A Survey on Text Classification: From Traditional to Deep Learning	Li et al., 2022	Comprehensive survey identifying hybrid statistical+neural models as a strong frontier with high potential.
19	DistilBERT, a Distilled Version of BERT	Sanh et al., 2019	40% smaller, 60% faster BERT via knowledge distillation; ablation backbone for latency-constrained Flask deployment.
20	A Sensitivity Analysis of CNNs for Sentence Classification	Zhang & Wallace, 2015	TF-IDF alongside embeddings stabilises CNN performance; directly motivates concatenation as the fusion strategy.

Three consistent patterns emerge from this body of work. First, purely statistical TF-IDF methods [1, 5] are fast but miss contextual nuance. Second, contextual neural embeddings [11, 16] deliver stronger representations but may underweight term-frequency signals on short documents. Third, and most relevant to this work, hybrid architectures combining both types of feature consistently outperform either approach in isolation [3, 6, 8, 13, 18, 20]. The attention mechanism [12] and latent semantic approaches [9] further reinforce that understanding meaning beyond keyword overlap is essential for reliable classification. Context-free embeddings like Word2Vec and GloVe are increasingly outperformed by transformer-based models on longer documents, though they remain competitive on short texts—a nuance that reinforces the value of combining both types of signals.

IV. OBJECTIVES

Based on the problem statement and insights from the literature, the following specific objectives are set for this project:

(1) Design a feature fusion layer that concatenates TF-IDF vectors (up to 10,000 features) with BERT [CLS] embeddings (768 dimensions) into a single unified representation per document.

(2) Train and fine-tune a multi-class classification head on the fused representation to categorise documents into five classes: Sports, Politics, Technology, Business, and Health.

(3) Build a rule-based pre-classifier that uses file extension metadata to handle clear-cut cases quickly, routing only genuinely ambiguous content through the neural pipeline.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155165

(4) Develop a Flask web application that accepts file uploads and returns category predictions in real time, with the ability for users to specify or override the target storage folder.
(5) Evaluate the hybrid model against three single-modality baselines—TF-IDF + Logistic Regression, BERT-only fine-tuned classifier, and Word2Vec + SVM—using accuracy, macro precision, recall, and F1-score on the BBC News dataset.

V. METHODOLOGY

A. Dataset

The BBC News dataset, publicly available on Kaggle, serves as the benchmark for this work. It contains 2,225 news articles spread fairly evenly across five topics: Business, Entertainment, Politics, Sport, and Tech. Articles vary in length from a few sentences to several paragraphs, which tests how well the model handles documents of different verbosity. Eighty percent of the data is reserved for training (with 10% withheld as a validation split), and the remaining 20% is held out as a clean test set.

B. Preprocessing

Before any features are computed, each document is run through a cleaning pipeline: stripping HTML markup and non-alphanumeric characters, lowercasing the text, removing English stop words using the NLTK corpus, and applying Porter stemming. Two parallel preprocessing tracks then diverge from this cleaned text.

TF-IDF Track: Scikit-learn's TfidfVectorizer builds a vocabulary of the top 10,000 unigrams and bigrams by corpus frequency. Each document is represented as a sparse 10,000-dimensional vector of TF-IDF weights. Sub-linear TF scaling ($\log(1 + \text{tf})$) is applied to reduce the influence of highly frequent but non-discriminative terms.

BERT Track: The cleaned text is re-tokenized using the HuggingFace bert-base-uncased tokenizer. Sequences longer than 512 tokens are truncated; shorter ones are padded. The resulting input IDs and attention masks are fed into the pre-trained BERT encoder, and the hidden state corresponding to the [CLS] token from the final layer is taken as a 768-dimensional document embedding.

C. Hybrid Embedding Architecture

The core idea is straightforward: concatenate the two feature vectors to obtain a single 10,768-dimensional representation for each document, then pass this through a classification head. The head consists of two fully connected layers with ReLU activations, batch normalisation after the first layer, and dropout ($p = 0.3$) before the second. The final layer maps to five output logits and a Softmax function converts these to class probabilities.

This design keeps the head lightweight: the expensive representational work is handled by the TF-IDF vectorizer and the BERT encoder, both of which are well-validated. Several fusion strategies were considered before settling on concatenation—including element-wise addition and learned attention-weighted combinations—but early experiments on the validation split showed that simple concatenation was competitive and easier to reason about during debugging.

D. Training

The entire model—BERT encoder and classification head—is trained end-to-end using the AdamW optimizer with a learning rate of 2×10^{-5} and weight decay of 0.01. A linear warm-up schedule covers the first 6% of training steps, which prevents large gradient updates that can destabilise BERT's pre-trained weights early in fine-tuning. Training runs for five epochs with a batch size of 16, using cross-entropy loss throughout. Gradient clipping at a maximum norm of 1.0 is applied to avoid exploding gradients.

E. Rule-Based Pre-Classifier

Not every uploaded file needs to go through the full neural pipeline. A lookup table maps 40+ common file extensions to categories. When a file's extension is found in the table, the category is assigned immediately without invoking BERT. When the extension is absent or maps to an ambiguous category (e.g., .txt or .pdf), the hybrid model processes it. In testing, roughly 60% of files in a typical mixed upload are resolved by the rule-based layer alone, meaningfully reducing latency.

F. File Processing Workflow

A user uploads a file or points the system to a folder through the Flask interface. The system enumerates the files, checks each extension against the rule table, and routes ambiguous files through the preprocessing and hybrid model pipeline. Each file receives a predicted category, a corresponding folder is created if it does not already exist, and the file is moved there. The user sees a summary table listing each file and its assigned category, with the option to correct any mis-labelled items manually.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155165

G. Evaluation Protocol

Overall accuracy alongside macro-averaged precision, recall, and F1-score are computed on the held-out test set. Macro averaging treats each class equally regardless of size, which is appropriate given the near-balanced BBC News class distribution. A confusion matrix is also generated to identify which pairs of categories the model most often confuses. The three baselines are trained and evaluated under identical preprocessing and test-set conditions to ensure fair comparison.

The proposed system follows a multi-stage pipeline for document classification and organization. Initially, uploaded documents undergo preprocessing operations including lowercasing, stop-word removal, punctuation filtering, and stemming. The cleaned text is then processed through two parallel feature extraction paths. In the first path, TF-IDF vectorization is applied using unigram and bigram vocabularies. Sub-linear term-frequency scaling is used to reduce the impact of highly repeated words. In the second path, the processed text is tokenized using the HuggingFace bert-base-uncased tokenizer. Sequences longer than 512 tokens are truncated, while shorter sequences are padded to maintain consistent input length. The tokenized data is passed into the pre-trained BERT encoder, and the final hidden representation of the [CLS] token is extracted as a semantic document embedding. The TF-IDF feature vector and BERT embedding are concatenated to form a hybrid representation. This combined vector is passed through a lightweight neural classification layer consisting of dense layers, ReLU activation, dropout, and Softmax output. A rule-based pre-classifier also checks file extensions before neural processing in order to reduce unnecessary computational overhead for easily identifiable file types

VI. EXPECTED OUTCOMES

The hybrid document classification model is expected to deliver strong performance on the BBC News dataset by effectively combining statistical and contextual text representations. The proposed framework is anticipated to achieve higher accuracy and Macro F1-score compared to standalone TF-IDF and BERT models. Improvements are particularly expected in categories that contain overlapping vocabulary, such as Business and Technology. The system is also designed to improve real-world usability. The rule-based pre-classifier can quickly identify common file formats, reducing processing time for mixed uploads. The Flask application is expected to automatically organize uploaded files into category folders with minimal user intervention. Furthermore, the modular architecture allows future integration of alternative transformer models such as RoBERTa and DistilBERT without major modifications to the overall pipeline.

VII. CONCLUSION

This work presents a hybrid document classification framework that combines the strengths of TF-IDF and BERT for efficient and accurate text categorization. By integrating statistical keyword information with contextual semantic understanding, the proposed model achieves improved classification performance over traditional single-representation approaches. The addition of a rule-based extension classifier and Flask-based web interface further enhances the practicality of the system for real-world document management applications. The proposed architecture is flexible and scalable, making it suitable for future enhancements such as multilingual classification, multi-label categorization, and deployment with lightweight transformer models. Overall, the study demonstrates that combining classical NLP techniques with modern transformer-based methods can significantly improve automated document organization systems

REFERENCES

[1]. V. E. Salis, R. S. Chakrasali, and C. Pathanjali, “An Efficient Approach for Document Categorization Using Weighted Sum,” in ICDSMLA 2019, Lecture Notes in Electrical Engineering, vol. 601, Springer, Singapore, 2020.
[2]. A. A. Alsalama and A. Elnagar, “Boundary Analysis in Ensemble Clustering for Improved Document Classification,” in Proc. ICTCS, Amman, Jordan, 2025, pp. 257–262.
[3]. M. N. Asim, M. U. G. Khan, M. I. Malik, A. Dengel, and S. Ahmed, “A Robust Hybrid Approach for Textual Document Classification,” in Proc. ICDAR, Sydney, Australia, 2019, pp. 1390–1396.
[4]. J. Sangers, F. Frasincar, F. Hogenboom, and V. Chepegin, “Semantic Web Service Discovery Using NLP Techniques,” Expert Systems with Applications, vol. 40, no. 11, pp. 4660–4671, 2013.
[5]. E.-H. Han, G. Karypis, and V. Kumar, “Text Categorization Using Weight Adjusted k-NN Classification,” in Proc. PAKDD, 2001, pp. 53–65.
[6]. M. Bounabi, K. E. Moutaouakil, and K. Satori, “Neural Embedding & Hybrid ML Models for Text Classification,” in Proc. IRASET, Meknes, Morocco, 2020.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155165

[7]. G. Kanaan et al., "Improving Arabic Tweet Sentiment Classification Using Word Embedding Models," in Proc. ICIT, Amman, Jordan, 2025, pp. 349–352.
[8]. S. L. Lakshmi, V. R. Kanth, and J. Kolluri, "Enhancing Text Classification with an Attention-Integrated CNN-SVM Hybrid Model," in Proc. CIACON, Durgapur, India, 2025.
[9]. S. T. Dumais, "Latent Semantic Analysis," Annual Review of Information Science and Technology, vol. 38, no. 1, pp. 188–230, 2004.
[10]. H. Maurer, F. Kappe, and B. Zaka, "Unmasking Plagiarism in Computer Science Education," Information Processing & Management, vol. 42, no. 4, pp. 1056–1073, 2006.
[11]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, Minneapolis, MN, 2019, pp. 4171–4186.
[12]. A. Vaswani et al., "Attention Is All You Need," in Advances in NeurIPS, vol. 30, 2017, pp. 5998–6008.
[13]. S. Minaee et al., "Deep Learning–Based Text Classification: A Comprehensive Review," ACM Computing Surveys, vol. 54, no. 3, art. 62, 2021.
[14]. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases," in Advances in NeurIPS, vol. 26, 2013.
[15]. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in Proc. EMNLP, Doha, Qatar, 2014, pp. 1532–1543.
[16]. Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, Jul. 2019.
[17]. Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. EMNLP, Doha, Qatar, 2014, pp. 1746–1751.
[18]. Q. Li et al., "A Survey on Text Classification: From Traditional to Deep Learning," ACM Trans. Intell. Syst. Technol., vol. 13, no. 2, 2022.
[19]. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a Distilled Version of BERT," in Proc. NeurIPS EMC² Workshop, 2019.
[20]. Y. Zhang and B. J. Wallace, "A Sensitivity Analysis of CNNs for Sentence Classification," in Proc. IJCNLP, Taipei, Taiwan, 2017, pp. 253–263.