Deep learning-based system for document analysis and classification*

Oleg Pursky[1,*,†], Tetiana Filimonova[1,†], Anna Selivanova[1,†], Sofia Minetska[1,†] and Pavlo Demidov[1,†]

[1] *State University of Trade and Economics, Kioto str. 19, 02156, Kyiv, Ukraine*

Abstract

In this work, an automated system of document analysis and classification based on deep learning methods has been developed. The system is able to process text data using both traditional models (BoW+LR and TF-IDF+LR) and deep learning models (LSTM, BERT). The system allows users to select a data set, perform text preprocessing, perform classification using four approaches, and receive detailed reports on the performance of the models. The main attention is paid to comparing the effectiveness of the methods, which is implemented through the evaluation of metrics (accuracy, completeness, F1-measure) and visualization of the results. The BERT model showed the highest accuracy. The testing confirmed the system's functionality.

Keywords

document analysis, text classification, deep learning, neural networks, automation

1. Introduction

The modern world is characterized by a constant increase in the amount of information, which requires effective methods of processing, analyzing and structuring it. This problem is especially acute in the context of text documents, the number of which is growing exponentially due to the development of the Internet, social networks, electronic libraries and digital archives.

In modern information systems, document analysis and classification are fundamental processes for effective management, processing and interpretation of information. Thanks to these processes, organizations are able not only to store large volumes of data, but also to systematize them, highlight key patterns and use the results obtained to make informed decisions in business, science, education and other industries [1].

The relevance of the work lies in the fact that the development of a document analysis and classification system based on deep learning allows to significantly increase the efficiency of processing large volumes of text data, ensuring high accuracy and automation of processes.

The purpose of the research is to develop an automated document analysis and classification system using deep learning methods to optimize text data processing processes, improve classification accuracy, and automate information analysis.

*Applied Information Systems and Technologies in the Digital Society (AISTDS-2025), October 01, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

o.pursky@knute.edu.ua (O. Pursky); t.filimonova@knute.edu.ua (T. Filimonova); a.selivanova@knute.edu.ua (A. Selivanova); s.minetska_fit_14_21_b_d@knute.edu.ua (S. Minetska); p.demidov@knute.edu.ua (P. Demidov);

0000-0002-1230-0305 (O. Pursky); 0000-0001-9467-0141 (T. Filimonova); 0000-0001-6559-1508 (A. Selivanova); 0009-0009-5661-9658 (S. Minetska); 0000-0003-4085-2809 (P. Demidov)

2. Literature review

The rapid development of machine learning allows simplifying research in various fields of science. The application of machine learning methods for document classification is a relevant direction, using achievements to solve practical problems.

The article [2] analyzes a number of machine learning methods that can be used to preserve cultural heritage, summarizes their characteristics and areas of application, describes the advantages of using machine learning in recognizing cultural heritage documents, reveals the mechanism of document classification using the Random Forest algorithm as one of the most effective tools for training the model; presents a conceptual model of a machine learning information system for document classification.

The study [3] solves the applied problem of developing an automatic system for processing scanned/photo documents based on machine learning approaches. The ata augmentation approach is used - this is a technique for artificially increasing the size of the training data set by creating modified versions of images in the data set. The paper proposes to use generative-competitive neural networks as a generator of unique instances of the training sample. The study of the effects of the main parameters of generative-competitive networks on the quality of new instances generated by the generator has been conducted. The results of this paper show the possibility of using the proposed approach when developing systems of this type for artificial, programmatic expansion of the training sample with new, unique representatives.

The article [4] is devoted to the development of a system for automated analysis of medical text using modern artificial intelligence and natural language processing technologies. Various methods and technologies were used, such as text tokenization, natural language processing, text classification and clustering, semantic analysis and text generation. The developed system is able to recognize and classify symptoms, establish possible diagnoses and provide treatment recommendations. Integration with electronic medical records ensures the relevance and completeness of information, which is important for medical practice. The test results showed a high level of accuracy and efficiency in the analysis of medical texts.

The problem of organizing software and algorithms for neural networks for natural language processing is considered in the work [5]. A multilevel methodology is proposed that combines the preliminary classification of text arrays, semantic clustering and the use of a bidirectional LSTM neural network model. The practical implementation of the method was tested using an automated text analysis application, which demonstrated a stable reduction in the loss function and acceptable resource consumption. It was noted that the main advantages of the developed approach are the ability to adapt to different types of text data, reducing resource consumption while maintaining high quality of analysis, and suitability for deployment in environments with low computing power.

The study [6] considered ways to classify student documents downloaded in PDF format and required for university education. Three possible methods for solving this problem were proposed. The first approach is based on optical character recognition (OCR) and traditional machine learning methods. The second is based exclusively on deep learning. The third is based on a combination of deep learning methods based on entropy. The proposed methods can classify twelve different types of digital documents. The validity of the proposed methods was verified by the Student Affairs Department of Kocaeli University in Turkey. The system not only improved the efficiency of online document upload stages for students, but also reduced the human cost of document tracking. The highest F-measure (94.45%) was obtained by the EfficientNetB3 and ExtraTree ensemble.

The article [7] presents an approach to building a model of an intelligent document management system using machine learning methods to ensure effective work of employees in organizations. A number of problems were solved to optimize each of the document management subsystems, as a result of which a model of an intelligent document management system was developed, which can be effectively applied in enterprises, state and corporate institutions. At the same time, the paper considers the application of thematic modeling methods and text analysis algorithms based on a multi-agent approach, which can be used to build an intelligent document management system.

The multimodal deep learning architecture TechDoc is presented in [8]. The model uses three types of information, including natural language texts and descriptive images in documents, as well as connections between documents. The architecture synthesizes a convolutional neural network, a recurrent neural network and a graph neural network using an integrated learning process. The architecture was applied to a large multimodal database of technical documents, and the model was trained to classify documents based on the hierarchical system of the International Patent Classification. The results show that TechDoc has higher classification accuracy than unimodal methods and other modern benchmarks.

In the work [9] an analysis and comparison of recurrent neural networks (RNN), in particular the LSTM and GRU models, was carried out to determine the effectiveness of these models in analyzing the sentiment of text messages. A public dataset from the Twitter social networking platform was used to train the model. The obtained research results allow us to conclude that compared to LSTM, the GRU model was more effective in the task of text classification.

3. Methodology

The developed document analysis and classification system using is a software package designed for automatic processing of texts from various domains (news, movie reviews, medical transcriptions) and determining their category or sentiment. The main approach to classification is to use both traditional machine learning methods Bag of Words and TF-IDF with logistic regression (BoW+LR and TF-IDF+LR), and deep learning models, in particular the recurrent neural network LSTM and the BERT (Bidirectional Encoder Representations from Transformers) architecture. The system provides flexibility in choosing a data set, comparing the effectiveness of different methods, and creating visualized reports with the results.

Modern document analysis and classification systems use a variety of approaches, including traditional machine learning methods and deep learning technologies. Traditional methods, such as BoW+LR and TF-IDF+LR, are easy to implement and effective for small or structured data sets, but they have limitations in detecting complex semantic relationships and processing large, diverse text data. In contrast, deep learning methods such as recurrent neural networks (RNNs, including LSTM) and transformers (e.g., BERT) provide significantly higher accuracy and flexibility when dealing with texts containing nuance, context, and noise.

Traditional methods (BoW+LR and TF-IDF+LR) are fast and suitable for small data sets, but they do not take into account semantic relationships and context, which limits their effectiveness when working with texts with complex structure or ambiguity (for example, medical terms or sarcastic reviews). In contrast, LSTM and BERT allow modeling long text dependencies and semantic relationships, which is crucial for accurate classification of documents in areas such as news, reviews or medical transcriptions.

It should be noted that deep learning methods face problems, especially in terms of interpreting the obtained results.

Taking into account these limitations, system architecture is presented, designed according to the modular principle, which allows it to be easily adapted to new data sets or models (Fig. 1).

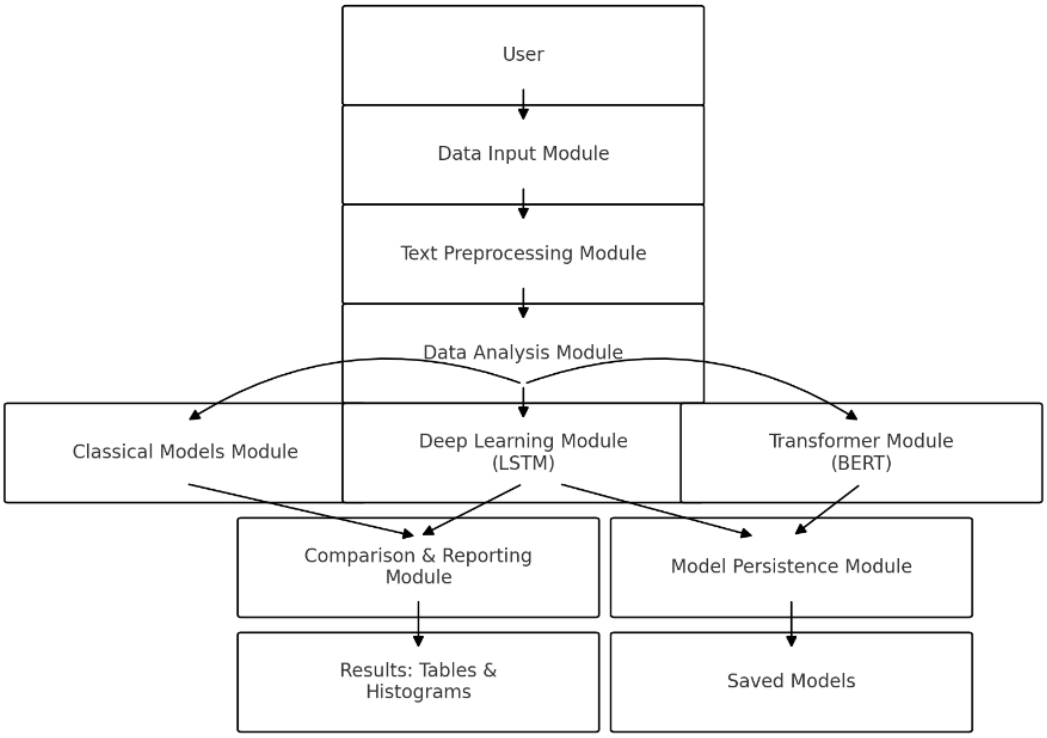

Figure 1: System architecture.

With Figure 1, it can be seen that the system has a modular architecture, consisting of separate components that are responsible for:

1. Data entry (selection and loading of the data set);
2. Text preprocessing;
3. Document classification using traditional methods (BoW+LR and TF-IDF+LR) and deep learning models (LSTM, BERT);
4. Data analysis (visualization of category distribution);
5. Evaluation of results and comparison of models;
6. Storage of trained models.

This architecture ensures its scalability and extensibility. Experimental research into the effectiveness of the proposed document classification model is aimed at developing a methodology for evaluating four approaches: BoW+LR and TF-IDF+LR, LSTM and BERT [10, 11].

The research methodology is based on a literature review and theoretical modeling of the document classification process using deep learning methods and classical approaches. Let us list the main stages of the research.

7. Data selection - to assess the adaptability of the models to different types of documents, it is planned to use various text corpora, such as news articles, user reviews and specialized texts (for example, news sets) [1].
8. Pre-processing - to prepare the data for analysis, text cleaning (removal of stop words, punctuation marks), tokenization and normalization are provided.
9. Modeling approaches:
    - BoW+LR - converting text into a word frequency vector with subsequent application of a statistical classifier [12];
    - TF-IDF+LR - vectorization based on word weights to extract key features [13] (TfidfVectorizer);

- LSTM - context-sensitive sequence modeling using recurrent neural networks [14] (LSTM layer);
- BERT - using transformers for bidirectional contextual text analysis [15].

10. Evaluation - comparing models by defined metrics, taking into account their theoretical performance on different types of data [16].

The methodology takes into account the influence of factors such as the volume of data, its structure and stylistic features, which may affect the classification results [17]. The developed methodology and experimental design create the basis for evaluating BoW+LR, TF-IDF+LR, LSTM and BERT in document classification. The selected metrics (accuracy, completeness, F1-measure) will provide a comprehensive analysis of the effectiveness [18]. The main aspects are given in Table 1.

Table 1
Experimental design for comparing approaches

Approach	Purpose	Conditions	Expectations
BoW + LR	Basic approach evaluation	Vectorization (up to 10,000 words)	High speed, lower accuracy
TF-IDF + LR	Comparison with BoW take into account word weights	Vectorization with a focus on terms	Improved accuracy
LSTM	Sequence processing analysis	Fixed length (100 tokens)	Higher accuracy on complex texts
BERT	Context analysis evaluation	Pretrained model, fine-tuning	Highest accuracy, more resources

Theoretical expectations indicate an accuracy advantage for BERT, but practical testing will refine these assumptions [10].

4. Results and Discussion

Data preparation for deep learning models is a critical step in the document analysis and classification system, which ensures high-quality processing of text data from three datasets - News Category Dataset, IMDB Movie Reviews and medical transcriptions - for further use in LSTM and BERT models in the document classification module. Data tokenization, vectorization and optimization were performed.

To demonstrate the results, the News Category Dataset [19] dataset was selected, which contains the fields: category, title, authors, links, short description and news publication date. The performance of four models BoW+LR, TF-IDF+LR, LSTM and BERT is compared on this corpus.

The goal was not only to determine the best approach, but also to identify the strengths and weaknesses of each model, as well as to develop recommendations for improving the system, taking into account its practical application in real conditions. Testing included several stages: system startup, data structure analysis, performance metrics evaluation (accuracy, completeness, F1-measure), results visualization and model performance comparison for different usage scenarios.

Testing began with system initialization, which allowed us to check its basic performance and preparation for data processing. The output data (Figure 2) is presented, which reflect the system startup and initial configuration.

```
GPU availability: True
TensorFlow version: 2.17.1
Loaded stop words from /kaggle/input/nltk-stopwords
Choose a dataset:
1 - News Category Dataset v3 (news)
2 - IMDb Movie Reviews (reviews)
3 - Medical Transcriptions (medical data)
Choose a dataset (1, 2, or 3):  1
Loaded News Category Dataset, number of records: 50000
```

Figure 2: Console output during system startup and boot.

The console shows the successful loading of stopwords from the /kaggle/input/nltk-stopwords directory, GPU availability (True), which positively affects the processing speed of deep models, the TensorFlow version (2.17.1), and the dataset selection menu. The user selected the News Category Dataset (1).

Before classification, an initial analysis of the structure of the News Category Dataset dataset was performed to assess its complexity and identify key categories for testing. A graph (Figure 3) is shown, which displays the distribution of categories in the dataset. The first 15 categories are displayed.

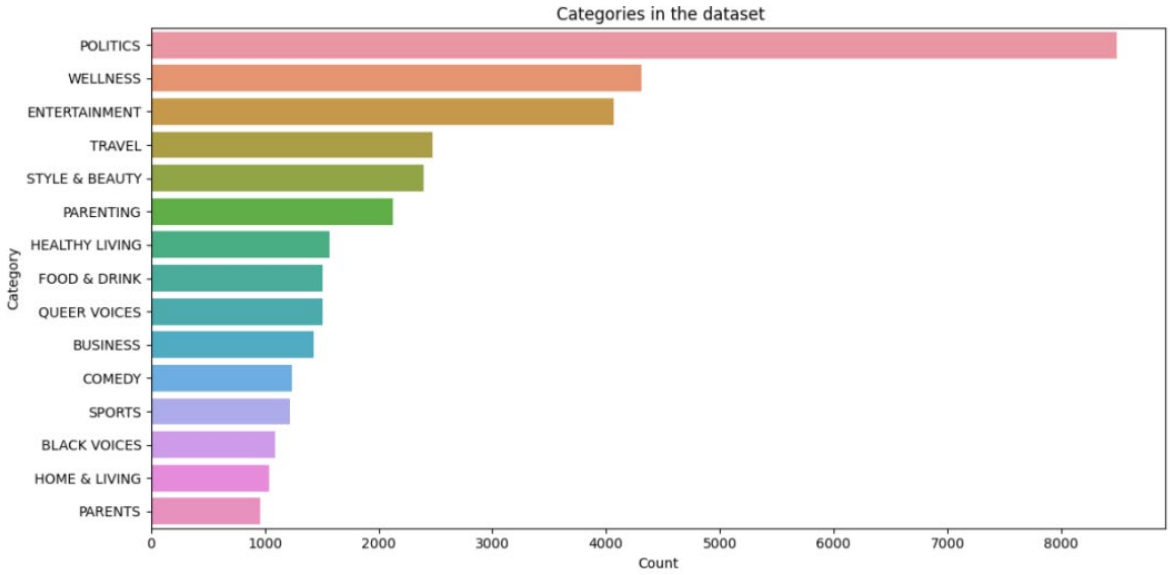

Figure 3: Distribution of categories in the News Category Dataset.

The histogram displays the frequencies of news categories in descending order. POLITICS has the highest values, followed by WELLNESS and ENTERTAINMENT, while other categories are characterized by significantly lower frequencies. This justifies the selection of the ten most common categories for further processing. The analysis revealed an uneven distribution, which affected the classification strategies.

The performance testing of the models was carried out on a dataset of the top 10 categories of the News Category Dataset. Each model was evaluated by the metrics of accuracy, completeness and F1-measure (macro/weighted), the results are presented in the form of tables, confusion matrices and graphs. This approach allowed us to analyze in detail the effectiveness of each approach and identify their features. Figure 4 shows the results of the BoW model on a given sample.

	Category	Count
0	POLITICS	8491
1	WELLNESS	4315
2	ENTERTAINMENT	4071
3	TRAVEL	2476
4	STYLE & BEAUTY	2400
5	PARENTING	2130
6	HEALTHY LIVING	1571
7	FOOD & DRINK	1505
8	QUEER VOICES	1505
9	BUSINESS	1430

Figure 4: Top 10 BoW categories.

The Figure 4 displays the number of records for each category, which is used for pre-filtering the data before classification. Figures 5 and 6 contain the classification report of the BoW+LR and TF-IDF+LR models. Note that they were trained with class balancing, and the optimal hyperparameters were determined using the GridSearchCV method with five-fold cross-validation.

Category	precision	recall	f1-score	support
BUSINESS	0.527600	0.601400	0.562100	286.000000
ENTERTAINMENT	0.777000	0.832300	0.803700	787.000000
FOOD & DRINK	0.768300	0.790000	0.779000	319.000000
HEALTHY LIVING	0.334400	0.331200	0.332800	314.000000
PARENTING	0.729800	0.733200	0.731500	431.000000
POLITICS	0.917900	0.856700	0.886200	1696.000000
QUEER VOICES	0.780700	0.734300	0.756800	286.000000
STYLE & BEAUTY	0.878700	0.828400	0.852800	472.000000
TRAVEL	0.769500	0.803900	0.786300	515.000000
WELLNESS	0.668900	0.691900	0.680200	873.000000
accuracy	0.764500	0.764500	0.764500	0.764500
macro avg	0.715300	0.720300	0.717100	5979.000000
weighted avg	0.769700	0.764500	0.766400	5979.000000

Figure 5: Classification report of the BoW+LR model.

Category	precision	recall	f1-score	support
BUSINESS	0.528500	0.615400	0.568700	286.000000
ENTERTAINMENT	0.782000	0.829700	0.805200	787.000000
FOOD & DRINK	0.775100	0.799400	0.787000	319.000000
HEALTHY LIVING	0.322000	0.331200	0.326500	314.000000
PARENTING	0.736600	0.733200	0.734900	431.000000
POLITICS	0.922500	0.856100	0.888100	1696.000000
QUEER VOICES	0.795500	0.734300	0.763600	286.000000
STYLE & BEAUTY	0.870500	0.826300	0.847800	472.000000
TRAVEL	0.766100	0.807800	0.786400	515.000000
WELLNESS	0.673700	0.695300	0.684300	873.000000
accuracy	0.765800	0.765800	0.765800	0.765800
macro avg	0.717300	0.722900	0.719300	5979.000000
weighted avg	0.772400	0.765800	0.768300	5979.000000

Figure 6: Classification report of the TF-IDF+LR model.

Confusion matrices for these models are also given in Figures 7 and 8.

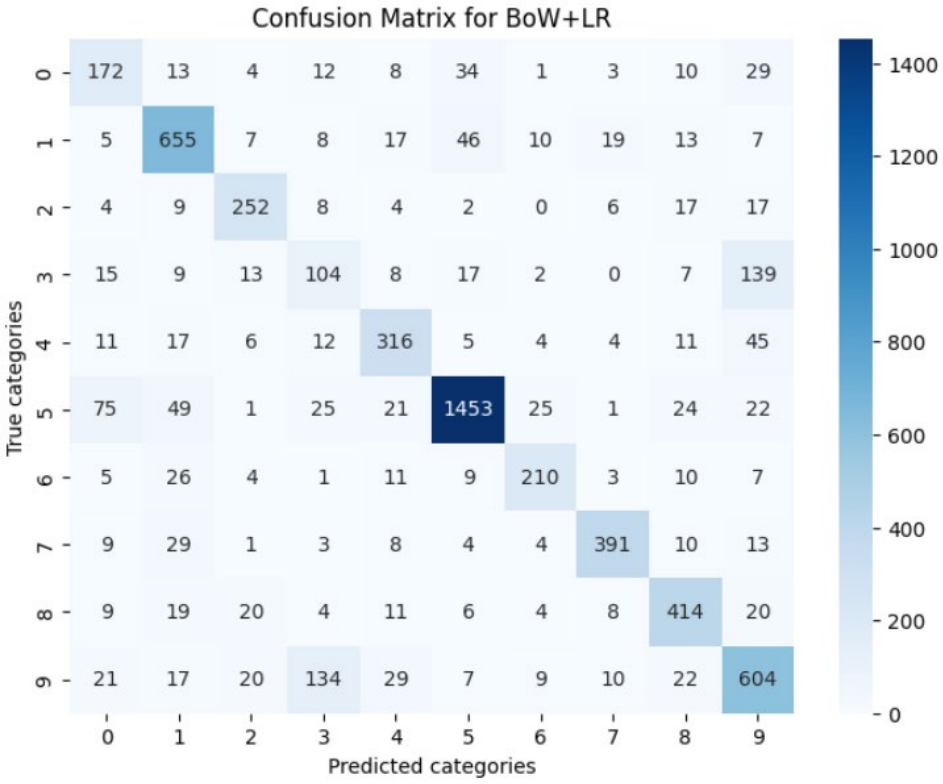

Figure 7: Confusion matrix of the BoW+LR model.

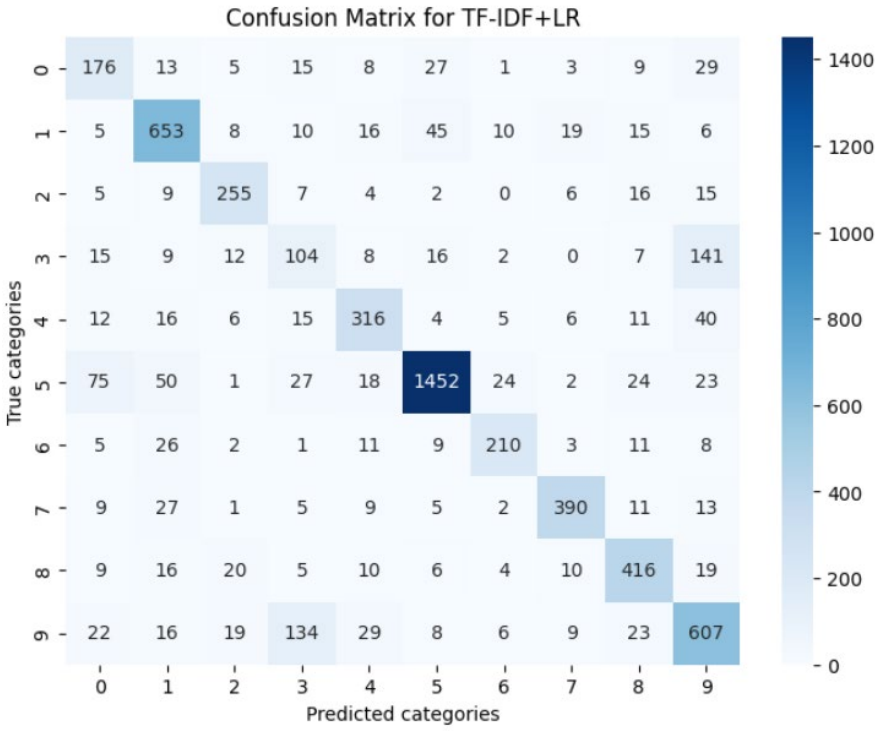

Figure 8: Confusion matrix of the TF-IDF+LR model.

Within the study, two models - BoW+LR and TF-IDF+LR were compared. Both approaches showed similar results (accuracy 0.765; macro-F1 0.72; weighted-F1 0.77), with TF-IDF+LR providing a small but stable improvement in weighted F1 compared to BoW+LR.

Confusion matrix analysis shows misclassification between thematically adjacent classes. The transition from BoW+LR to TF-IDF+LR moderately increases the proportion of correct classifications and reduces cross-over errors. In the course of the study, an LSTM neural network was built, the architecture of which is presented in Figure 9.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 256)	5,120,000
spatial_dropout1d (SpatialDropout1D)	(None, 200, 256)	0
bidirectional (Bidirectional)	(None, 200, 512)	1,050,624
dropout (Dropout)	(None, 200, 512)	0
lstm_1 (LSTM)	(None, 200, 128)	328,192
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 128)	16,512
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

Total params: 19,549,856 (74.58 MB)
Trainable params: 6,516,618 (24.86 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 13,033,238 (49.72 MB)

Figure 9: LSTM model architecture.

The figure shows that the model contains two recurrent layers, as well as additional levels for representation formation, regularization, feature aggregation and classification.

AdamW was used as the optimizer, and categorical crossentropy was used as the loss function. During training, early stopping and speed reduction were applied to prevent overtraining; as a result, the model was trained for 5 epochs. The graphs (Figure 10) are shown, which show the dynamics of LSTM learning. The graphs illustrate the reduction in loss and increase in accuracy during training.

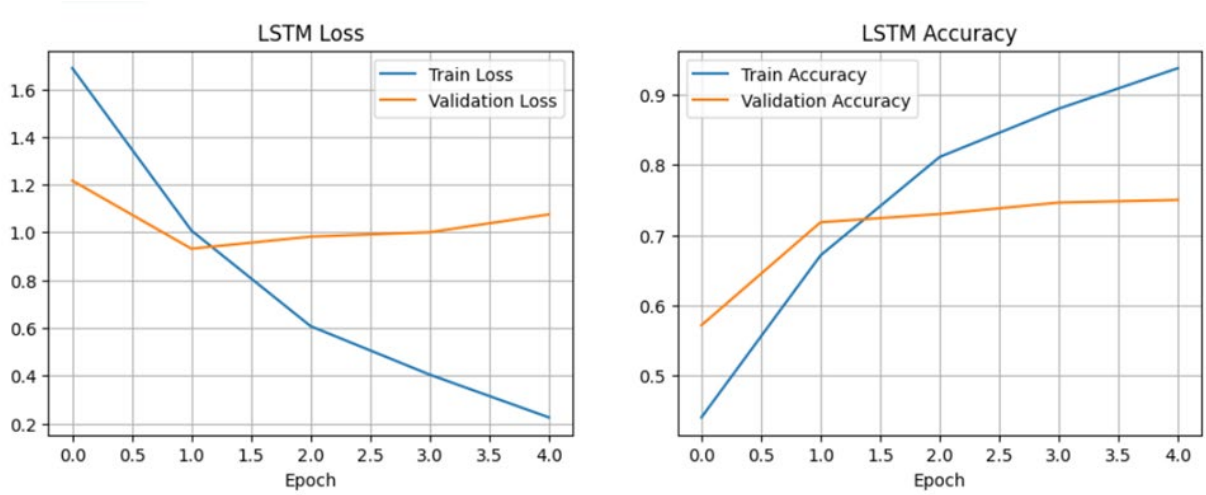

Figure 10: Loss and accuracy graphs of the LSTM model.

A Figure 11 is provided, which contains the classification report for the LSTM model.

Category	precision	recall	f1-score	support
0	0.300500	0.192300	0.234500	286.000000
1	0.690400	0.838600	0.757300	787.000000
2	0.514200	0.623800	0.563700	319.000000
3	0.505400	0.149700	0.231000	314.000000
4	0.734300	0.679800	0.706000	431.000000
5	0.870600	0.868500	0.869500	1696.000000
6	0.652700	0.545500	0.594300	286.000000
7	0.808600	0.760600	0.783800	472.000000
8	0.659400	0.702900	0.680500	515.000000
9	0.634500	0.753700	0.689000	873.000000
accuracy	0.712800	0.712800	0.712800	0.712800
macro avg	0.637100	0.611500	0.611000	5979.000000
weighted avg	0.703600	0.712800	0.700200	5979.000000

Figure 11: LSTM model classification report.

A confusion matrix (Figure 12) is given, which shows the distribution of LSTM predictions.

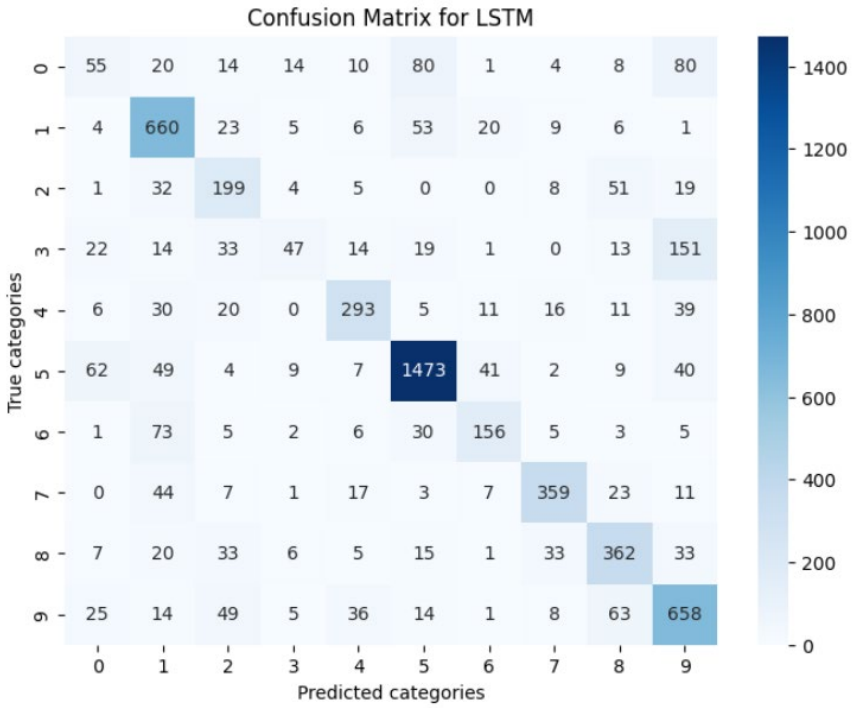

Figure 12: Confusion matrix of the LSTM model.

From Figures 11 and 12, we can conclude that compared to the BoW+LR and TF-IDF+LR models, the LSTM demonstrates worse integral indicators: accuracy 0.713 vs. 0.765, macro-F1 0.611 vs. 0.72, weighted-F1 0.7 vs. 0.77. The confusion matrix for the LSTM has a larger off-diagonal dispersion, especially for less represented classes, local improvements are observed only for individual categories. Therefore, on short texts (title + short description) TF-IDF+LR is a stronger baseline approach, while LSTM requires more powerful features or contextual models to achieve an advantage.

In this study, the BERT model was used for the text classification task. The input sequences were formed by the WordPiece tokenizer, and the token indices and attention mask were passed to the model. Training was performed using fine-tuning over 6 epochs with a batch size of 16 (to limit memory consumption). Optimization was performed using AdamW with a linear learning rate decay. Categorical crossentropy was used as the loss function. The model architecture is shown in Figure 13.

```
Model: "tf_bert_for_sequence_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bert (TFBertMainLayer)      multiple                  109482240

 dropout_37 (Dropout)        multiple                  0

 classifier (Dense)          multiple                  7690

=================================================================
Total params: 109489930 (417.67 MB)
Trainable params: 109489930 (417.67 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 13: BERT model architecture.

The learning curves (Figure 14) are shown, which demonstrate the BERT learning process. The graphs show that the accuracy increases during the training process, the loss function approaches zero.

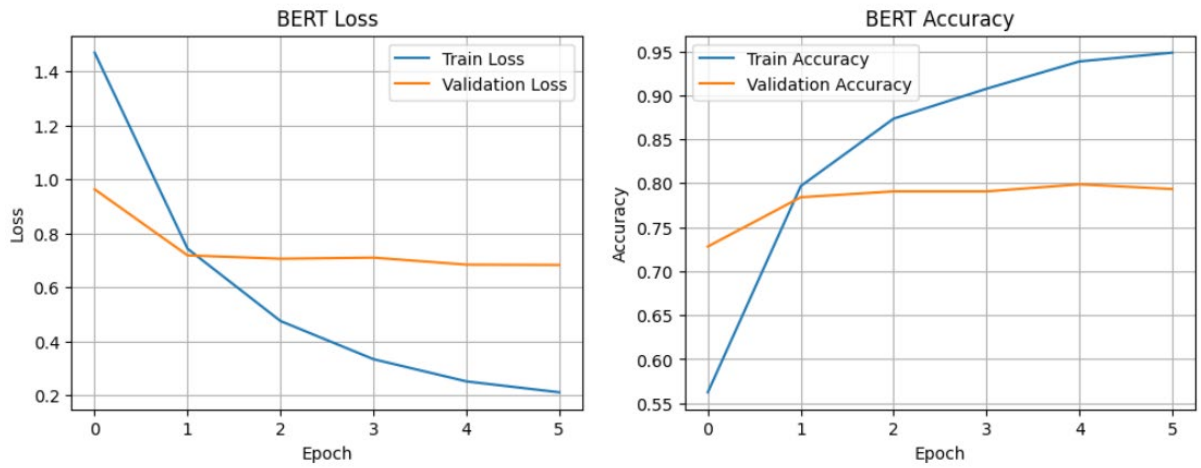

Figure 14: Loss and accuracy graphs of the BERT model.

The next Figure 15 is provided, which contains the classification report of the BERT model, which shows an overall accuracy of 0.793, with a macro-F1 of 0.740 and a weighted-F1 of 0.786.

Category	precision	recall	f1-score	support
0	0.500000	0.633300	0.558800	30.000000
1	0.731200	0.755600	0.743200	90.000000
2	0.911800	0.794900	0.849300	39.000000
3	0.571400	0.190500	0.285700	42.000000
4	0.754100	0.836400	0.793100	55.000000
5	0.873300	0.885300	0.879300	218.000000
6	0.744700	0.814000	0.777800	43.000000
7	0.844800	0.803300	0.823500	61.000000
8	0.942300	0.859600	0.899100	57.000000
9	0.734800	0.843500	0.785400	115.000000
accuracy	0.793300	0.793300	0.793300	0.793300
macro avg	0.760800	0.741600	0.739500	750.000000
weighted avg	0.792000	0.793300	0.785800	750.000000

Figure 15: BERT model classification report.

A confusion matrix (Figure 16) is presented, which illustrates BERT predictions. The matrix shows a predominance of diagonal elements, which indicates a higher accuracy of BERT compared to the baseline models. Errors are single and mainly fall on semantically close classes, which confirm the better recognition ability of the contextual model on short texts.

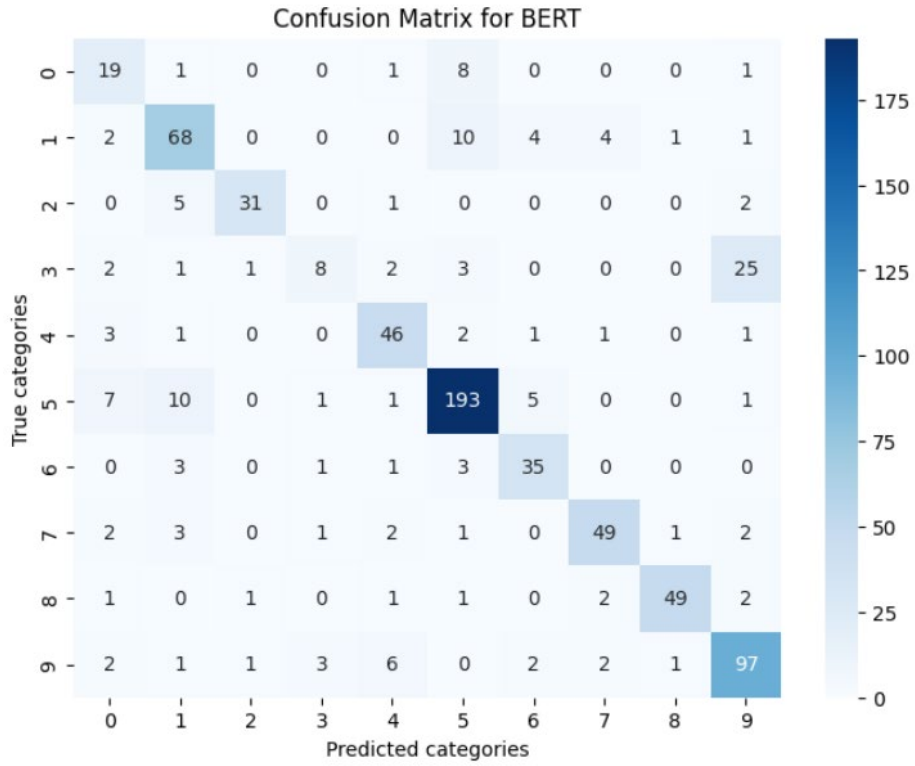

Figure 16: Confusion matrix of the BERT model.

The final stage of testing involved comparing all models to determine their efficiency and practical applicability. A histogram (Figure 17) is presented, which reflects the comparison of the accuracy of all system models (BoW+LR, TF-IDF+LR, LSTM and BERT) for the News Category Dataset, created by the results evaluation and model comparison module.

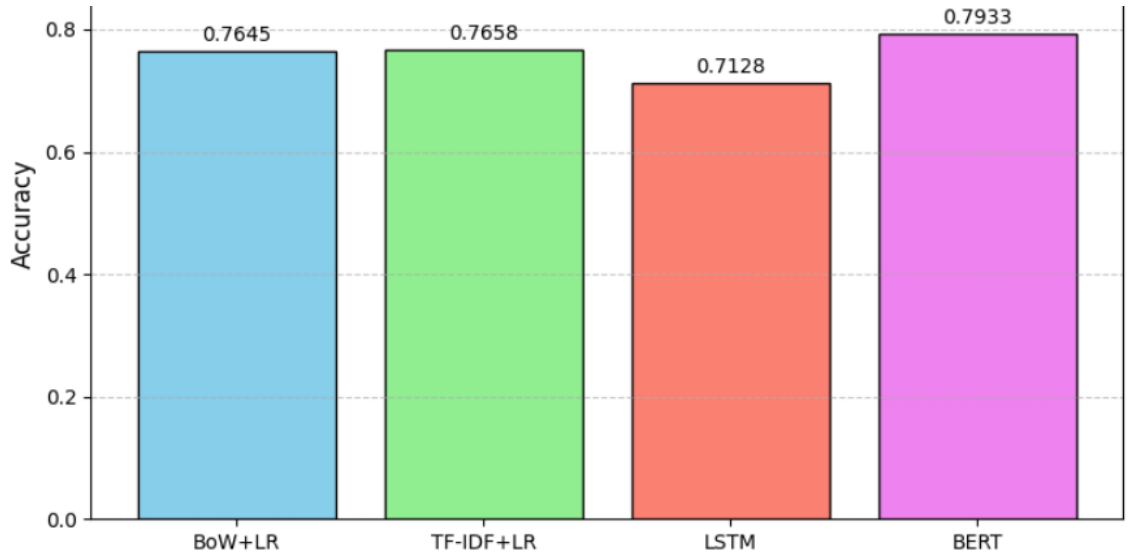

Figure 17: Comparison of the accuracy of all models.

The histogram shows that the BERT model yields the best result, with a score of 0.793. The BoW+LR and TF-IDF+LR models give similar values (0.765-0.766), while LSTM is significantly inferior (0.713). Testing confirmed the system's performance and its ability to classify various texts with varying accuracy depending on the model. The recommendations aim to enhance the accuracy, speed, and scalability of the system, making it promising for practical use in document analysis automation.

5. Conclusions

This study covers the following stages: software architecture design, classification algorithm development, and system testing on real data. The developed system is capable of processing text data using both traditional methods (BoW+LR and TF-IDF+LR) and deep learning models (LSTM, BERT).

The system allows users to select a data set, perform text preprocessing, perform classification using four approaches, and receive detailed reports on model performance. The main focus is on comparing the effectiveness of methods, which is implemented through the evaluation of metrics (accuracy, completeness, F1-measure) and the visualization of results.

Key achievements of the developed system:

1. A modular architecture has been created for easy adaptation to new data;
2. Classification with a maximum accuracy of 0.793 (BERT) has been implemented;
3. Comparison of models with visualization of results is provided;
4. Data for deep learning models (LSTM, BERT) is prepared;
5. Testing was carried out, which confirmed the system's operability.

Thus, the created system is a practical tool for automating document classification, which can be the basis for the further development of information systems in various industries. Its implementation is able to optimize the processing of large volumes of text. Development prospects include the integration of a web interface and improving models by optimizing them and using larger computing resources.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Ann, Document analysis – What, why and how. Adaptive US, 2023. URL: https://www.adaptiveus.com/blog/document-analysis/.

[2] H. Lypak, T. Lypak, N. Kunanets, Design of an information system based on machine learning for preservation and classification of documentary heritage artifacts. Herald of Khmelnytskyi National University. Technical Sciences 339(4) (2024) 176–182. doi:10.31891/2307-5732-2024-339-4-29 [in Ukrainian].

[3] O. Zhuchenko, A. Korotynskyi, A. Savula, Development of an automatic document processing system using generative adversarial neural networks, Electromechanical and Energy Saving Systems 58(2) 2022 50–55. doi:10.30929/2072-2052.2022.2.58.49-52 [in Ukrainian].

[4] I. Pasemko, Y. Fedonyuk, Automated analysis system for natural language texts using transformers, Information Systems and Networks, 17 (2025) 366–381. doi:10.23939/sisn2025.17.366 [in Ukrainian].

[5] V. Shkurko, A. Poliakov, Organization of software and neural network algorithms for machine analysis of textual data presented in natural language, Innovative Technologies and Scientific Solutions for Industries 2(32) (2025) 151–167. doi:10.30837/2522-9818.2025.2.151.

[6] S.İ. Omurca, et al., A document image classification system fusing deep and machine learning models, Applied Intelligence 53 (2023) 15295–15310. doi:10.1007/s10489-022-04306-5.

[7] M. Sambetbayeva, et al., Development of intelligent electronic document management system model based on machine learning methods, Eastern-European Journal of Enterprise Technologies 1(2) (2022) 68–76. doi:10.15587/1729-4061.2022.251689.
[8] S. Jiang, J. Hu, C.L. Magee, J. Luo. "Deep learning for technical document classification". IEEE Transactions on Engineering Management 71 (2024): 1163–1179. doi:10.1109/TEM.2022.3152216.
[9] T. Filimonova, O. Pursky, V. Babenko, A. Nechepourenko, V. Shvets, V. Gamaliy, Text sentiment analysis using different types of recurrent neural networks. In: Proceedings of the 5th International Conference on Image Processing and Capsule Networks, ICIPCN, Dhulikhel, Nepal, 2024, pp. 383–387. doi:10.1109/ICIPCN63822.2024.00068.
[10] Faizur Rashid, Suleiman M. A. Gargaare, Abdulkadir H. Aden, Afendi Abdi, Machine learning algorithms for document classification: Comparative analysis. International Journal of Advanced Computer Science and Applications (IJACSA) 13(4) (2022) 260-265. doi.org/10.14569/IJACSA.2022.0130430.
[11] S. Leviner, Types of document classification methods. Charactell, 2023. URL: https://www.charactell.com/resources/types-of-document-classification-methods/.
[12] D. Mwiti, Python bag of words model: A complete guide. DataCamp, 2024. URL: https://www.datacamp.com/tutorial/python-bag-of-words-model.
[13] TfidfVectorizer. Scikit-learn Documentation, 2025. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.
[14] LSTM layer. Keras Documentation, 2025. URL: https://keras.io/api/layers/recurrent_layers/lstm/.
[15] BERT. Hugging Face Transformers Documentation, 2025. URL: https://huggingface.co/docs/transformers/model_doc/bert.
[16] Y. HaCohen-Kerner, D. Miller, Y. Yigal, The influence of preprocessing on text classification using a bag-of-words representation, PLoS ONE, 15(5) 2020 e0232525. doi:10.1371/journal.pone.0232525.
[17] L. Stewart, Document analysis – How to analyze text data for research. ATLAS.ti Research Hub, 2025. URL: https://atlasti.com/research-hub/document-analysis.
[18] P. Kashyap, Understanding precision, recall, and F1 score metrics. Medium, 2024. URL: https://medium.com/@piyushkashyap045/understanding-precision-recall-and-f1-score-metrics-ea219b908093.
[19] R. Misra, R. News Category Dataset (Version 3) [Data set]. Kaggle 2022. URL: https://www.kaggle.com/datasets/rmisra/news-category-dataset.