Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
2025, vol. 30, no. 1, pp. 40–46
https://doi.org/10.2478/acss-2025-0005
https://content.sciendo.com

Hierarchical Text Classification: Fine-tuned GPT-2 vs BERT-BiLSTM

Djelloul Bouchiha[1*], Abdelghani Bouziane[2], Noureddine Doumi[3], Benamar Hamzaoui[4], Sofiane Boukli-Hacene[5]
[1,2,4]*University Centre of Naama, Naama, Algeria*
[3]*University of Saida, Saida, Algeria*
[5]*University of Sidi Bel Abbes, EEDIS Lab., Sidi Bel Abbes, Algeria*

***Abstract* – Hierarchical Text Classification (HTC) is a specialised task in natural language processing that involves categorising text into a hierarchical structure of classes. This approach is particularly valuable in several domains, such as document organisation, sentiment analysis, and information retrieval, where classification schemas naturally form hierarchical structures. In this paper, we propose and compare two deep learning-based models for HTC. The first model involves fine-tuning GPT-2, a large language model (LLM), specifically for hierarchical classification tasks. Fine-tuning adapts GPT-2's extensive pre-trained knowledge to the nuances of hierarchical classification. The second model leverages BERT for text pre-processing and encoding, followed by a BiLSTM layer for the classification process. Experimental results demonstrate that the fine-tuned GPT-2 model significantly outperforms the BERT-BiLSTM model in accuracy and F1 scores, underscoring the advantages of using advanced LLMs for hierarchical text classification.**

***Keywords* – BERT, BiLSTM, fine-tuned GPT-2, generative pre-trained transformer (GPT), hierarchical text classification (HTC), large language model (LLM).**

I. Introduction

Artificial Intelligence (AI) has transformed numerous industries by incorporating machine learning and deep learning methods to address complex challenges across a wide range of domains. Within AI, natural language processing (NLP) has emerged as a critical area focusing on the interaction between computers and human languages [1]. One of the key tasks in NLP is text classification, which involves categorising text into predefined classes [2]. Text classification has a wide range of applications, including document organisation, sentiment analysis, and spam detection.

Historically, text classification challenges have been addressed using machine learning methods, which involve training models on labelled datasets to identify patterns and produce predictions [3]. However, the emergence of deep learning has greatly enhanced the effectiveness of text classification models. Deep learning, a subset of machine learning, utilises neural networks (NNs) with several layers, hence the term "deep", to capture and model intricate patterns in data [4]. This ability to learn from huge amounts of data and capture intricate patterns has led to the development of sophisticated models, such as Bidirectional Encoder Representations from Transformers (BERT), Long Short-Term Memory (LSTM) networks, and Bidirectional LSTM (BiLSTM). These models have revolutionised NLP by providing robust solutions for various tasks, including text classification.

In recent years, large language models (LLMs), like GPT-2, have become highly effective tools in the field of NLP [5]. These models utilise deep learning methods to analyse and produce human-like text by being trained on extensive datasets. LLMs can be fine-tuned to address a range of problems, including text classification.

Hierarchical Text Classification (HTC) presents an additional layer of complexity compared to flat text classification [6]. While flat text classification assigns a single label to a document, HTC involves assigning multiple labels that follow a hierarchical structure. This added complexity arises from the need to accurately navigate and classify texts within a hierarchy, which requires models to understand both the content and the hierarchical relationships between labels.

In this paper, we propose and compare two deep learning-based models for HTC. The first approach leverages GPT-2, a prominent LLM, by fine-tuning it for HTC tasks. Fine-tuning adapts GPT-2's extensive pre-trained knowledge to the specific requirements of hierarchical classification. The second approach uses BERT for text pre-processing and encoding, followed by a BiLSTM layer for the classification process.

Our comparison demonstrates that fine-tuning GPT-2 is more efficient and effective in hierarchical text classification compared to the BERT-BiLSTM model. The results show that GPT-2 not only achieves higher accuracy but also performs better in terms of F1 scores, highlighting the potential of advanced LLMs in addressing the complexities of HTC.

The remainder of the paper is organised as follows: Section 2 covers related work, with an emphasis on text classification. Section 3 presents the fine-tuned GPT-2 based model, detailing the adaptation of GPT-2 for hierarchical text classification and

* Corresponding author's e-mail: bouchiha@cuniv-naama.dz
Article received 2024-11-07; accepted 2025-02-26

its system architecture. In Section 4, the proposed BERT-BiLSTM based model is introduced, highlighting how BERT is integrated with BiLSTM for hierarchical classification tasks. Section 5 offers a comparative study between the fine-tuned GPT-2 and BERT-BiLSTM models, presenting performance metrics and empirical results. Finally, Section 6 concludes the paper with insights drawn from the comparative analysis and discusses future research directions in HTC.

II. Related Work

Text classification (TC) is among the most extensively studied tasks in the NLP community [7]. Essentially, TC methods are supervised learning algorithms aimed at mapping texts to a predefined set of labels. Over the years, numerous surveys and reviews have covered a broad spectrum of techniques in NLP, from traditional approaches to state-of-the-art deep learning methods [2], [7]–[12].

As a subtask of TC, HTC shares many similarities with standard text classification methods. The primary distinction lies in the organisation of labels in a multi-level hierarchy, adding complexity to the classification process. Given the extensive range of HTC approaches, we offer a high-level overview of recent methodologies, building upon and extending the study by Zangari et al. [13] as follows:

- *Sequence-to-sequence (Seq2Seq)* methods reinterpret hierarchical text classification as a Seq2Seq task, where the input is the document (text) to categorise, and the output is a sequence of labels. This approach improves the modelling of local label hierarchies [14]. Target labels for each document are converted into a linear sequence, ordered through techniques such as breadth-first or depth-first search. During prediction, each step produces the next label according to the input document and the label predicted earlier, enabling the model to determine dependencies between labels. Various frameworks using this method have shown enhancements in label generation and more effective use of co-occurrence information [15], [16]. Many of these works highlight better label generation and the improved exploitation of co-occurrence data [17]–[21].
- *Graph-based* methods represent hierarchical structures with specialised encoders, facilitating improved use of shared information among labels. Both graph-based and tree-based models have shown promising results [22], [23]. Several approaches utilise graph convolutional neural networks (GCNs) to encode label hierarchies, often combined with text encoders to enrich the overall representation [24]. Hierarchy-aware and label-balanced models have also been suggested for HTC [25], alongside techniques aimed at enhancing label correlation and co-occurrence modelling [26]–[29]. However, some researchers, like Ning et al. [30], argue that graph-based methods might overlook the directional nature of tree hierarchies, proposing unidirectional message-passing constraints as a solution. Others have leveraged GCNs to boost feature sharing among labels at the same hierarchical level [31]. Although graph neural networks (GNNs) and GCNs are widely favoured, transformer-based models such as Graphormers have also been explored [32]. Furthermore, knowledge graphs (KGs) have been employed to create knowledge-aware embeddings, incorporating external data and improving generalizability across different tasks [33].
- *Tuning* methods for large models (LMs) are becoming more and more popular in NLP and have also yielded promising outcomes in hierarchical text classification. One such technique, prompt-tuning, seeks to bridge the gap between pre-training and fine-tuning by embedding inputs in natural language templates, effectively transforming the classification task into a masked language modelling task [34]–[36]. Another method, known as prefix-tuning, optimises and stores a smaller set of parameters compared to full fine-tuning, while still delivering similar performance levels [37].

In this paper, we implement two deep learning-based models for HTC. The BERT-BiLSTM model, a transformer-based approach, can be classified under the graph-based method, as it leverages transformers to encode text and the hierarchical structure. On the other hand, the fine-tuned GPT-2 model represents an LM tuning method, adapting a large language model to the hierarchical classification task.

III. Fine-tuned GPT-2 Based Model

This section presents the fine-tuned GPT-2 system and its architecture for hierarchical text classification.

A. GPT-2 and the Fine-Tuning Process

GPT-2 (Generative Pre-trained Transformer 2) is a large language model (LLM) created by OpenAI [38]. It is built on the Transformer architecture [39], which leverages self-attention mechanisms to efficiently handle long-range dependencies in textual data. GPT-2 employs a series of transformer decoder layers and is trained on extensive datasets to perform a range of NLP tasks, such as translation, text generation, and summarisation.

Fine-tuning GPT-2 consists of additional training on a specific task or dataset to customise the pre-trained model for a new context. In our case, the task is hierarchical text classification. Fine-tuning allows the model to retain the general language understanding from its pre-training phase while specialising in the nuances of hierarchical classification. This process typically involves adjusting hyperparameters, such as learning rate, number of epochs, and batch size, to enhance performance on the target task.

B. System Architecture

As illustrated in Fig. 1, the GPT-2 fine-tuning process for hierarchical text classification consists of several key components:

- Data layer: The dataset is loaded and split into training and test sets in a 70-30 ratio.
- Pre-processing layer: This includes:

  - Tokenization: Using the GPT-2 tokenizer to convert text into token IDs, ensuring that the padding token is defined.
  - Label encoding: Mapping hierarchical categories and super-categories to numerical labels.
- Model layer:
  - Model initialisation: Initialising two separate GPT-2 models, one for category classification and another for super-category classification, with each configured for the appropriate number of labels.
  - Training arguments: Setting the hyperparameters for training, which include the number of epochs, learning rate, weight decay, and batch size.
- Training and Evaluation layer:
  - Trainer: Managing the training and evaluation processes using the Trainer class, with data collators handling padding and batching.
  - Metrics calculation: Computing evaluation metrics such as accuracy and F1 scores for categories, super-categories, and overall hierarchical classification.

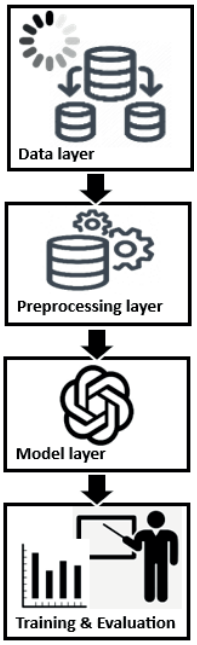

Fig. 1. Fine-tuned GPT-2 system architecture.

IV. BERT-BiLSTM Based Model

This section introduces the proposed BERT-BiLSTM based system and its architecture for hierarchical text classification.

A. BERT and BiLSTM

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model created by Google that utilises the transformer architecture to generate contextual embeddings [40]. Unlike traditional models, BERT considers the context of a word based on both its preceding and succeeding words, making it bidirectional. This bidirectional approach enhances the model's understanding of natural language, allowing it to achieve state-of-the-art results in various NLP tasks such as named entity recognition and sentiment analysis.

BiLSTM (Bidirectional Long Short-Term Memory) is an enhancement of the conventional LSTM model, aimed at addressing the challenges of standard recurrent neural networks (RNNs) in capturing long-term dependencies. LSTM units have gates that control the flow of information, permitting the network to either retain or discard information as needed over time [41]. BiLSTM further improves upon this by processing input data in both forward and backward directions, capturing context from both the left and right ends of the sequence [42]. This bidirectional processing provides a more comprehensive understanding of the text.

B. System Architecture

As illustrated in Fig. 2, the BERT-BiLSTM system architecture for hierarchical text classification comprises the following components:

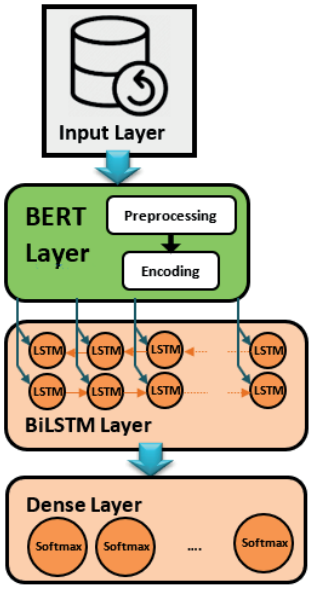

Fig. 2. BERT-BiLSTM system architecture.

- Input layer: The dataset is loaded and split into a 70 % training set and a 30 % test set.
- BERT embedding layer: We use a pre-trained BERT model to create contextual embeddings (encoding) for the input text sequences after an initial pre-processing step. These embeddings consist of dense vectors that reflect contextual information, with each token's embedding influenced by the entire sequence. This approach captures subtle language patterns and meanings.
- Bidirectional LSTM layer: Two LSTM layers analyse the BERT embeddings, one processes the text in the forward direction, while the other processes it backward. This bidirectional approach captures long-range dependencies and context from both directions, enhancing the model's understanding of the hierarchical structure and relationships within the input data.
- Dense layer: This layer assigns the text to predefined categories or super-categories using a softmax activation

function. The dense (fully connected) layer applies softmax to the aggregated output and generates a probability distribution over potential categories and super-categories, which facilitates hierarchical classification.

V. Comparative Study

To conduct our comparative study, we implemented two classifiers, Fine-tuned GPT-2 and BERT-BiLSTM, and made them publicly available on GitHub [2] for the AI and NLP communities. We also used a large hierarchical dataset structured into three levels. Additionally, we utilised a GPU-powered machine to run the classifiers. The following sections detail the experiments and results.

A. Fine-tuned GPT-2 Classifier

The implementation of the hierarchical text classifier using GPT-2 follows these key steps:

- Data preparation and Tokenization: The dataset is loaded and split into 70 % for training and 30 % for testing. The GPT-2 tokenizer is initialized, and the padding token is set to the end-of-sequence token.
- Dataset encoding and Model initialisation: Labels are encoded for both category and super-category classifications. Two GPT-2 models are initialised, one for category classification and the other for super-category classification.
- Training arguments: We need to define several arguments for training:
  - Epochs: Specify the total number of times the model processes the entire training dataset.
  - Batch size: The number of samples processed before updating the model's parameters.
  - Warmup steps: Gradual increase in learning rate over a set number of steps.
  - Weight decay: Regularization to prevent overfitting by applying a penalty to the weights.
  - Evaluation strategy: The model is evaluated at the end of each epoch to monitor its performance.
- Training and Evaluation: The models are trained with the specified parameters and evaluated on the test set. The evaluation metrics include category accuracy, category F1 score, super-category accuracy, super-category F1 score, and hierarchical F1 score.

The default parameters for the fine-tuned GPT-2 classifier, epochs = 3, batch size = 1, warmup steps = 500, weight decay = 0.01, and evaluation strategy = 'epoch', were chosen to optimise training while managing computational resources. Setting epochs to 3 allows the model to learn effectively without overfitting, while a batch size of 1 is selected due to GPT-2's high memory requirements, especially with longer sequences. Warmup steps of 500 help the model stabilise early in training by gradually increasing the learning rate, preventing sudden large updates. A weight decay of 0.01 is applied to avoid overfitting by slightly penalising large weights. The evaluation strategy of 'epoch' ensures that the model's performance is assessed after each epoch, allowing for adjustments if necessary. These settings aim to strike a balance between achieving strong model performance and maintaining efficient use of resources.

B. BERT-BiLSTM Classifier

The implementation of the BERT-BiLSTM hierarchical text classifier involves several stages:

- Data loading and splitting: The dataset is loaded and split into 70 % for training and 30 % for testing.
- Pre-processing and Encoding: BERT handles pre-processing tasks such as text cleaning and tokenization. The text is then encoded as a 768-dimensional vector.
- Parameter settings: Several parameters are set for training:
  - Epochs: Correspond to the total number of times the model processes the entire dataset.
  - Batch size: The number of samples processed in each iteration before updating the model's parameters.
  - Sequence length: Fixed length of input sequences during tokenization.
  - Hidden size: Dimensionality of the BERT embeddings and LSTM units.
  - Dense layer: Number of units and activation functions in the fully connected layers.
  - Attention masks: Binary masks indicating which tokens should be attended to during processing.
- Model Training and Evaluation: The models are trained with the specified parameters and evaluated on the test dataset. Evaluation metrics include category accuracy, category F1 score, super-category accuracy, super-category F1 score, and hierarchical F1 score.

The default parameters for the BERT-BiLSTM model, epochs = 3, batch size = 16, sequence length = 128, hidden size = 768, dense layer size = 768, and attention masks, were chosen to balance performance and computational efficiency. Setting epochs to 3 ensures sufficient training without risking overfitting, while a batch size of 16 effectively manages GPU memory constraints and maintains stable training. A sequence length of 128 captures the majority of relevant text context while fitting within typical GPU memory limits. The hidden size of 768 and the dense layer of the same size align with BERT's architecture, preserving model integrity and maximizing the benefits of its pre-trained weights. Attention masks are used to ensure that the model correctly focuses on actual tokens, ignoring padding, which helps maintain accuracy during training. These settings are well-established in practice for achieving reliable performance without excessive resource demands.

C. Dataset

For our experiments, we used a hierarchical text classification dataset built from Amazon product reviews [3], consisting of forty thousand texts. The dataset is organised into the following levels: six classes at level 1, sixty-four classes at

[2] https://github.com/khouloud-1/Fine-tunedGPT-2vsBERT-BiLSTM
[3] https://www.kaggle.com/datasets/kashnitsky/hierarchical-text-classification

level 2, and five hundred ten classes at level 3. The level 1 classes include grocery gourmet food, beauty, pet supplies, health personal care, baby products, and toys games. Before training the models, the dataset format was adapted from its original structure ("Text | Category_level1 | Category_level2 | Category_level3") to the format required by our models ("Text | Category | Super_category").

Selecting the optimal proportion for splitting the dataset into training and test sets is a challenge in classification problems. Several studies have addressed this issue. Bichri et al. found that using at least 70 % of the dataset for training yields the best performance [43]. Similarly, after conducting multiple experiments, Vrigazova recommended a 70/30 train/test split [44]. Based on these findings, the Amazon product reviews dataset was randomly divided into a 70/30 split for both models.

D. Evaluation Metrics

The following metrics are used to evaluate the fine-tuned GPT-2 and BERT-BiLSTM models in the context of HTC:

- Category Accuracy: Measures the proportion of correct predictions for category labels [45].

$$\text{Category Accuracy} = \frac{\text{Number of Correct Category Predictions}}{\text{Total Number of Predictions}}. \quad (1)$$

- Category F1 Score: Balances precision and recall for category classification by calculating the harmonic mean of the two metrics [46].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}}. \quad (2)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}. \quad (3)$$

$$\text{Category F1 Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}. \quad (4)$$

- Super-Category Accuracy: Measures the proportion of correct predictions for super-category labels [45].

$$\text{Super\_Category Accuracy} = \frac{\text{Number of Correct Super\_Category Predictions}}{\text{Total Number of Predictions}}. \quad (5)$$

- Super-Category F1 Score: Balances precision and recall for super-category classification by calculating the harmonic mean of the two metrics [46].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}}. \quad (6)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}. \quad (7)$$

$$\text{Super\_Category F1 Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}. \quad (8)$$

- Hierarchical F1 Score: Evaluates the correctness of predictions across both category and super-category levels by accounting for the hierarchical structure of the labels [47]. Let TP_H be the number of true positives considering the hierarchy, FP_H be the false positives, and FN_H be the false negatives.

$$\text{Hierarchical Precision} = \frac{TP_H}{TP_H+FP_H}. \quad (9)$$

$$\text{Hierarchical Recall} = \frac{TP_H}{TP_H+FN_H}. \quad (10)$$

$$\text{Hierarchical F1 Score} = 2 \times \frac{\text{Hierarchical Precision}\times\text{Hierarchical Recall}}{\text{Hierarchical Precision+Hierarchical Recall}}. \quad (11)$$

E. Results and Discussion

The classification experiments were performed on a GPU machine with the following specifications:

- GPU: MSI GeForce RTX 3060 GAMING 12G;
- Graphics Chipset: NVIDIA GeForce RTX 3060;
- Chipset Frequency: 1320 MHz;
- Boosted Frequency: 1777 MHz;
- PCI Express 4.0 16x bus;
- Video Memory Size: 12 GB;
- Video Memory Frequency: 15000 MHz;
- Memory Type: GDDR6;
- CUDA Cores: 3584 cores.

As illustrated in Fig. 3, the results indicate that the Fine-tuned GPT-2 model outperformed the BERT-BiLSTM model in every evaluation metric. Notably, GPT-2 achieved a category accuracy of 58.1 % and a super-category accuracy of 67.4 %, which were both higher than the BERT-BiLSTM model's respective accuracies of 45.2 % and 62.4 %.

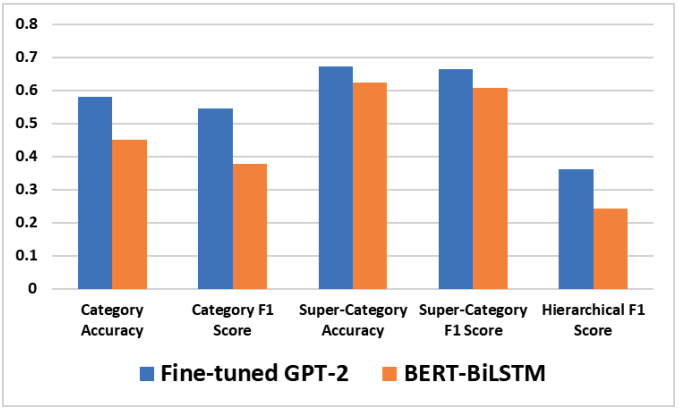

Fig. 3. Evaluation results of HTC with Fine-tuned GPT-2 and BERT-BiLSTM models.

Additionally, the F1 scores, which provide a balanced view of precision and recall, further emphasise GPT-2's superiority. The hierarchical F1 score, which is particularly important for hierarchical text classification tasks, was significantly higher for GPT-2 (0.3620) compared to BERT-BiLSTM (0.2431).

These results suggest that GPT-2's generative capabilities, fine-tuned for this specific task, are more effective in capturing the hierarchical relationships within the text. BERT-BiLSTM, while still a strong baseline, struggled to match the performance

of GPT-2 in this context. Future research could explore the integration of additional features or further hyperparameter optimisation to enhance the BERT-BiLSTM model's performance.

VI. Conclusion and Perspectives

In this study, we proposed and compared two deep learning models, Fine-tuned GPT-2 and BERT-BiLSTM, for the hierarchical text classification. Through comprehensive experiments using a large dataset of Amazon product reviews, we demonstrated that Fine-tuned GPT-2 outperformed BERT-BiLSTM across most evaluation metrics, particularly in Accuracy and Hierarchical F1 Score. These results highlight the effectiveness of generative pre-trained transformers in handling hierarchical text classification tasks, where contextual understanding at multiple levels is crucial.

However, both models present opportunities for further enhancement. For instance, while Fine-tuned GPT-2 delivered superior performance, its complexity and training time are notable challenges. On the other hand, BERT-BiLSTM, although less accurate, could benefit from further optimisation and parameter adjustments.

There are several promising ways to enhance the performance of both models. Adjusting key hyperparameters, such as the batch size, number of epochs, and learning rate, could potentially lead to better model convergence and improved classification accuracy. Experimenting with different tokenization strategies, attention mechanisms, and model architectures could further refine the results. Exploring transfer learning and ensembling techniques may also enhance the model generalization ability across various datasets and domains.

Moreover, expanding the dataset to include more diverse texts and labels or incorporating additional hierarchical levels could test the robustness of the proposed approaches. We also envision the possibility of applying these methods to other languages and domains, contributing to the growing body of research in hierarchical text classification.

References

[1] J. Eisenstein, *Introduction to Natural Language Processing*. The MIT Press, 2019.

[2] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, Apr. 2019, Art. no. 150. https://doi.org/10.3390/info10040150

[3] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, "Twenty years of machine-learning-based text classification: A systematic review," *Algorithms*, vol. 16, no. 5, Apr. 2023, Art. no. 236. https://doi.org/10.3390/a16050236

[4] J. M. Patel, "Deep learning: Concepts, architectures, workflow, applications and future directions," *International Journal for Multidisciplinary Research*, vol. 5, no. 6, pp. 1–7, Nov.–Dec. 2023. https://doi.org/10.36948/ijfmr.2023.v05i06.11497

[5] D. H. Hagos, R. Battle, and D. B. Rawat, "Recent advances in generative AI and large language models: Current status, challenges, and perspectives," *arXiv:2407.14962*, Aug. 2024. https://doi.org/10.48550/arXiv.2407.14962

[6] A. Zangari, M. Marcuzzo, M. Schiavinato, M. Rizzo, A. Gasparetto, and A. Albarelli, "Hierarchical text classification: A review of current research," *Electronics*, vol. 13, no. 7, Mar. 2024, Art. no. 1199. https://doi.org/10.3390/electronics13071199

[7] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," *Information*, vol. 13, no. 2, Feb. 2022, Art. no. 83. https://doi.org/10.3390/info13020083

[8] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From traditional to deep learning," *ACM Trans. Intell. Syst. Technol*., vol. 13, no. 2, Apr. 2022, Art. no. 31. https://doi.org/10.1145/3495162

[9] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: A comprehensive review," *ACM Comput. Surv*., vol. 54, no. 3, Apr. 2021, Art. no. 62. https://doi.org/10.1145/3439726

[10] A. Gasparetto, A. Zangari, M. Marcuzzo, and A. Albarelli, "A survey on text classification: Practical perspectives on the Italian language," *PLoS ONE*, vol. 17, no. 7, Jul. 2022, Art. no. e0270904. https://doi.org/10.1371/journal.pone.0270904

[11] J. Fields, K. Chovanec, and P. Madiraju, "A survey of text classification with transformers: How wide? How large? How long? How accurate? How expensive? How safe?," *IEEE Access*, vol. 12, pp. 6518–6531, Jan. 2024. https://doi.org/10.1109/ACCESS.2024.3349952

[12] S. Bhawsar, S. Dubey, S. Kushwaha, and S. Sharma, "Text classification using deep learning: A survey," in *Proceedings of International Conference on Computational Intelligence*, Singapore, 2023, pp. 205–216. https://doi.org/10.1007/978-981-19-2126-1_16

[13] A. Zangari, M. Marcuzzo, M. Rizzo, L. Giudice, A. Albarelli, and A. Gasparetto, "Hierarchical text classification and its foundations: A review of current research," *Electronics*, vol. 13, no. 7, Mar. 2024, Art. no. 1199. https://doi.org/10.3390/electronics13071199

[14] Z. Yang and G. Liu, "Hierarchical sequence-to-sequence model for multi-label text classification," *IEEE Access*, vol. 7, pp. 153012–153020, Oct. 2019. https://doi.org/10.1109/ACCESS.2019.2948855

[15] W. Zhao, H. Gao, S. Chen, and N. Wang, "Generative multi-task learning for text classification," *IEEE Access*, vol. 8, pp. 86380–86387, May 2020. https://doi.org/10.1109/ACCESS.2020.2991337

[16] K. Rivas Rojas, G. Bustamante, A. Oncevay, and M. A. Sobrevilla Cabezudo, "Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 2252–2257. https://doi.org/10.18653/v1/2020.acl-main.205

[17] J. Risch, S. Garda, and R. Krestel, "Hierarchical document classification as a sequence generation task," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, China, Aug. 2020, pp. 147–155. https://doi.org/10.1145/3383583.3398538

[18] J. Yan, P. Li, H. Chen, J. Zheng, and Q. Ma, "Does the order matter? A random generative way to learn label hierarchy for hierarchical text classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 276–285, Nov. 2024. https://doi.org/10.1109/TASLP.2023.3329374

[19] J. Kwon, H. Kamigaito, Y.-I. Song, and M. Okumura, "Hierarchical label generation for text classification," in *Proceedings of the Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia, May 2023, pp. 625–632. https://doi.org/10.18653/v1/2023.findings-eacl.46

[20] F. Torba, C. Gravier, C. Laclau, A. Kammoun, and J. Subercaze, "A study on hierarchical text classification as a Seq2seq task," in *Advances in Information Retrieval*. ECIR 2024, Cham, Mar. 2024, pp. 287–296. https://doi.org/10.1007/978-3-031-56063-7_20

[21] J. Zhang, Y. Li, F. Shen, Y. He, H. Tan, and Y. He, "Hierarchical text classification with multi-label contrastive learning and KNN," *Neurocomputing*, vol. 577, Apr. 2024, Art. no. 127323. https://doi.org/10.1016/j.neucom.2024.127323

[22] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, and G. Liu, "Hierarchy-aware global model for hierarchical text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 1106–1117. https://doi.org/10.18653/v1/2020.acl-main.104

[23] H. Liu, X. Huang, and X. Liu, "Improve label embedding quality through global sensitive GAT for hierarchical text classification," *Expert Systems with Applications*, vol. 238, Mar. 2024, Art. no. 122267. https://doi.org/10.1016/j.eswa.2023.122267

[24] H. Chen, Q. Ma, Z. Lin, and J. Yan, "Hierarchy-aware label semantics matching network for hierarchical text classification," in *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. J. Conf. on Nat. Language Processing, ACL/IJCNLP 2021*, Aug. 2021, pp. 4370–4379. https://doi.org/10.18653/v1/2021.acl-long.337

[25] J. Zhang, Y. Li, F. Shen, C. Xia, H. Tan, and Y. He, "Hierarchy-aware and label balanced model for hierarchical text classification," *Knowledge-Based Systems*, vol. 300, Sep. 2024, Art. no. 112153. https://doi.org/10.1016/j.knosys.2024.112153
[26] A. Pal, M. Selvakumar, and M. Sankarasubbu, "MAGNET: Multi-label text classification using attention-based graph neural network," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020)*, Valletta, Malta, 2020, pp. 494–505. https://doi.org/10.5220/0008940304940505
[27] R. Zhao, X. Wei, C. Ding, and Y. Chen, "Hierarchical multi-label text classification: Self-adaption semantic awareness network integrating text topic and label level information," in *Proceedings of the Knowledge Science, Engineering and Management*, Hangzhou, China, Aug. 2021, pp. 406–418. https://doi.org/10.1007/978-3-030-82147-0_33
[28] J. Chen, S. Zhao, F. Lu, F. Liu, and Y. Zhang, "Research on patent classification based on hierarchical label semantics," in *2022 3rd International Conference on Education, Knowledge and Information Management (ICEKIM)*, Harbin, China, Aug. 2022, pp. 1025–1032. https://doi.org/10.1109/ICEKIM55072.2022.00223
[29] Z. Yao, H. Chai, J. Cui, S. Tang, and Q. Liao, "HITSZQ at SemEval-2023 Task 10: Category-aware sexism detection model with self-training strategy," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada, Jul. 2023, pp. 934–940. https://doi.org/10.18653/v1/2023.semeval-1.129
[30] B. Ning, D. Zhao, X. Zhang, C. Wang, and S. Song, "UMP-MG: A uni-directed message-passing multi-label generation model for hierarchical text classification," *Data Science and Engineering*, vol. 8, pp. 112–123, Jun. 2023. https://doi.org/10.1007/s41019-023-00210-1
[31] J. Song, F. Wang, and Y. Yang, "Peer-label assisted hierarchical text classification," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada, Jul. 2023, pp. 3747–3758. https://doi.org/10.18653/v1/2023.acl-long.207
[32] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, May 2022, pp. 7109–7119. https://doi.org/10.18653/v1/2022.acl-long.491
[33] Y. Liu, K. Zhang, Z. Huang, K. Wang, Y. Zhang, Q. Liu, and E. Chen, "Enhancing hierarchical text classification through knowledge graph integration," in *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, ON, Canada, Jul. 2023, pp. 5797–5810. https://doi.org/10.18653/v1/2023.findings-acl.358
[34] Z. Wang, P. Wang, T. Liu, Y. Cao, Z. Sui, and H. Wang, "HPT: Hierarchy-aware prompt tuning for hierarchical text classification," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 3740–3751. https://doi.org/10.18653/v1/2022.emnlp-main.246
[35] K. Ji, Y. Lian, J. Gao, and B. Wang, "Hierarchical verbalizer for few-shot hierarchical text classification," in *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, ON, Canada, Jul. 2023, pp. 2918–2933. https://doi.org/10.18653/v1/2023.acl-long.164
[36] Y. Zhang, R. Yang, X. Xu, R. Li, J. Xiao, J. Shen, and J. Han, "TELEClass: Taxonomy enrichment and LLM-enhanced hierarchical text classification with minimal supervision," *arXiv:2403.00165*, 2024. https://arxiv.org/pdf/2403.00165
[37] L. Chen, H. Chou, and X. Zhu, "Developing prefix-tuning models for hierarchical text classification," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Abu Dhabi, UAE, Dec. 2022, pp. 390–397. https://doi.org/10.18653/v1/2022.emnlp-industry.39
[38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, 2019, Art. no. 9.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017. https://arxiv.org/pdf/1706.03762
[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long and Short Papers), Minneapolis, Minnesota, 2019, pp. 4171–4186.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. https://doi.org/10.1162/neco.1997.9.8.1735
[42] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, Nov. 1997. https://doi.org/10.1109/78.650093
[43] H. Bichri, A. Chergui, and M. Hain, "Investigating the impact of train/test split ratio on the performance of pre-trained models with custom datasets," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 2, 2024. https://doi.org/10.14569/IJACSA.2024.0150235
[44] B. Vrigazova, "The proportion for splitting data into training and test set for the bootstrap in classification problems," *Business Systems Research Journal*, vol. 12, no. 1, pp. 228–242, May 2021. https://doi.org/10.2478/bsrj-2021-0015
[45] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *International Journal of Machine Learning Technology*, vol. 2, no. 1, pp. 37–63, Jan. 2011. https://doi.org/10.9735/2229-3981
[46] P. Christen, D. J. Hand, and N. Kirielle, "A review of the F-measure: Its history, properties, criticism, and alternatives," *ACM Comput. Surv.*, vol. 56, no. 3, Oct. 2023, Art. no. 73. https://doi.org/10.1145/3606367
[47] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *Data Mining and Knowledge Discovery*, vol. 29, pp. 820–865, May 2015. https://doi.org/10.1007/s10618-014-0382-x

Djelloul Bouchiha received his Engineer degree in Computer Science from Sidi Bel Abbes University, Algeria, in 2002, and *M. Sc*. in Computer Science from Sidi Bel Abbes University, Algeria, in 2005, and Ph. D. in 2011. Between 2005 and 2010, he joined the Department of Computer Science, Saida University, Algeria, as a Lecturer. He became an Associate Professor in September 2013 at the University Center of Naama, Algeria. Since January 2018, he is a Full Professor. His research interests include semantic web, software engineering, knowledge management, natural language processing, artificial intelligence, and machine learning. ORCID iD: https://orcid.org/0000-0001-7314-4329

Abdelghani Bouziane is an Associate Professor in Computer Science at the University Centre of Naama. Specialising in artificial intelligence (AI), machine learning, deep learning, and natural language processing, he has made several contributions to the field. He earned his Ph.D. in Computer Science from the University of Sidi Bel Abbès, Algeria, in 2019. Since 2012, he has been at the forefront of integrating AI technologies into research and teaching. At the University Centre of Naama, he serves as the Head of Field at the Computer Science Department and Chief of Doctoral Training in AI. His research primarily focuses on artificial intelligence and natural language processing.
ORCID iD: https://orcid.org/0000-0002-8583-3858

Noureddine Doumi is currently an Assistant Professor at the Computer Science Department of the University of Saida. He received his Ph.D. in Computer Science from the University of Sidi-Bel-Abbes in 2017. He is a member of EEDIS lab in UDL-SBA and an active member as a developer in Unitex/GramLab project at the University of Paris-Est Marne-La-Vallée. His research interest includes Arabic NLP, linguistic resources, finite-state machines and machine learning.
ORCID iD: https://orcid.org/0000-0002-3108-4732

Benamar Hamzaoui received his Engineer degree in Computer Science from Sidi Bel Abbes University, Algeria, in 1997, and a Master's degree in Computer Science from the same university in 2022. Since 2023, he has been pursuing a Ph.D. in Computer Science focusing on Artificial Intelligence. His research interests include image processing, databases and database management systems (DBMS), artificial intelligence, machine learning, deep learning, and natural language processing (NLP).
ORCID iD: https://orcid.org/0009-0001-4821-7343

Sofiane Boukli-Hacene is a Full Professor at the Computer Science Department of Djillali Liabes University (U.D.L) of Sidi Bel Abbes, Algeria. He received an engineering degree (first class honours) from U.D.L in 2002, the *M. Sc.* degree from Al Al Bayt University at Mafraq, Jordan in 2005, the Ph.D. degree from U.D.L in 2012, and the habilitation to supervise research (HDR) in 2014. He is the Head of the Advanced Networks Research Team and Evolutionary Engineering and Distributed Information Systems Laboratory at the U.D.L. He has supervised a number of PhD theses, and he is a member of several scientific committees for conferences and journal review boards. His research interests lie in networking, including wireless ad-hoc, sensor networks, vehicular networks, network security, QoS, and pervasive computing.
ORCID iD: https://orcid.org/0000-0002-9760-3806