Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 03, September 2025.

ENHANCING FEW-SHOT LEARNING PERFORMANCE WITH BOOSTING ON TRANSFORMERS: EXPERIMENTS ON SENTIMENT ANALYSIS TASKS

Lenh Phan Cong Pham and Huan Thai Phong

(Received: 23-Dec.-2024, Revised: 11-Jun.-2025, Accepted: 4-Jul.-2025)

ABSTRACT

This study addresses challenges in sentiment analysis for low-resource educational contexts by proposing a framework that integrates Few-Shot Learning (FSL) with Transformer-based ensemble models and boosting techniques. Sentiment analysis of student feedback is crucial for improving teaching quality, yet traditional methods struggle with data scarcity and computational inefficiency. The proposed framework leverages self-attention mechanisms in Transformers and combines models through Gradient Boosting to enhance performance and generalization with minimal labeled data. Evaluated on the UIT-VSFC dataset, comprising Vietnamese student feedback, the framework achieved superior F1-scores in sentiment and topic-classification tasks, outperforming individual models. Results demonstrate the potential of the proposed framework for extracting actionable insights to enhance educational experiences. Despite its effectiveness, the approach faces limitations, such as reliance on pre-trained models and computational complexity. Future work could optimize lightweight models and explore applications in other domains, like healthcare and finance.

KEYWORDS

Few-shot learning, Boosting, Transformer models, Sentiment analysis.

1. INTRODUCTION

In natural language processing (NLP), sentiment analysis, also referred to as opinion mining, is a method used for evaluating the emotional state of a given text [1]. This technique has become a valuable tool for extracting user opinions from product and service reviews, providing businesses with actionable insights to improve their offerings [2]. Student feedback is essential for assessing learning-management systems, instructional strategies and course material in the educational setting [3]. To facilitate efficient analysis, these feedback responses, which are frequently in the form of text, need to be pre-processed using NLP techniques as feature extraction and selection [4].

The initial step in sentiment analysis involves labeling text with emotional categories, like positive, negative, or neutral, reflecting students' feelings about the courses and services provided [5]. However, the manual annotation process can be time-consuming and require substantial resources, as well as an understanding of educational content. This challenge has been addressed through automated methods powered by AI and machine learning [6]. With its ability to process and analyze vast amounts of student input, artificial intelligence (AI) greatly improves the precision and effectiveness of sentiment categorization [7]. Even when feedback is unlabeled, machine learning, deep learning and transformer models are very good at using attention processes to identify students' feelings [8].

In the age of online and blended learning, where emotional cues may be harder to discern, leveraging sentiment-analysis tools becomes essential for extracting meaningful insights from textual data [9]. Furthermore, various machine-learning algorithms, such as Naive Bayes, Support Vector Machines (SVMs) and lexicon-based methods, have been used to analyze sentiments in student feedback, demonstrating their effectiveness in processing and interpreting these responses [10]–[12]. With these advancements, sentiment analysis not only contributes to enhancing teaching quality, but also provides valuable insights into the experiences and perspectives of students in the educational process.

Traditional supervised-learning approaches have been extensively applied in sentiment analysis, yet they are constrained by inherent limitations. One major challenge arises in scenarios with limited labeled training data, where traditional machine-learning models often suffer from overfitting, rendering them unable to generalize effectively to unseen data [13]. This limitation is particularly problematic in

P. C. P. Lenh (Corresponding Author) and T. P. Huan are with Faculty of Artificial Intelligence, FPT University, Can Tho, Vietnam. Emails: Lenhppcce180059@fpt.edu.vn and huantpce180685@fpt.edu.vn

sentiment analysis, where diverse and complex text patterns demand robust generalization. Moreover, while humans can intuitively generalize concepts with minimal exposure or partial information, machine-learning models struggle to replicate this ability [14]. As a result, traditional methods falter in low-data settings, leaving critical gaps in performance and scalability.

Previously, sentiment analysis has depended on supervised techniques that handle issues, like lexical variety and long-distance interdependence, present in textual data. To capture these relationships, sequence models such as RNNs and LSTM networks, have been frequently employed. While these models can encode complex relationships within text, their serialized processing makes them computationally inefficient and limits their scalability, especially in real-world applications. Through the application of parallelized processing, Transformer models, on the other hand, have transformed sentiment analysis and greatly increased computational effectiveness while maintaining the capacity to identify long-distance relationships. Their self-attention mechanisms allow for a more comprehensive understanding of text structure and semantics, making them well-suited for sentiment analysis. However, these models often require large amounts of labeled data to perform effectively, which poses a challenge in resource-constrained environments.

To address these challenges of data scarcity and computational inefficiency, Few-Shot Learning (FSL) has emerged as a promising solution. FSL enables models to generalize effectively from only a few labeled examples, mimicking human-like learning. However, traditional supervised methods still face limitations in terms of overfitting and dependency on large datasets. To overcome these issues, integrating ensemble learning with Transformer architecture and FSL offers a novel approach. By combining multiple Transformer models trained with few-shot data, ensemble learning can improve generalization and robustness, mitigating the risks of overfitting. The hybrid approach leverages the computational efficiency of Transformers, the contextual power of self-attention mechanisms and the scalability of FSL, offering a more effective and resource-efficient framework for sentiment analysis in real-world applications.

While traditional sentiment-analysis approaches have demonstrated strong performance on large-scale datasets, their applicability is limited in low-resource educational environments, where collecting and annotating large volumes of labeled data are often impractical due to time, budgetary and expertise constraints. Deep learning and transformer-based techniques have achieved promising results in educational contexts, such as analyzing course feedback or evaluating learning-management systems [60–62]. However, these approaches are highly dependent on the availability of comprehensively labeled datasets, which poses a significant barrier in many real-world educational scenarios, particularly in under-resourced institutions or less-documented languages. Moreover, existing research has paid limited attention to the use of boosting strategies for ensembling Transformer-based models in educational sentiment analysis. Most prior studies, such as [63] and [64], have focused on combining traditional deep-learning models and basic machine-learning techniques rather than leveraging the potential diversity and complementary strengths of multiple Transformer architectures. This reflects a research gap in exploring ensemble-learning techniques, particularly boosting, in conjunction with modern pre-trained language models for low-resource educational contexts.

To address the critical challenge of data scarcity in analyzing student feedback, particularly for under-resourced languages, like Vietnamese, within educational settings, this paper proposes a novel approach. We investigate the synergistic integration of Few-Shot Learning (FSL) with boosting-enhanced Transformer-based ensemble models. While FSL addresses the limited data and Transformers offer powerful text representation, the strategic application of boosting techniques over an ensemble of such FSL-trained Transformers is a relatively unexplored configuration aimed at maximizing performance and robustness specifically for this low-resource niche.

The purpose of the research includes:

- To rigorously assess the viability and effectiveness of integrating FSL with boosted Transformer ensembles for sentiment analysis specifically on scarce Vietnamese student-feedback data, thereby demonstrating a practical solution for low-resource educational contexts.
- To explore and apply boosting methods to combine model predictions and evaluate the effectiveness of ensemble techniques in improving accuracy and prediction performance for sentiment and topic

classification tasks.

- To develop and provide a high-performance model for student sentiment analysis, particularly suited for small datasets, to support research and enhance sentiment-analysis methods in the educational context.
- To evaluate the proposed model on an additional sentiment-analysis dataset from a different domain to ensure the model's robustness and generalizability across various contexts, thereby supporting its applicability in broader sentiment-analysis tasks beyond the educational setting.

2. Related Work

2.1 Contrastive Learning in Sentiment Analysis

The primary objective of contrastive learning (CL), a self-supervised machine-learning technique, is to develop representations through the comparison of various data samples. More specifically, CL learns to push negative pairings farther apart and bring positive pairs closer together in the representation space. In order to decrease dimensionality and enhance classification and recognition performance, CL was presented as a technique that involves learning an invariant mapping [15]. With a momentum encoder that continuously updates negative samples, it was shown how important the quantity of negative samples is to improving representation learning [16]. Constructing effective positive pairs was highlighted as a critical factor in learning high-quality representations in CL [17].

Contrastive learning has shown itself to be an effective technique in sentiment-analysis applications. Supervised CL has been directly used in a number of research studies [18]-[20] to align sentiment representations with corresponding sentiment labels in order to develop fine-grained sentiment representations. In order to promote more efficient sentiment-analysis learning, supervised CL creates positive pairings based on labels, where samples with the same label are regarded as positive pairs and samples with different labels are regarded as negative pairs [21]. Additionally, to improve the accuracy and resilience of sentiment-analysis models, multi-aspect samples for CL were created using an in-domain generator and a cross-channel data-augmentation technique [22]. In order to enhance sentiment-analysis performance, cross-lingual contrastive learning also employed token-level and sentence-level data-augmentation techniques in addition to sentiment identifying [23].

2.2 Boosting

Boosting is a method of machine learning that combines weak learners in an ensemble style to turn them into a strong classifier. Its main goal is to minimize bias, which aids in the improvement of highly biased models. Combining the outcomes of each iteration using a weighted vote for classification or a weighted sum for regression yields the final output of boosting [24].

2.2.1 AdaBoost

Adaptive boosting is a powerful boosting algorithm introduced by [25], designed to combine weak learners, typically decision stumps (decision trees with a single split), into a strong classifier. It is widely regarded as one of the most robust machine-learning algorithms, with AdaBoost.M1 being a notable implementation for binary-classification tasks [26]. AdaBoost requires little hyper-parameter tuning and is simple to deploy [27]. To create the strong classifier, the several base learners are added one after the other and weighted [28]. The learning process involves iteratively training base classifiers, updating sample weights based on their classification performance and prioritizing misclassified samples in subsequent iterations. Initially, all samples are assigned equal weights:

$$\mathrm{D}_1(i) = \frac{1}{m}, \qquad i = 1,2, \ldots, m.$$

The weights are then updated after each iteration using the formula:

$$\mathrm{D}_{t+1}(i) = \frac{\mathrm{D}_t(i)}{Z_t} \exp(-\mathrm{\alpha}_t \mathrm{y}_i \mathrm{h}_t(\mathrm{x}_i).$$

Here, the importance of each base classifier is quantified as:

$$\mathrm{\alpha}_t = \frac{1}{2} \ln(\frac{1 - \epsilon_t}{\epsilon_t})$$

where ϵ_t is the error rate of the base classifier. After $\mathrm{T}_{iterations}$, the final strong classifier is computed as:

$$\mathrm{H(x)} = \mathrm{sign}(\sum_{t=1}^{T} \alpha_t \mathrm{h}_t(\mathrm{x})$$

This approach ensures a weighted combination of base classifiers to optimize performance. AdaBoost's adaptability and sequential focus on hard-to-classify samples make it highly effective for diverse applications.

2.2.2 Gradient Boosting

A popular machine-learning technique, called gradient boosting, iteratively combines weaker base learners, usually decision trees, to create a powerful prediction model. Because it uses decision trees as essential building elements, it is frequently referred to as Gradient Boosted Decision Tree (GBDT). [29] was the first reference to describe the concept, demonstrating that boosting can be seen as an optimization problem that aims to achieve a certain loss function.

An advanced version of this approach was later developed [30], focusing on sequentially training models to construct a robust ensemble classifier. Unlike other boosting methods, the key idea in Gradient Boosting is to design base learners that align with the negative gradient of the loss function for the overall ensemble [31].

For a given training dataset $S = \{(\mathrm{x}_i, y_i\}_{i=1}^{N}$, the goal of Gradient Boosting is to approximate a function $\mathrm{F}^*(\mathrm{x})$ that predicts the response variable y based on input features x, by minimizing a pre-defined loss function L(y, F(x)). This approximation is achieved iteratively by creating an additive model expressed as:

$$\mathrm{F_m(x)} = \mathrm{F_{m-1}(x)} + \rho_\mathrm{m}\mathrm{h_m(x)}$$

Here:

- $\mathrm{F_m(x)}$: The prediction at iteration m.
- $\mathrm{F_{m-1}(x)}$: The prediction from the previous iteration.
- $\rho\mathrm{m}$: The weight of the mth learner.
- $\mathrm{h_m(x)}$: The mth base learner, typically a decision tree.

The initial model, $\mathrm{F_0(x)}$, is determined by minimizing the loss across all samples:

$$F_0(x) = \arg\min_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha)$$

In subsequent iterations, each new learner $h_m(x)$ is trained to minimize the error of the current model:

$$h_m(x) = \arg\min_{h} \sum_{i=1}^{N} L\big(y_i, F_{m-1}(x_i) + \rho h(x_i)\big)$$

A critical aspect of this process involves computing pseudo residuals, which represent the gradients of the loss function with respect to the model's predictions. These are calculated as:

$$r_{mi} = \left[\frac{\partial L\big(y_i, F(x)\big)}{\partial F(x)}\right] \quad F(x) = F_{m-1}(x)$$

The optimal weight ρ_m is subsequently obtained through a line-search procedure.

To mitigate overfitting, the algorithm applies shrinkage, scaling the contribution of each step by a learning rate y (commonly set to 0.1):

$$F_m(x) = F_{m-1}(x) + v\rho_m h_m(x)$$

Gradient boosting stands out for its ability to uncover intricate patterns in data by systematically addressing errors in previous iterations. However, it is susceptible to overfitting, especially with noisy datasets, if regularization techniques are not adequately employed [31 - 32]. Despite this, it remains a powerful choice, particularly for small datasets [33].

2.2.3 XGBoost

Extreme Gradient Boosting, or XGBoost, is a decision tree-based ensemble technique that uses the gradient-boosting framework and is incredibly effective and scalable. Because of its excellent accuracy in both classification and regression tasks, it has become more well-known. After winning many Kaggle tournaments, XGBoost has emerged as a major force in machine learning in recent years. Originally developed by [34], XGBoost introduces several enhancements over traditional gradient-boosting algorithms. A key feature of XGBoost is the incorporation of a regularization term in its loss function, which helps prevent overfitting [35].

The regularized loss function used in XGBoost is defined as:

$$L_M(F(x_i)) = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m)$$

where $L(y_i, F(x_i))$ measures the error between the predicted and actual values and $\Omega(h_m)$ represents the regularization term. The regularization term is expressed as:

$$\Omega(h) = \gamma^T + \frac{1}{2}\lambda|\omega|^2$$

In this expression, γ regulates the complexity of the trees, T is the number of tree leaves, λ serves as a penalty parameter and ω corresponds to the outputs from the leaf nodes.

Unlike standard gradient boosting, which uses first-order derivatives, XGBoost improves upon this by using a second-order Taylor approximation to optimize the loss function more effectively. The revised form of the loss function is:

$$L_M \approx \sum_{i=1}^{n} \left[g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(h_m)$$

where g_i and h_i represent the first and second derivatives of the loss function, respectively. The total loss is computed by summing the contributions from each leaf node, as described by:

$$L_M = \sum_{j=1}^{T} \sum_{i \in I_j} g_i \omega_j + \frac{1}{2} \sum_{i \in I_j} h_i + \lambda \omega_j^2 + \gamma T$$

The objective function is approximated quadratically as a result of this modification to the optimization process. Furthermore, according to [36], the regularization term makes sure that XGBoost is immune to overfitting. In order to prevent overfitting, XGBoost uses parameters, like tree depth, learning rate and sub-sampling, just like conventional gradient boosting.

One of the key advantages of XGBoost is its ability to handle minimal feature engineering, including dealing with missing values, data normalization and feature scaling. Furthermore, XGBoost can output feature importance, making it easier to understand the significance of different input features and perform feature selection. It can handle big datasets effectively, is quicker than the majority of machine-learning algorithms and frequently performs better than other models. This has made XGBoost a popular choice, particularly in Kaggle competitions. However, a disadvantage is that it has many hyper-parameters, which can make the model-tuning process quite complex [37]-[38].

2.3 Base Transformer Models for Ensemble Learning Boosting

The Transformer, introduced by [39], was designed to overcome the limitations of RNNs and traditional encoder-decoder architectures. By replacing RNNs with attention mechanisms, it enables efficient long-term memory handling. With feed-forward layers, residual connections and normalization layers combined with multi-head attention layers, the model concentrates on every token from the past. With attention weights derived from the encoder hidden states (K) and decoder state (Q), the attention mechanism aids the model in focusing on pertinent information depending on the current input. These weights are generated by an alignment function and distribution function, such as SoftMax, to enhance processing efficiency. Self-attention further enables the model to link positions within a single sequence to form comprehensive representations. Table 1 summarizes the transformer models experimented with in this study.

Table 1. Base models for boosting in transformer-based architectures.

Type	**Model**	**Supported Language**	**Training Data Source**	**Base Model**	**Highlights**	**Citation**
Mono-lingual	PhoBERT	Vietnamese	20GB pre-training dataset, including: (i) Vietnamese Wikipedia (~1GB); (ii) Vietnamese news dataset (~19GB)	RoBERTa	Uses syllable-level tokenizer, trained on a large Vietnamese dataset with fastBPE.	[40]
Mono-lingual	viBERT	Vietnamese	0GB Vietnamese news datasets (vnexpress.net, dantri.com.vn, baomoi.com, zingnews.vn, vitalk.vn, …etc.)	BERT	Improved performance on Vietnamese text processing tasks due to training on Vietnamese-specific data and pre-training techniques.	[41]
Mono-lingual	BARTpho	Vietnamese	The training data is an undivided variant of the PhoBERT pre-training corpus (about 4 billion syllable tokens)	BART	Combines Transformer structure with BERT, using a large Vietnamese dataset to enhance text generation and summarization quality.	[42]
Mono-lingual	ViT5	Vietnamese	- CC100 Dataset: Total size 138GB of raw text. - Data split: - 69GB short sentences for 256-length model. - 71GB long sentences for 1024-length model	T5	ViT5 applies Transformer-based Encoder-Decoder architecture, with two versions: Base (310M parameters) and Large (866M parameters). The model uses 36K sub-words generated by SentencePiece and trained with span-corruption self-supervision (15% rate).	[43]
Multi-lingual	XLM-RoBERTa-Base	100 languages	CommonCrawl, Wikipedia	RoBERTa	Trained on 100 languages. Uses Masked Language Modeling (MLM) objective. Vocabulary size = 250K, using SentencePiece. Training data from CommonCrawl and Wikipedia, with improved support for low-resource languages.	[44]
Multi-lingual	BERT	English	Wikipedia (2.5 billion words), BooksCorpus (800 million words)	Transformer	Trained using two unsupervised tasks: Masked LM and Next Sentence Prediction, utilizing a bidirectional Transformer architecture.	[45]
Multi-lingual	mT5	Over 100 languages, including Vietnamese	mC4 dataset (Massive Multi-lingual Crawled Corpus) collected from billions of web pages	T5	Multilingual pretraining, supports numerous languages using the T5 architecture.	[46]

In the context of this research, various Transformer-based models serve as the base models for the boosting methods explored. These models, which include both mono-lingual and multi-lingual variants, are pre-trained on large, domain-specific datasets and exhibit remarkable performance in natural-language processing tasks. Table 1 summarizes these base models, their training data sources and key highlights, showing how they contribute to enhancing model performance through boosting techniques.

2.4 Few-shot Learning Using Contrastive Learning

Few-shot learning (FSL) presents a significant challenge, as it requires models to adapt and generalize effectively with only a limited amount of data. Contrastive learning, a self-supervised method, has proven to be highly effective in addressing this challenge by learning meaningful and discriminative feature representations. By emphasizing similarities and differences among data points, contrastive learning aligns well with the objectives of FSL, where the focus is on distinguishing between unseen classes using minimal training data.

Contrastive-learning methods for FSL are often based on principles, such as noise contrastive estimation (NCE) [47]-[48] or N-pair losses [49], which facilitate the learning of robust feature spaces. For instance, SimCLR [17] employs data augmentation and non-linear transformations to train encoders that pull embeddings of similar data points closer together while pushing apart embeddings of dissimilar ones. Additionally, supervised contrastive learning [21] extends this framework to leverage labeled data, which is particularly useful in FSL scenarios where labeled support sets are small, but crucial.

In the context of FSL, contrastive learning enhances the effectiveness of models by improving the quality of representations derived from the support set (training examples). Key methods include:

- Instance-based Representations: Non-parametric softmax classifiers, such as those introduced in [50], focus on maximizing the separation between instance-level feature embeddings, helping models better distinguish between novel classes in FSL tasks.
- Multi-view Learning: Techniques like Time-Contrastive Networks (TCNs) [51] make use of multi-view data, aligning positive pairs (e.g. related samples, such as video frames) while separating negative pairs. In FSL, this can help bridge gaps between the limited support and query sets.
- Maximizing Information Representation: Methods, such as Deep InfoMax [52] among others [53], aim to maximize mutual information either within input-output pairs or across views of the same data. These methods ensure robust and meaningful feature extraction, improving FSL task performance.

Contrastive learning naturally integrates with metric-based FSL approaches, such as Prototypical Networks [54] and Siamese Networks [55], which rely on embedding distances. Discriminative representations learned through contrastive losses can significantly enhance the performance of these methods. Moreover, episodic training, commonly used in FSL, complements contrastive learning by structuring tasks to mimic real-world applications.

By leveraging contrastive learning, FSL models are better equipped to generalize from minimal data, offering a robust pathway for improving performance on tasks with scarce training resources. This combination demonstrates substantial potential to advance the effectiveness of few-shot learning in various domains.

3. METHODOLOGY

3.1 Dataset

3.1.1 Vietnamese Student Feedback

The dataset used in this study is the UIT-VSFC corpus, which consists of student feedback collected from a Vietnamese university. The dataset comprises 16,175 feedback sentences annotated with three sentiment categories: negative (0), neutral (1) and positive (2). Additionally, the dataset includes classifications for four main topics: Lecturer (0), Curriculum (1), Facility (2) and Others (3). Feedback was gathered between 2013 and 2016 through an automated survey system at the end of each semester. The surveys employed a 5-point Likert scale to assess pre-defined criteria and open-ended questions to gather more detailed feedback.

A key strength of this dataset is its reliability, demonstrated by an inter-annotator agreement score of 91%, which reflects a high level of consistency in sentiment labeling [56]. To evaluate few-shot learning scenarios, sub-sets of the training data were constructed with limited labeled samples per class. This setup ensured that the models were trained and tested under minimal data conditions, providing a robust assessment of their generalization capabilities with few-shot learning. Table 2 presents some examples from the dataset.

Table 3 presents the distribution of sentiment and topic categories. The dataset is highly imbalanced, with positive and negative sentiments each accounting for nearly 50%, while neutral feedback represents only 4.32%. In terms of topic labels, the majority of the feedback pertains to the Lecturer category (71.76%), followed by Curriculum (18.79%), indicating that students tend to comment most frequently on teaching-related aspects.

Furthermore, a linguistic analysis of the dataset reveals that student feedback tends to be concise: over 83% of the sentences contain 15 words or fewer. As shown in Table 4, negative sentences are generally longer than positive or neutral ones, likely because they often include justifications or suggestions for improvement. Table 5 displays the length distribution by topic, where feedback related to Lecturer, Curriculum and Facility frequently involves more detailed expressions (i.e., more than five words), reflecting students’ emphasis on those aspects.

Table 2. Examples of the UIT-VSFC dataset.

No.	Sentence	Sentiment	Topic
1	Giảng dạy nhiệt tình, liên hệ thực tế khá nhiều, tương tác với sinh viên tương đối tốt. (Enthusiastic teaching, incorporating a lot of real-life examples and relatively good interaction with students.)	Positive (2)	Lecturer (0)
2	Tính thực tế cũng cao so với việc thi lý thuyết lấy điểm. (It is also more practical compared to taking theoretical exams for	Positive (2)	Curriculum (1)
3	Phòng máy cũ, nhưng nhìn chung thì không có ảnh hưởng gì vì thầy dạy rất nhiệt tình. (The computer lab is outdated, but overall, it doesn't affect much, because the teacher is very enthusiastic.)	Neutral (1)	Facility (2)
4	Học thì quá ít nhưng khi thi thì quá nhiều yêu cầu viết code trong đề thi thì sao mà sinh viên có thể làm được. (The amount of learning is too little, but the exam demands too much coding. How can students possibly handle it?)	Negative (0)	Others (3)

Table 3. Distribution of sentiment and topic labels in the UIT-VSFC corpus (%).

Topic	Positive (%)	Negative (%)	Neutral (%)	Total (%)
Lecturer	33.57	25.38	1.81	71.76
Curriculum	3.40	14.39	1.00	18.79
Facility	0.11	4.21	0.08	4.4
Others	1.61	2.01	1.43	5.04
Total	**49.69**	**45.99**	**4.32**	**100**

Table 4. Distribution of sentences by sentiment and sentence length (%).

Length (words)	Positive (%)	Negative (%)	Neutral (%)	Total (%)
1–5	17.26	9.75	2.31	29.32
6–10	21.00	15.34	1.17	37.55
11–15	7.19	8.59	0.51	16.29
16–20	2.37	5.17	0.15	7.69
21–25	1.06	2.85	0.07	3.98
26–30	0.37	1.72	0.07	2.16
>30	0.40	2.57	0.04	3.01
Total	**49.65**	**45.99**	**4.32**	**100**

Table 5. Sentence-length distribution by topic (%).

Length (words)	**Lecturer (%)**	**Curriculum (%)**	**Facility (%)**	**Others (%)**	**Total (%)**
1–5	20.80	3.61	2.63	2.28	29.32
6–10	27.84	6.69	1.94	1.08	37.55
11–15	11.93	2.61	0.84	0.91	16.29
16–20	5.44	1.35	0.46	0.44	7.69
21–25	2.96	0.62	0.25	0.15	3.98
26–30	1.56	0.32	0.19	0.09	2.16
>30	1.13	0.59	0.10	1.19	3.01

3.1.2 Customer Product Reviews Dataset

To further evaluate model generalization, particularly for few-shot learning tasks across different domains, we utilized the "Vietnamese Sentiment Analyst" dataset, herein referred to as Customer Product Reviews. This corpus contains 31,460 Vietnamese customer reviews focused on various products. Each review is labeled with one of three sentiment polarities: positive, negative, or neutral. Table 6 presents some examples from the dataset.

Table 7 details the distribution of sentiment labels and sentence lengths within this dataset. Overall, positive sentiment is predominant (63.87%, N=20,093). In terms of sentence length, reviews are generally concise, with the highest concentration of positive reviews in the 1-5 word (20.84% of total dataset) and 6-10 word (21.14%) brackets.

Table 6. Examples of the customer product reviews dataset.

No.	**Sentence**	**Sentiment**
1	Chất lượng sản phẩm đúng như hình. Đóng gói sản phẩm tạm được. (The product quality is just like in the pictures. The packaging is acceptable.)	Positive (2)
2	Cơ mà tôi mua hôm nay, ngày mai shop làm flash sale là sao. (But I bought it today and now the shop is doing a flash sale tomorrow — what's that about?)	Neutral (1)
3	Có giống hình nhưng vải rất mỏng không đúng như trong hình. Giá tiền tương đương với sản phẩm. (It looks like the picture, but the fabric is very thin and not as shown. The price is equivalent to the product.)	Negative (0)

Table 7. Distribution of sentiment labels by review length.

Length (words)	**Positive (%)**	**Negative (%)**	**Neutral (%)**
1–5	20.84	6.61	5.18
6–10	21.14	7.47	5.36
11–15	9.46	3.53	2.4
16–20	4.96	1.71	1.06
21–25	2.83	0.79	0.52
26–30	1.96	0.48	0.2
>30	2.68	0.62	0.21
Total	63.87	21.2	14.93

3.2 Model Evaluation Metrics

These metrics are typically calculated using weighted averages to better reflect performance, especially in imbalanced datasets.

Precision measures the ratio of correctly predicted positive instances to all predicted positive instances. It is crucial in problems where false positives have high costs. Precision ranges from 0 to 1 and can be calculated as a weighted average, considering class sample sizes.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

Recall measures the model’s ability to detect actual positive instances. It is important in problems where missing positive cases can have severe consequences. Like Precision, Recall ranges from 0 to 1 and can be computed as a weighted average.

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

F1-score combines Precision and Recall to give a comprehensive performance measure, especially useful in imbalanced datasets. It ranges from 0 to 1, with higher values indicating a better balance between Precision and Recall. When calculated as a weighted average, it reflects the model's overall performance across all classes.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3.3 Software and Hardware

For the proposed research, Python was used as the programming language within the Google Colab runtime environment, which provides access to powerful hardware acceleration through GPUs. Specifically, the NVIDIA Tesla T4 GPU was utilized, equipped with 2560 CUDA cores designed to support deep-learning tasks. These cores, along with specialized Tensor Cores, allow for efficient execution of matrix-heavy operations commonly used in neural-network models. The environment ran on a CPU with an Intel (R) Core (TM) i3-4005U Processor at 1.70 GHz, paired with 4 GB of RAM.

To clarify the computational cost, Table 8 presents the number of trainable parameters and the approximate model size (in MB) for each transformer-based model evaluated in this study. Models with a higher number of parameters and larger memory footprints—such as mBART Large EN-RO (610M parameters, ~2.3GB) or mT5 Base (390M parameters, ~1.5GB)—require significantly more GPU memory, training time and processing power for both fine-tuning and inference. In contrast, smaller models, like ViBERT and PhoBERT, are comparatively lightweight and faster to train, making them more suitable for environments with limited computational resources. Table 8 presents the number of parameters and the sizes of the transformer models used in this study.

Table 8. Trainable parameters and approximate model sizes of pretrained transformer models.

Model	**Trainable Parameters**	**Model Size (MB)**
PhoBERT	134,998,272	514.98
ViBERT	115,354,368	440.04
XLM-RoBERTa Base	278,043,648	1,060.65
BERT Base Uncased	109,482,240	417.64
mT5 Base	390,315,264	1,488.93
BERT Base Multilingual Cased	177,853,440	678.46
mBART Large EN-RO	610,851,840	2,330.21
BARTpho-syllable	395,814,912	1,509.91
ViT5 Base	225,950,976	861.93

3.4 Experimental Framework

Few-shot Learning was implemented with varying levels of data availability (N = 1, 5 and 20) to evaluate the performance of several transformer-based models on limited labeled data. The models included PhoBERT, ViBERT, XLM-RoBERTa, mT5, multi-lingual BERT, base BERT, MBart, BARTpho and

ViT5. Each model was fine-tuned using a contrastive learning approach and their performances were evaluated using the F1-score. In addition to transformer-based models, the study also conducted experiments with several classical machine-learning architectures, including RNN, GRU and LSTM, to serve as comparative baselines. This inclusion provides a broader perspective on the effectiveness of modern pre-trained models under low-resource conditions.

For the ensemble-learning stage, our primary selection criterion was individual model performance. Consequently, the top three models demonstrating the highest average F1-scores were chosen as base learners. To validate this selection, we conducted pairwise statistical significance tests (paired t-tests), which confirmed that these models belonged to a top-performing tier, showing statistically significant improvements over most other models. This approach ensures that the components of our ensemble are strong and reliable individual predictors.

To further improve prediction accuracy, a supervised ensemble strategy based on boosting was applied. Instead of using simple combination methods, such as majority voting or averaging, the outputs from the top-three transformer models served as input features for three ensemble learners: AdaBoost, Gradient Boosting and XGBoost. These ensemble models were trained to learn from the prediction patterns of the base models, functioning as meta-learners that integrate their outputs into a final decision. This method is analogous to a stacking framework, where boosting algorithms iteratively focus on samples that are harder to classify, thereby refining predictions and enhancing overall generalization performance. Detailed descriptions of the proposed method and framework are presented in Figure 1.

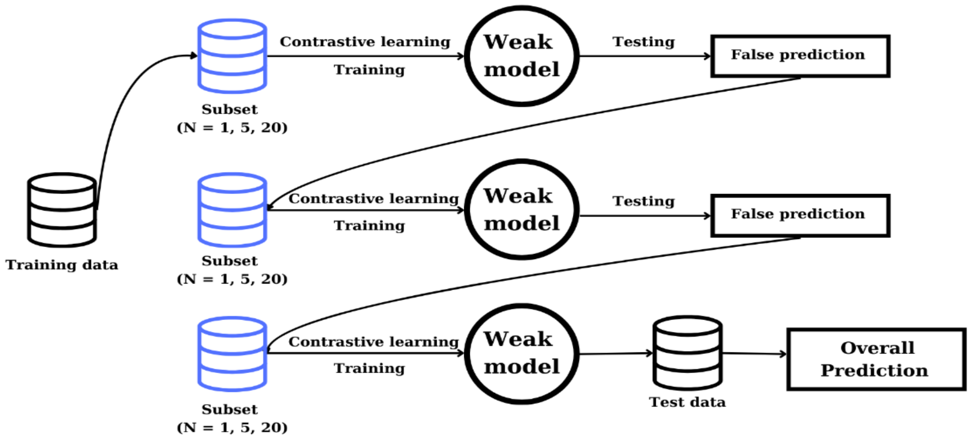

Figure 1. Flow diagram of proposed methodology. The framework trains weak models on data subsets (N = 1, 5, 20) using contrastive learning. False predictions are identified during testing and outputs are combined to produce the final overall prediction on test data [56]–[57].

3.5 Hyper-parameter Tuning

Bayesian optimization is a powerful and efficient method for hyper-parameter tuning, especially in complex machine-learning models where traditional techniques, such as Grid Search and Random Search, fall short due to their inefficiency or lack of strategic sampling. By modeling the objective function using a probabilistic surrogate model, Bayesian optimization intelligently selects the next sampling point based on past evaluations, effectively balancing exploration and exploitation. This approach is particularly suitable for combinatorial optimization problems where gradient-based methods are not applicable. Bayesian optimization is the top choice for optimizing objective functions [57-59]. In this study, Bayesian optimization is employed to tune hyper-parameters for boosting algorithms, including AdaBoost, Gradient Boosting and XGBoost. Examples of optimized parameters include the learning rate, number of estimators, maximum tree depth, …etc.

Tables 9, 10 and 11 present the hyper-parameters of the boosting models—AdaBoost, Gradient Boosting and XGBoost—that were optimized using Bayesian optimization. These tables detail the specific parameters selected for tuning, such as learning rate, number of estimators and maximum depth, among others, which play a crucial role in controlling model complexity, convergence behaviour and overall predictive performance.

Table 9. Optimized hyper-parameters using Bayesian optimization for AdaBoost across datasets and N-shot settings.

Dataset	N-shot	Learning Rate	N estimators
UIT-VSFC (Sentiment)	N=1	0.010	820
	N=5	0.650	29
	N=20	0.279	884
UIT-VSFC (Topic)	N=1	0.159	920
	N=5	0.677	1000
	N=20	0.558	180
Customer Product Reviews	N=1	0.820	884
	N=5	0.128	730
	N=20	0.159	920

Table 10. Optimized hyper-parameters using Bayesian optimization for XGBoost across datasets and N-shot settings.

Dataset	N-shot	Column Subsample	Learning Rate	Max. Depth	No. of Estimators	L1 Regularization	L2 Regularization	Subsample Ratio
UIT-VSFC (Sentiment)	N=1	0.300	0.010	11	506	0.703	0.955	1.000
	N=5	0.680	0.229	7	854	0.324	0.051	0.785
	N=20	0.969	0.108	11	474	0.381	0.211	0.500
UIT-VSFC (Topic)	N=1	1.000	0.168	12	1000	1.000	0.000	0.873
	N=5	0.300	0.062	5	1000	0.000	1.000	1.000
	N=20	0.611	0.228	4	490	0.188	0.454	0.578
Customer Product Reviews	N=1	1	0.027	3	100	1	0	1
	N=5	0.969	0.108	11	474	0.381	0.211	0.5
	N=20	1	0.025	9	551	1	0.549	0.519

Table 11. Optimized hyper-parameters using Bayesian optimization for Gradient Boosting across datasets and N-shot settings.

Dataset	N-shot	Learning Rate	Maximum Depth	Minimum Samples per Leaf	Minimum Samples to Split	Number of Estimators	Subsample Ratio
UIT-VSFC (Sentiment)	N=1	0.082	10	4	9	633	0.797
	N=5	0.072	11	2	8	812	0.504
	N=20	0.279	7	10	2	173	0.597
UIT-VSFC (Topic)	N=1	0.029	3	1	2	337	0.913
	N=5	0.013	8	2	4	600	0.900
	N=20	0.170	12	10	2	100	0.774
Customer Product Reviews	N=1	0.258	9	9	5	443	0.606
	N=5	0.298	10	9	9	195	0.520
	N=20	0.146	11	2	7	608	0.531

The hyper-parameters optimized in this study critically influence the balance between model bias and variance, as well as training efficiency. Learning rate determines the step size during model updates, affecting convergence speed and overfitting risk. Number of estimators specifies how many weak learners (trees) are combined, impacting the model's capacity and complexity.

For XGBoost, additional parameters, such as column sub-sample ratio, control the fraction of features used per tree to prevent overfitting. Maximum tree depth limits the complexity of individual trees. L1 (reg_alpha) and L2 (reg_lambda) regularization terms penalize model complexity to enhance robustness, while sub-sample ratio governs the portion of training data sampled per tree, reducing variance.

In Gradient Boosting, besides learning rate and number of estimators, the minimum samples per leaf and minimum samples to split parameters regulate tree growth by specifying thresholds for leaf-node formation and internal-node splitting, further preventing overfitting.

3.6 Statistical Significance Testing and Confidence Intervals

A paired t-test is used to determine whether the difference in performance between models is statistically significant. Instead of using k-fold cross-validation, the models are run multiple times with different random initializations to generate sets of performance results. For each run, the performance difference between two models A and B is calculated as:

$$d_i = acc_i(A) - acc_i(B)$$

From these differences, the sample mean is computed as:

$$m = \frac{1}{N}\sum_{n=1}^{N} \mathrm{diff}_n$$

and the sample standard deviation is:

$$sd = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(\mathrm{diff}_n - m)^2}$$

The t-statistics are then calculated as:

$$t = \frac{m\sqrt{N}}{sd}$$

Finally, the t-value is compared against the critical value from the t-distribution with N−1 degrees of freedom to test the null hypothesis. If the p-value is less than 0.05 ($p<0.05$), it can be concluded that the difference between the two models is statistically significant. Using the paired t-test thus helps strengthen the reliability of selecting more effective models.

Besides the paired t-test, the 95% Confidence Interval (CI) is used to provide a range within which the true performance metric of each model is likely to fall with 95% certainty. Each model is run 5 times with different random seeds to capture the variability caused by random initialization. Reporting the mean performance along with the 95% CI reflects the stability and reliability of the models.

This approach allows for a more comprehensive evaluation by quantifying the uncertainty around the average performance, ensuring that model comparison and selection consider not only the mean accuracy, but also the consistency across multiple runs.

4. RESULTS

4.1 Few-shot Learning Experiments on Transformer Models

The experimental results of transformer models are presented on the dataset for two tasks: sentiment classification and topic classification. Additionally, experiments were conducted on sentiment analysis using the customer product reviews dataset. Each model is evaluated on the same training dataset with setups of N = 1, N = 5 and N = 20. The training environment and hyper-parameters are identical across all models. The reports highlight the precision, recall and F1-score achieved by each model, specifying which transformers perform well in 1-shot learning (N = 1), few-shot learning (N = 5) and scenarios with a significant amount of data.

Table 12 shows the experimental results on the sentiment-analysis task, with XLM-RoBERTa outperforming other models and achieving the highest F1-scores. This model demonstrates the best performance in precision, recall and F1-score, making it the most effective model for sentiment analysis. Other models, such as BARTpho and BERT multi-lingual, also show strong results.

Table 13 shows the experimental results on the topic-classification task. The highest F1-score for N = 20 is 0.817, achieved by XLM-RoBERTa. PhoBERT and BARTpho also show strong performance, but XLM-RoBERTa leads in this setup. Table 14 presents the experimental results on the customer product reviews dataset. The highest F1-score for N = 20 is 0.744, achieved by mT5. ViBERT and ViT5 also show strong performance.

Notably, the confidence intervals (CIs) among transformer-based models show minimal variation, with differences generally remaining below 0.02. This indicates consistent and stable performance across different runs. In contrast, traditional models, such as LSTM, RNN and GRU, exhibit greater fluctuations in their CI values, reflecting less stability and higher variability in performance.

Table 12. The experimental results of transformer models for sentiment analysis.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.449± 0.0375	0.251± 0.0451	0.322± 0.0396	0.520± 0.0296	0.387± 0.0416	0.444± 0.0312	0.645± 0.0261	0.502± 0.0421	0.565± 0.0364
GRU	0.369± 0.0223	0.287± 0.0322	0.323± 0.0268	0.552± 0.0575	0.477± 0.0428	0.512± 0.0443	0.654± 0.0370	0.591± 0.0503	0.621± 0.0449
LSTM	0.381± 0.0122	0.381± 0.0320	0.381± 0.0289	0.626± 0.0366	0.504± 0.0198	0.558± 0.0217	0.657± 0.0366	0.586± 0.0310	0.619± 0.0343
PhoBERT	0.610± 0.0081	0.591± 0.0098	0.596± 0.0079	0.759± 0.0048	0.708± 0.0053	0.733± 0.0036	0.846± 0.0055	0.812± 0.0045	0.829± 0.0049
ViBERT	0.549± 0.0121	0.278± 0.0106	0.369± 0.0088	0.580± 0.0083	0.499± 0.0036	0.536± 0.0076	0.723± 0.0083	0.608± 0.0076	0.661± 0.0077
XLM-RoBERTa	0.603± 0.0075	0.470± 0.0089	0.528± 0.0077	0.720± 0.0040	0.625± 0.0066	0.669± 0.0058	0.843± 0.0081	0.834± 0.0075	***0.838± 0.0075***
BERT base	0.597± 0.0098	0.527± 0.0032	0.560± 0.0038	0.692± 0.0020	0.460± 0.0088	0.553± 0.0033	0.672± 0.0038	0.630± 0.0081	0.650± 0.0076
mT5	0.606± 0.0072	0.471± 0.0025	0.530± 0.0057	0.769± 0.0047	0.653± 0.0027	0.653± 0.0046	0.779± 0.0096	0.692± 0.0052	0.721± 0.0080
BERT multilingual	0.656± 0.0125	0.655± 0.0098	***0.655± 0.0101***	0.748± 0.0186	0.672± 0.0143	0.672± 0.0153	0.801± 0.0142	0.743± 0.0096	0.765± 0.0138
MBart	0.582± 0.0069	0.525± 0.0052	0.552± 0.0057	0.685± 0.0091	0.638± 0.0093	0.661± 0.0090	0.811± 0.0076	0.793± 0.0096	0.801± 0.0082
BARTpho	0.608± 0.0093	0.533± 0.0082	0.568± 0.0081	0.764± 0.0091	0.712± 0.0087	***0.737± 0.0090***	0.843± 0.0064	0.780± 0.0097	0.806± 0.0084
ViT5	0.594± 0.0188	0.590± 0.0157	0.592± 0.0165	0.745± 0.0109	0.611± 0.0146	0.671± 0.0138	0.825± 0.0070	0.742± 0.0051	0.771± 0.0069

4.2 Pairwise Statistical Significance Testing Using Paired T-test

After training and evaluating all models on two primary tasks, sentiment analysis and topic classification, additional experiments were also conducted on sentiment analysis using the customer product reviews dataset. The three models with the highest F1-scores were selected to undergo paired t-test evaluation against each of the remaining models. The objective was to assess whether the performance differences between models are statistically significant.

Each model was run five times with different random seeds to capture variation introduced by random initialization. The performance differences (in terms of F1-score) between each model pair were calculated and a paired t-test was conducted using a significance threshold of $p<0.05$. The results show that the top three models consistently outperformed most other models with statistically significant differences, confirming their superiority in a reliable manner. Notably, the model with the lowest average performance still achieved statistically significant results ($p < 0.05$) in two comparisons, indicating that it also qualifies for inclusion in the ensemble model.

Table 13. The experimental results of transformer models for topic analysis.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.540± 0.0411	0.197± 0.0325	0.289± 0.0336	0.599± 0.0233	0.297± 0.0341	0.397± 0.0302	0.624± 0.0265	0.388± 0.0372	0.478± 0.0298
GRU	0.481± 0.0231	0.237± 0.0421	0.318± 0.0403	0.633± 0.0158	0.356± 0.0229	0.456± 0.0196	0.644± 0.0331	0.669± 0.0253	0.656± 0.0268
LSTM	0.491± 0.0321	0.229± 0.0210	0.312± 0.0298	0.649± 0.0254	0.323± 0.0187	0.431± 0.0203	0.524± 0.0135	0.715± 0.0201	0.605± 0.0184
PhoBERT	0.708± 0.0101	0.647± 0.0128	0.676± 0.0120	0.762± 0.0063	0.667± 0.0098	0.711± 0.0088	0.821± 0.0063	0.767± 0.0041	0.791± 0.0055
ViBERT	0.679± 0.0156	0.214± 0.0203	0.325± 0.0139	0.708± 0.0109	0.534± 0.0056	0.609± 0.0063	0.774± 0.0182	0.682± 0.0099	0.725± 0.0103
XLM-RoBERTa	0.639± 0.0095	0.646± 0.0127	0.642± 0.0110	0.741± 0.0063	0.630± 0.0036	0.681± 0.0054	0.841± 0.0096	0.795± 0.0082	***0.817*±** ***0.0079***
BERT base	0.588± 0.0153	0.278± 0.0102	0.378± 0.0115	0.691± 0.0118	0.497± 0.0064	0.578± 0.0082	0.754± 0.0053	0.644± 0.0089	0.695± 0.0076
mT5	0.672± 0.0089	0.448± 0.0056	0.538± 0.0076	0.734± 0.0038	0.451± 0.0025	0.559± 0.0030	0.836± 0.0056	0.719± 0.0089	0.773± 0.0088
BERT multilingual	0.696± 0.0145	0.696± 0.0096	0.696± 0.0135	0.790± 0.0202	0.594± 0.0158	0.678± 0.0166	0.820± 0.0083	0.719± 0.0103	0.766± 0.0096
MBart	0.642± 0.0080	0.547± 0.0088	0.591± 0.0082	0.823± 0.0093	0.738± 0.0066	***0.778*±** ***0.0083***	0.846± 0.0103	0.768± 0.0152	0.805± 0.0109
BARTpho	0.692± 0.0132	0.419± 0.0122	0.522± 0.0126	0.783± 0.0102	0.661± 0.0123	0.744± 0.099	0.850± 0.0101	0.763± 0.0095	0.804± 0.0097
ViT5	0.736± 0.0052	0.684± 0.0085	***0.709*±** ***0.0063***	0.786± 0.0102	0.660± 0.0092	0.741± 0.0091	0.846± 0.0064	0.780± 0.0092	0.812± 0.0066

Table 14. The experimental results of transformer models for customer product reviews dataset.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.305± 0.0482	0.321± 0.0554	0.313± 0.0501	0.462± 0.0363	0.453± 0.0382	0.457± 0.0351	0.515± 0.0334	0.496± 0.0312	0.503± 0.0305
GRU	0.324± 0.0505	0.343± 0.0578	0.332± 0.0524	0.481± 0.0381	0.472± 0.0403	0.475± 0.0372	0.533± 0.0352	0.514± 0.0331	0.521± 0.0323
LSTM	0.342± 0.0521	0.361± 0.0595	0.350± 0.0543	0.503± 0.0402	0.491± 0.0425	0.494± 0.0391	0.552± 0.0373	0.535± 0.0354	0.543± 0.0342
PhoBERT	0.456± 0.0121	0.484± 0.0142	0.470± 0.0135	0.623± 0.0083	0.616± 0.0102	0.619± 0.0091	0.701± 0.0072	0.679± 0.0064	0.690± 0.0068
ViBERT	0.469± 0.0163	0.484± 0.0211	0.476± 0.0184	0.685± 0.0119	0.680± 0.0098	0.682± 0.0105	0.729± 0.0121	0.729± 0.0103	0.729± 0.0112
XLM-RoBERTa	0.397± 0.0112	0.535± 0.0135	0.456± 0.0121	0.694± 0.0091	0.643± 0.0103	0.668± 0.0095	0.725± 0.0087	0.677± 0.0079	0.700± 0.0081
BERT base	0.471± 0.0185	0.516± 0.0199	***0.492± 0.0191***	0.620± 0.0131	0.622± 0.0124	0.621± 0.0128	0.679± 0.0093	0.670± 0.0108	0.674± 0.0099
mT5	0.457± 0.0138	0.508± 0.0145	0.481± 0.0141	0.699± 0.0095	0.676± 0.0115	***0.687± 0.0101***	0.748± 0.0086	0.741± 0.0094	***0.744± 0.0090***
BERT multilingual	0.451± 0.0152	0.427± 0.0148	0.439± 0.0149	0.685± 0.0122	0.632± 0.0138	0.657± 0.0129	0.728± 0.0098	0.697± 0.0113	0.712± 0.0104
MBart	0.437± 0.0115	0.456± 0.0128	0.446± 0.0119	0.658± 0.0081	0.628± 0.0094	0.643± 0.0094	0.756± 0.0079	0.677± 0.0091	0.714± 0.0084
BARTpho	0.444± 0.0141	0.441± 0.0153	0.442± 0.0148	0.669± 0.0112	0.632± 0.0109	0.650± 0.0110	0.760± 0.0081	0.709± 0.0092	0.734± 0.0087
ViT5	0.483± 0.0102	0.485± 0.0115	0.484± 0.0108	0.653± 0.0092	0.669± 0.0105	0.661± 0.0097	0.726± 0.0074	0.734± 0.0082	0.730± 0.0078

This evaluation approach, based on paired t-tests, ensures that model selection is not solely based on average performance, but also considers stability and statistical significance across multiple runs, thereby enhancing the robustness and reliability of the final model-selection process. The results of the paired t-tests are reported in Tables from 15 to 23.

Note on statistical-significance levels: (*: p < 0.05), (**: p < 0.01) and (***: p < 0.001).

Table 15. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***	***	0.0245	***	***	***	***	0.0377
BERT multilingual	***	***	***	***	***		***	***	***
ViT5	0.0377	***	***	***	***	***	***	***	

Table 16. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***	***	***	***	0.0108	***	***	***
BERT multilingual	0.0108	***	***	***	***		***	***	***
BARTpho	***	***	***	***	**	***	***0.0518***		***

Table 17. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
XLM-RoBERTa	***	***		***	***	***	***	***	***
PhoBERT		***	***	***	***	***	***	***	***
BARTpho	***	***	***	***	***	***	***		***

Table 18. Pairwise statistical significance testing using paired t-test on topic-classification task (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***		***	***	***	***	***	***
BERT multilingual	**	***	***	***	***		***	***	0.0249
ViT5	***	***	***	***	***	0.0249	***	***	

Table 19. Pairwise statistical significance testing using paired t-test on topic-classification task (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
MBart	***	***	***	***	***	***		***0.6952***	***0.6951***
BARTpho	**	***	***	***	***	***	***0.6952***		**
ViT5	0.0730	***	***	***	***	***	***	**	

Table 20. Pairwise statistical significance testing using paired t-test on topic-classification Task (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
XLM-RoBERTa	***	***		***	***	***	***	***	**
MBart	***	***	***	***	***	***		***0.3903***	0.0479
ViT5	***	***	**	***	***	***	0.0479	***	

Table 21. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
ViBERT	0.0322		***	0.0359	***	***	***	**	**
BERT base	***	0.0359	***		**	***	***	*******	**
mT5	***	***	**	**		***	***	**	

Table 22. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
mT5	***	**	0.0122	***		***	***	***	**
ViBERT	***		***	***	**	***	***	**	***
XLM-RoBERTa	***	***		***	0.0122	***	**	***	***

Table 23. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
mT5	***	***	***	***		***	***	***	***0.5856***
BARTpho	***	***	***	***	***	**	***		0.0152
ViT5	***	***	***	***	***0.5856***	***	***	0.0152	

4.3 Experiments on Boosting Models with Transformers

Based on the few-shot learning experiments with transformers, the study conducted boosting experiments using the best-performing models. Specifically, the three models with the highest F1-scores were selected as base models for three boosting methods. Table 24 and Table 25 present the experimental results for two tasks: sentiment analysis and topic classification. Table 26 presents the experimental results on the customer product reviews dataset. The results indicate that Gradient Boosting achieved the best performance across all tasks and base models. With N=20, Gradient Boosting reached an F1-score of 0.836 on the sentiment-analysis task and 0.824 on the topic-classification task. However, the performance of the other two methods was also very promising.

Table 24. Experimental results of boosting on the sentiment-analysis task.

N	**Base model**	**AdaBoost**			**Gradient Boosting**			**XGBoost**		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	PhoBERT + BERT multilingual + ViT5	0.639	0.670	0.648	0.665	0.675	***0.661***	0.638	0.671	0.653
5	PhoBERT + BERT multilingual+ BARTpho	0.754	0.785	0.765	0.792	0.796	***0.776***	0.772	0.796	0.774
20	XLM-RoBERTa +BERT multilingual+ BARTpho	0.798	0.841	0.819	0.837	0.853	***0.836***	0.833	0.849	***0.836***

Table 25. Experimental results of boosting on the topic-classification task.

N	**Base model**	**AdaBoost**			**Gradient Boosting**			**XGBoost**		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	PhoBERT + BERT multilingual + ViT5	0.732	0.754	0.709	0.723	0.758	***0.725***	0.717	0.748	0.723
5	MBart + BARTpho + ViT5	0.799	0.803	0.735	0.811	0.812	***0.804***	0.789	0.804	0.790
20	XLM-RoBERTa+ Bart + ViT5	0.826	0.834	0.817	0.832	0.819	***0.824***	0.795	0.829	0.811

Table 26. Experimental results of boosting on the customer product reviews dataset.

N	Base model	AdaBoost			Gradient Boosting			XGBoost		
		P	R	F1	P	R	F1	P	R	F1
1	ViBERT + BERT base + mT5	0.532	0.556	0.544	0.536	0.573	***0.554***	0.530	0.555	0.542
5	mT5 + ViBERT+ XLM-RoBERTa	0.665	0.685	0.675	0.709	0.706	***0.707***	0.694	0.703	0.698
20	mT5 + BARTpho + ViT5	0.740	0.750	0.745	0.749	0.761	***0.755***	0.751	0.753	0.752

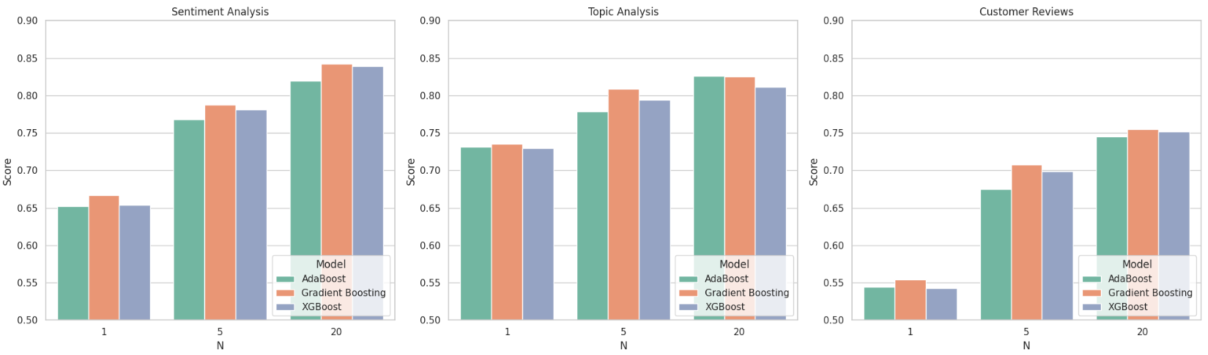

Figure 2. Comparison of F1-scores of boosting algorithms (AdaBoost, Gradient boosting, XGBoost) on two tasks: sentiment analysis and topic analysis, using different combined models.

5. Conclusions

The findings of this study have far-reaching implications that contribute to yet another theoretical and practical advancement in sentiment analysis, particularly in low-resource educational environments. To mitigate challenges, such as limited data and computational inefficiency, the proposed study introduces a novel framework that combines Few-Shot Learning (FSL) and Transformer-based ensemble models with boosting approaches.

By drawing on the strengths of both Transformer models using self-attention to learn patterns from rich data and adapting the FSL setting, this paper then introduces a hybrid methodology that addresses the shortcomings of traditional supervised approaches in low-data scenarios. Moreover, it presents the role of boosting techniques, such as Gradient boosting and XGBoost, and their capabilities in classifying the sentiments, which may set a pathway for forthcoming research on ensemble learning for NLP tasks.

On the practical side, the framework presented in this research will serve as a basis for providing actionable knowledge to educational institutes to better analyze students' feedback, hence improving their learning experience and the quality of teaching. The scalability of the method makes it relevant for a wide range of fields that experience a scarcity of labeled data. Furthermore, its efficient use of resources demonstrates its practicality for translating to practice, even in settings where computational power is limited. Although the model demonstrates effectiveness in sentiment-analysis tasks with limited training resources, particularly in educational feedback systems, this study acknowledges the ethical aspects associated with its real-world deployment. Fairness is a key concern when sentiment models are trained on imbalanced datasets in terms of class distribution, dialectal expressions and stylistic variations, which often predominantly reflect students' perspectives. This may result in systematic bias against certain groups.

Bias during evaluation and sentiment classification may lead the model to misinterpret students' feedback, especially when cultural context or specific expression styles are not accurately captured in the training data. For instance, negative feedback expressed politely or formally may be misclassified as neutral or even positive. This misunderstanding can delay necessary interventions by model users when addressing customer requests or student concerns. Another issue to consider is the impact of misclassification, which can lead to incorrect conclusions in both educational and customer-service evaluations. If negative feedback is misinterpreted as positive, educational administrators or customer-service staff may overlook significant issues raised by students or customers, potentially affecting the overall learning or service experience. To mitigate these risks, future research and deployments should apply fairness-aware training methods, such as data rebalancing and debiasing techniques, utilize more diverse datasets to increase representativeness and integrate human oversight during the result-validation process.

Despite the promising results, this study has several limitations that provide clear avenues for future research. First, our framework's effectiveness is contingent on the availability of high-quality pre-trained Transformer models. Consequently, its application may be challenging for low-resource languages or specialized domains that lack representative pre-training corpora. Second, the use of ensemble and boosting techniques, while improving performance, introduces additional computational complexity, which might be a barrier for organizations with limited resources. A third limitation lies in our ensemble selection logic. In this study, base models were chosen primarily based on their individual performance. While this ensures strong components, it does not explicitly guarantee model diversity, a critical factor for robust ensembling. Finally, as the evaluation was conducted on a single dataset (UIT-VSFC), the generalizability of our findings needs further validation on other datasets and across different domains.

Building on these limitations, future work can proceed in several promising directions. To address generalizability, the framework should be evaluated across diverse domains, such as healthcare or finance, and on datasets in other languages. To enhance the ensemble methodology, future research should explore more sophisticated, diversity-aware selection strategies that co-optimize for both model performance and diversity; for instance, by analyzing prediction correlations. Furthermore, performance in extremely low-data environments could be improved by optimizing Transformer architectures for lightweight deployments and leveraging advanced data-augmentation strategies. Finally, integrating human-in-the-loop feedback systems could improve model adaptability in ambiguous cases, making the framework more practical for real-world deployment. This research underscores the transformative potential of advanced NLP techniques in enhancing sentiment analysis, offering a valuable framework for addressing challenges in resource-constrained scenarios.

REFERENCES

[1] M. Bansal, S. Verma, K. Vig and K. Kakran, "Opinion Mining from Student Feedback Data Using Supervised Learning Algorithms," Lecture Notes in Networks and Systems, vol. 514, pp. 1–15, 2022.

[2] A. Ligthart, C. Catal and B. Tekinerdogan, "Systematic Reviews in Sentiment Analysis: A Tertiary Study," Artificial Intelligence Review, vol. 54, no. 7, pp. 4997–5053, 2021.

[3] A. I. M. Elfeky et al., "Advance Organizers in Flipped Classroom *via* e-Learning Management System and the Promotion of Integrated Science Process Skills," Thinking Skills and Creativity, vol. 35, 2020.

[4] H. Zhao et al., "A Machine Learning-based Sentiment Analysis of Online Product Reviews with a Novel Term Weighting and Feature Selection Approach," Inf. Process. Manag., vol. 58, no. 5, pp. 1–15, 2021.

[5] Y. Zhang, J. Wang and X. Zhang, "Conciseness is Better: Recurrent Attention LSTM Model for Document-level Sentiment Analysis," Neurocomputing, vol. 462, pp. 1–12, 2021.

[6] Z. Liu et al., "Temporal Emotion-aspect Modeling for Discovering What Students are Concerned about in Online Course Forums," Interactive Learning Environments, vol. 27, no. 5–6, pp. 1–15, 2019.

[7] J. J. Zhu et al., "Online Critical Review Classification in Response Strategy and Service Provider Rating: Algorithms from Heuristic Processing, Sentiment Analysis to Deep Learning," Journal of Business Research, vol. 129, pp. 1–12, DOI: 10.1016/j.jbusres.2020.11.007, 2021.

[8] F. A. Acheampong et al., "Transformer Models for Text-based Emotion Detection: A Review of BERT-based Approaches," Artificial Intelligence Review, vol. 54, no. 8, pp. 1–41, 2021.

[9] C. Dervenis, P. Fitsilis and O. Iatrellis, "A Review of Research on Teacher Competencies in Higher Education," Quality Assurance in Education, vol. 30, no. 2, pp. 1–15, 2022.

[10] M. Y. Salmony et al., "Leveraging Attention Layer in Improving Deep Learning Models' Performance for Sentiment Analysis," Int. J. of Information Technology (Singapore), vol. 15, no. 1, pp. 1–10, 2023.

[11] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, DOI: 10.1145/505282.505283, 2002.

[12] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques," Proc. of the 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79-86, DOI: 10.3115/1118693.1118704, 2002.

[13] N. Dong and E. P. Xing, "Few-shot Semantic Segmentation with Prototype Learning," Proc. of Brit. Mach. Vis. Conf. (BMVC 2018), [Online], Available: http://bmvc2018.org/contents/papers/0255.pdf.

[14] W. Li et al., "Revisiting Local Descriptor Based Image-to-class Measure for Few-shot Learning," Proc.

IEEE Conf. Comput. Vis. Pattern Recognit., pp. 7260-7268, DOI: 10.1109/CVPR.2019.00743, 2019.

[15] R. Hadsell et al., "Dimensionality Reduction by Learning an Invariant Mapping," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 1735-1742, DOI: 10.1109/CVPR.2006.100, 2006.

[16] K. He et al., "Momentum Contrast for Unsupervised Visual Representation Learning," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 9726-9735, DOI: 10.1109/CVPR42600.2020.00975, 2020.

[17] T. Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations," Proc. of the 37th Int. Conf. on Machine Learning (ICML 2020), pp. 1597-1607, Vienna, Austria, 2020.

[18] C. Li et al., "SentiPrompt: Sentiment Knowledge Enhanced Prompt-tuning for Aspect-based Sentiment Analysis," arXiv preprint, arXiv: 2109.08306, 2021.

[19] B. Liang et al., "Enhancing Aspect-based Sentiment Analysis with Supervised Contrastive Learning," Proc. Int. Conf. Inf. Knowl. Manage., pp. 3242-3247, DOI: 10.1145/3459637.3482096, 2021.

[20] J. J. Peper and L. Wang, "Generative Aspect-based Sentiment Analysis with Contrastive Learning and Expressive Structure," Proc. of Findings Assoc. Comput. Linguistics: EMNLP 2022, pp. 6086-6099, DOI: 10.18653/v1/2022.findings-emnlp.451, 2022.

[21] P. Khosla et al., "Supervised Contrastive Learning," Adv. Neural Inf. Process. Syst., vol. 33, pp. 18661-18673, 2020.

[22] Y. Wang, J. Wang, Z. Cao and A. Barati Farimani, "Molecular Contrastive Learning of Representations *via* Graph Neural Networks," Nature Machine Intelligent, vol. 4, no. 3, pp. 279-287, 2022.

[23] Z. Lin et al., "Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning," Proc. ACM Web Conf., pp. 2320-2329, DOI: 10.1145/3485447.3512104, 2022.

[24] J. Elith, J. R. Leathwick and T. Hastie, "A Working Guide to Boosted Regression Trees," Journal of Animal Ecology, vol. 77, no. 4, pp. 802-813, DOI: 10.1111/j.1365-2656.2008.01390.x, 2008.

[25] R. E. Schapire, "A Short Introduction to Boosting," Journal of the Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, DOI: 10.1.1.112.5912, 2009.

[26] M. Kuhn and K. Johnson, Applied Predictive Modeling, New York, NY: Springer, DOI: 10.1007/978-1-4614-6849-3, 2013.

[27] P. Wu and H. Zhao, "Some Analysis and Research of the AdaBoost Algorithm," Communications in Computer and Information Science, vol. 134, pp. 1-8, DOI: 10.1007/978-3-642-18129-0_1, 2011.

[28] F. Wang et al., "Feature Learning Viewpoint of AdaBoost and a New Algorithm," IEEE Access, vol. 7, pp. 149890-149899, DOI: 10.1109/ACCESS.2019.2947359, 2019.

[29] L. Breiman, "Arcing Classifiers," Annals of Statistics, vol. 26, no. 3, pp. 801-849, 1998.

[30] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, DOI: 10.1214/aos/1013203451, 2001.

[31] A. Natekin and A. Knoll, "Gradient Boosting Machines, a Tutorial," Frontiers in Neurorobotics, vol. 7, DOI: 10.3389/fnbot.2013.00021, Dec. 2013.

[32] B. Zhang et al., "Health Data Driven on Continuous Blood Pressure Prediction Based on Gradient Boosting Decision Tree Algorithm," IEEE Access, vol. 7, pp. 32423-32433, 2019.

[33] J. Jiang et al., "Boosting Tree-assisted Multitask Deep Learning for Small Scientific Datasets," Journal of Chemical Information and Modeling, vol. 60, no. 3, pp. 1235-1244, 2020.

[34] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 785-794, DOI: 10.1145/2939672.2939785, 2016.

[35] Y. Li and W. Chen, "A Comparative Performance Assessment of Ensemble Learning for Credit Scoring," Mathematics, vol. 8, no. 10, p. 1756, DOI: 10.3390/math8101756, 2020.

[36] W. Liang et al., "Predicting Hard Rock Pillar Stability Using GBDT, XGBoost and LightGBM Algorithms," Mathematics, vol. 8, no. 5, p. 765, DOI: 10.3390/MATH8050765, 2020.

[37] J. Nobre et al., "Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to Trade in the Financial Markets," Expert Systems with Applications, vol. 125, pp. 19-33, 2019.

[38] B. Zhang, Y. Zhang and X. Jiang, "Feature Selection for Global Tropospheric Ozone Prediction Based on the BO-XGBoost-RFE Algorithm," Scientific Reports, vol. 12, no. 1, 2022.

[39] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Inf. Proces. Syst., vol. 30, 2017.

[40] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained Language Models for Vietnamese," Proc. of Findings Assoc. Comput. Linguistics: EMNLP 2020, pp. 1037-1042, DOI: 10.18653/v1/2020.findings-emnlp.92, 2020.

[41] T. O. Tran and P. Le Hong, "Improving Sequence Tagging for Vietnamese Text Using Transformer-based Neural Models," Proc. of the 34th Pacific Asia Conf. Lang., Inf. Comput., pp. 13-20, 2020.

[42] N. L. Tran, D. M. Le and D. Q. Nguyen, "BARTpho: Pre-trained Sequence-to-sequence Models for Vietnamese," Proc. Interspeech, pp. 4895-4899, DOI: 10.21437/Interspeech.2022-10177, 2022.

[43] L. Phan et al., "ViT5: Pre-trained Text-to-text Transformer for Vietnamese Language Generation," Proc. NAACL-HLT Student Res. Workshop, pp. 128-135, DOI: 10.18653/v1/2022.naacl-srw.18, 2022.

[44] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," Proc. Annu. Meet. Assoc. Comput. Linguistics, pp. 8440-8451, DOI: 10.18653/v1/2020.acl-main.747, 2020.

[45] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, pp. 4171-4186, [Online], Available: https://aclanthology.org/N19-1423.pdf, 2019.

[46] L. Xue et al., "mT5: A Massively Multilingual Pre-trained Text-to-text Transformer," Proc. NAACL-HLT, pp. 483-498, DOI: 10.18653/v1/2021.naacl-main.41, 2021.

[47] M. Gutmann and A. Hyvärinen, "Noise-contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models," Journal of Machine Learning Research, vol. 9, pp. 297-304, 2010.

[48] A. Mnih and K. Kavukcuoglu, "Learning Word Embeddings Efficiently with Noise-contrastive Estimation," Advances in Neural Information Processing Systems, vol. 26, pp. 1-9, 2013.

[49] K. Sohn, "Improved Deep Metric Learning with Multi-class N-pair Loss Objective," Advances in Neural Information Processing Systems (NIPS 2026), vol. 29, 2016.

[50] Z. Wu et al., "Unsupervised Feature Learning *via* Non-parametric Instance Discrimination," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3733-3742, DOI: 10.1109/CVPR.2018.00393, 2018.

[51] P. Sermanet et al., "Time-contrastive Networks: Self-supervised Learning from Multi-view Observation," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 486-493, 2017.

[52] R. D. Hjelm et al., "Learning Deep Representations by Mutual Information Estimation and Maximization," arXiv preprint, arXiv: 1808.06670, 2019.

[53] Y. Tian, D. Krishnan and P. Isola, "Contrastive Multiview Coding," Lecture Notes in Computer Science, vol. 12356, pp. 776-794, DOI: 10.1007/978-3-030-58621-8_45, 2020.

[54] J. Snell, K. Swersky and R. Zemel, "Prototypical Networks for Few-shot Learning," Advances in Neural Information Processing Systems, vol. 30, 2017.

[55] E. van der Spoel et al., "Siamese Neural Networks for One-shot Image Recognition," Proc. of the 32nd Int. Conf. on Machine Learning, Lille, France, 2015.

[56] K. V. Nguyen et al., "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," Proc. Int. Conf. Knowl. Syst. Eng., pp. 115-120, DOI: 10.1109/KSE.2018.8573337, 2018.

[57] Z. Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence," Nature, vol. 521, no. 7553, pp. 452-459, DOI: 10.1038/nature14541, 2015.

[58] J. Snoek, H. Larochelle and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," Advances in Neural Information Processing Systems, vol. 25, 2012.

[59] Y. Xia et al., "A Boosted Decision Tree Approach Using Bayesian Hyper-parameter Optimization for Credit Scoring," Expert Systems With Applications, vol. 78, pp. 225-241, 2017.

[60] S. Tuan et al., "On Students' Sentiment Prediction Based on Deep Learning: Applied Information Literacy," SN Computer Science, vol. 5, no. 6, p. 928, 2024.

[61] R. Ahuja and S. C. Sharma, "Student Opinion Mining About Instructor Using Optimized Ensemble Machine Learning Model and Feature Fusion," SN Computer Science, vol. 5, no. 6, p. 672, 2024.

[62] D. V. Thin, D. N. Hao and N. L. Nguyen, "A Study of Vietnamese Sentiment Classification with Ensemble Pre-trained Language Models," Vietnam J. of Comp. Science, vol. 11, no. 2, pp. 137-165, 2023.

[63] X. Zhu, S. Wang, J. Lu, Y. Hao, H. Liu and X. He, "Boosting Few-shot Learning *via* Attentive Feature Regularization," arXiv preprint arXiv: 2403.17025, 2024.

[64] C. Huertas, "Gradient Boosting Trees and Large Language Models for Tabular Data Few-Shot Learning," Proc. Conf. on Computer Science and Information Systems, [Online], Available: https://www.semanticscholar.org/paper/273877899/paper/273877899, 2024.

ملخص البحث:

تتناول هذه الورقة التّحديات المرتبطة بتحليل المشاعر في السّياقات التّعليمية عبر اقتراح إطار عمل يجمع بين التّعليم قصير المدى والنّماذج القائمة على المحوّلات وتقنيات التّعزيز. حيث أنّ تحليل المشاعر يعد أمراً حاسماً لتحسين جودة التّعليم، ومن أبرز التّحديات شّح البيانات وضعف فاعلية الحوسبة. لذلك يعمل إطار العمل المقترح على تعزيز آليات الانتباه الذّاتي في المحوّلات، ويجمع بين النّماذج من خلال تقنيات التّعزيز لتحسين الأداء وإمكانية التّعميم بأقلّ قدْر من البيانات الموسومة. وجرى تطبيق إطار العمل المقترح على مجموعة بياناتٍ تحوي التّغذية الرّاجعة من الطّلبة باللّغة الفيتنامية، وحقّق نتائج مميزة في مهمّات تحليل المشاعر وتصنيف العناوين مقارنةً مع ما حققته النّماذج منفردة. وبالرغم من أنّ إطار العمل المقترح يمتلك الإمكانية لتحسين الخبرات التّعليمية إلّا إنّه يواجه بعض المحددات، مثل اعتماده على نماذج مدّربة مسبقاً وعلى تعقيد الحوسبة.